TEACHER STUDENT CONNECT

A PROJECT REPORT

Submitted by

M Ajay Bhargav
14MSE1101

In partial fulfillment for the award of the degree of

Master of Science

in

Software Engineering (5 Year Integrated Programme)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering
Vellore Institute of Technology
Vandalur - Kelambakkam Road, Chennai - 600 127

April – 2019

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

DECLARATION

I hereby declare that the project entitled **TEACHER STUDENT CONNECT** submitted by me to the School of Computing Science and Engineering, VIT Chennai, 600 127 in partial fulfillment of the requirements of the award of the degree of Master of Science in Software Engineering (5 year Integrated Programme) is a bona-fide record of the work carried out by me under the supervision of Prof. A.Muralidhar I further declare that the work reported in this project, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Place: Chennai
Date:

Signature of Candidate
(M. Ajay Bhargav)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

CERTIFICATE

This is to certify that the report entitled “Teacher Student Connect” is prepared and sub-mitted by M Ajay Bhargav (Reg. No. 14MSE1101) to VIT Chennai, in partial fulfillment of the requirement for the award of the degree of Master of Science in Software Engineering (5 years Integrated Programme) is a bona-fide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Guide/Supervisor
Name:
Date:

Program Chair
Name: Prof.A.Phamila
Date:

Examiner
Name:
Date:

Examiner
Name:
Date:

(seal of scse)

Acknowledgement

I obliged to give my appreciation to a number of people without whom I could not have completed this thesis successfully.

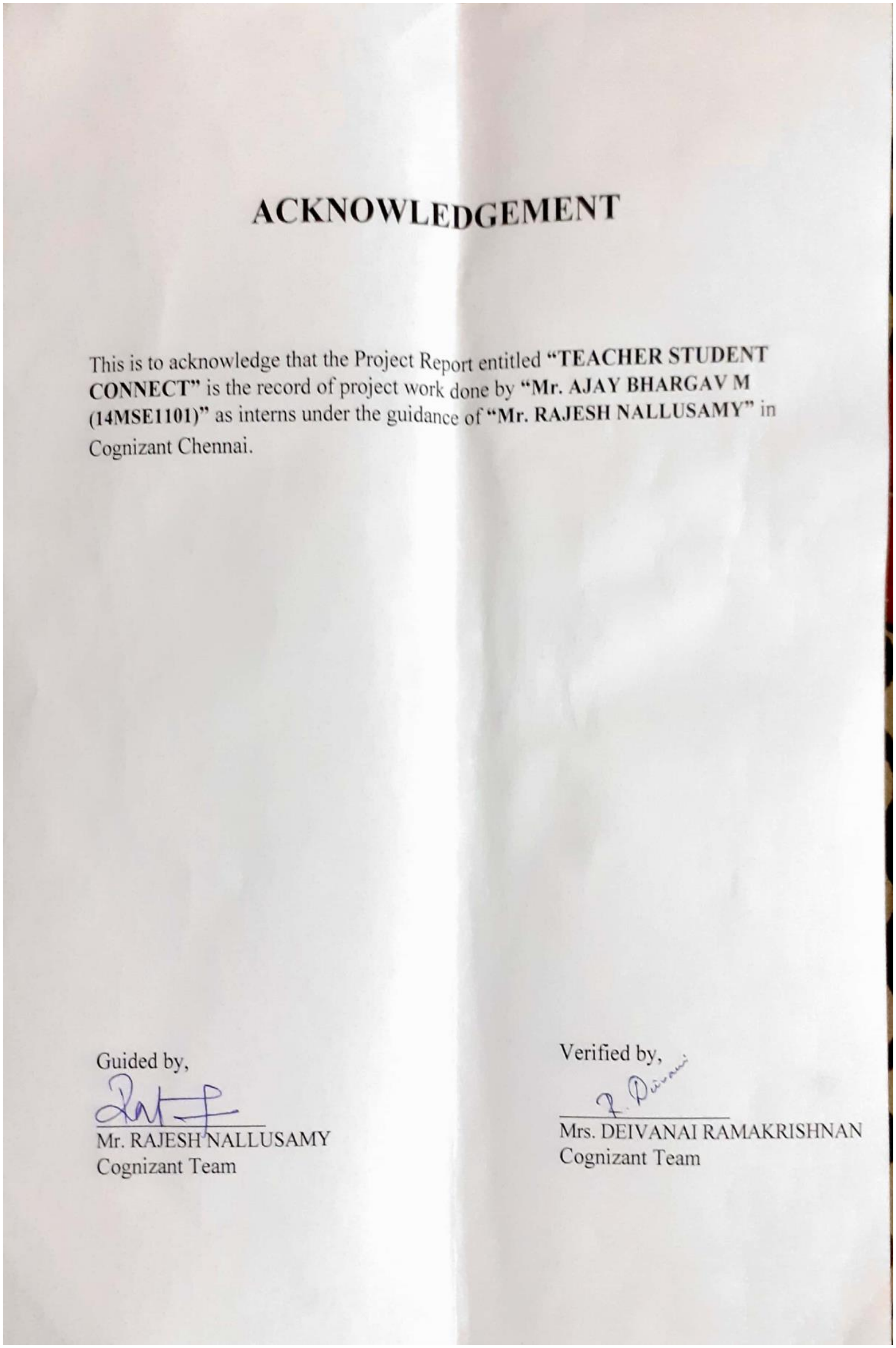
I would like to place on record my deep sense of gratitude and thanks to my external guide Nallusamy Rajesh, whose esteemed support and immense guidance encouraged me to complete the project successfully.

I would like to thank our Program manager Prof.A.Phamila, School of Computer Science and Engineering (SCSE) and Project Co-ordinator Dr.Malathi G, VIT University, for their valuable support and encouragement to take up and complete this thesis.

Special mention to our dean to Dr.VaidehiVijayakumar, Associate Deans Dr.Vijayakumar V and Dr.Subbulakshmi T, School of Computer Science and Engineering (SCSE), VIT University, for spending their valuable time and efforts in sharing his knowledge and for helping me in every minute aspect of software engineering.

I thank our management of VIT University, Chennai campus for permitting me to use the library resources. I also thank all the faculty members for giving me the courage and the strength that I needed to complete my goal. This acknowledgment would be incomplete without expressing the whole hearted thanks to my family and friends who motivated me during the course of my work.

M Ajay Bhargav
Reg. No. 14MSE1101



ACKNOWLEDGEMENT

This is to acknowledge that the Project Report entitled **"TEACHER STUDENT CONNECT"** is the record of project work done by **"Mr. AJAY BHARGAV M (14MSE1101)"** as interns under the guidance of **"Mr. RAJESH NALLUSAMY"** in Cognizant Chennai.

Guided by,

Mr. RAJESH NALLUSAMY
Cognizant Team

Verified by,

Mrs. DEIVANAI RAMAKRISHNAN
Cognizant Team

Abstract

A **Teacher Student Connect** project that acts as an online portal between students and the teachers. The system is designed for maintaining a good relation between student and teacher. It contains a teacher and student who can enter their own details in the portal. Students and Teachers can login using provided user id, password and they can download the documents. When students login they can see their teacher schedule and provide details of their profile with an image and students may download pdf format e-books from the web system. When teachers login they can upload and download the documents and they can edit their assigned classes schedule.

Contents

Declaration
Certificate ……………………………………………………………………1
Acknowledgement ……………………………………………………………………2
Abstract ……………………………………………………………………3

1. Introduction …………………………………………………………………...9

1.1. Scope & Purpose …………………………………………………………9

1.2. Project Overview …………………………………………………………10

1.2.1. Objectives …………………………………………………………10

2. Project description & Goals ……………………………………………………11

2.1. Proposed System …………………………………………………………..11

3. Technical Specifications …………………………………………………….12

3.1. Hardware Requirements …………………………………………………...12

3.2. Software Requirements ……………………………………………………12

3.3. Tools and Technologies used ………………………………………………13

3.3.1. Java ……………………………………………………………………13

3.3.2. JavaScript ………………………………………………………………13

3.3.3. HTML …………………………………………………………………14

3.3.4. CSS ……………………………………………………………………..14

3.3.5. Bootstrap ………………………………………………………………..15

3.3.6. MySQL …………………………………………………………………15

3.3.7. SQL ……………………………………………………………………...16

3.3.8. ABOUT MVC Framework ………………………………………………...16

3.3.9. MVC Components ……………………………………………………....16

3.3.9.1. Model ………………………………………………………………17

3.3.9.2. View ………………………………………………………………...17

3.3.9.3. Controller …………………………………………………………...17

3.3.10. MVC Framework ……………………………………………………………..18

3.3.11. Basic workflow of MVC ………………………………………………...18

3.3.12. Advantages of MVC …………………………………………………….20

3.3.13. Eclipse ……………………………………………………………………...20

3.3.13.1. Server Platform ……………………………………………...21

3.3.13.2. Web Tool Platform …………………………………………21

3.3.13.3. Modelling Platform …………………………………………21

3.3.14. MySQL Workbench ……………………………………………………..22

3.3.15. Apache Tomcat Server …………………………………………………..22

3.3.16. Components of Tomcat …………………………………………………23

3.3.16.1. Catalina ……………………………………………………..23

3.3.16.2. Coyote ………………………………………………………23

3.3.16.3. Jasper ………………………………………………………23

4. Design Approach & Details …………………………………………………..24

4.1. Introduction ……………………………………………………………24

4.2. System Architecture ……………………………………………………...24

4.3. UML Diagrams …………………………………………………………..25

4.3.1. Use-Case Diagram ………………………………………………...26

4.3.1.1. Login Module …………………………………………...27

4.3.1.2. Teacher Registration Module …………………………...27

4.3.1.3. Student Registration Module …………………………...28

4.3.1.4. Document Repository Module …………………………28

4.3.2. Sequence Diagram ………………………………………………..29

4.3.3. Class Diagram ……………………………………………………..30

4.3.4. Collaboration Diagram ……………………………………………31

4.3.5. Activity Diagram …………………………………………………32

5. Business Requirements …………………………………………………………….33

5.1. High Level Business requirements …………………………………………33

5.2. Detailed Business Requirements ……………………………………………33

6. Project Demonstration ……………………………………………………………..42

6.1. Implementation Practice ……………………………………………………42

6.2. Modules Implementation & Results …………………………………..43

6.2.1. Login Module …………………………………………………..43

6.2.2. Teacher Registration Module …………………………………..43

6.2.3. Student Registration Module …………………………………..44

6.2.4. Document Repository Module …………………………………44

7. Test Results ………………………………………………………………………...45

7.1. Test Cases ………………………………………………………………..45

8. Summary ……………………………………………………………………………...47

List of Figures

Fig 1: MVC Framework

Fig 2: Basic workflow of MVC

Fig 3: System Architecture

Fig 4: Use Case – Login Module

Fig 5: Use Case – Teacher Registration Module

Fig 6: Use Case - Student Registration Module

Fig 7: Use Case – Document Repository Module

Fig 8: Sequence Diagram

Fig 9: Class Diagram

Fig 10: Collaboration Diagram

Fig 11: Activity Diagram

1. INTRODUCTION

This chapter commences with a brief introduction about the organization in which the project was carried out and the services provided by the company. It also contains an overview of the project.

1.1. SCOPE & PURPOSE

This document is aimed at:

- Providing the necessary inputs to the detailed requirements gathering phase and further on for the SDLC processes.
- This document also serves to establish the traceability between the Business Objectives and the requirements identified in the proposed solution and how they satisfy the stated objectives.
- Provide expectation traceability in terms of the requirements and the user expectation
- Serves as a formal template for documenting the Business Requirements, which also includes statutory and regulatory requirements.

The purpose of this document is to systematically capture the requirements of the project and the system to be developed. The document also captures the Functional requirements and serves as an input for the scope of project.

1.2. PROJECT OVERVIEW

Creating an online Repository that contains a collection of learning and teaching materials flowing from Teachers and Students on a regular basis.

An Interactive UI platform that allows the Teachers and Students to digitally connect.

1.2.1 OBJECTIVES

Below are the objectives that shall be fulfilled post the execution of this project:

- Teachers and Students can view/engage in the courses from anywhere and anytime.
- Discovery, and retrieval of the learning materials from the repository along with other reference materials.
- Reuse of learning content.
- Cost effective and cost efficient solution of online knowledge base by hosting the repository on Cloud.
- Digital platform for the connect between Teachers and Students.

2. PROJECT DESCRIPTION AND GOALS

This chapter gives detail about the system that supposed to be developed. It provides the system requirements and explains in detail the various tools and technologies.

2.1. PROPOSED SYSTEM

In this project, I had developed for making a good relationship between teacher and student in a good manner and for easy access for the users.

Before proceeding for implementation of these requirements we have to give a proper study about the Teacher Student Connect and proceed the implementation of the requirements.

The technology used in this web application also should be known to implement the requirements; this project is mainly concentrate on the flexibility of the web application to the user.

For both teachers and students, the user interface was simple for them to maintain the account secure and hygiene. To make users to easily understand the access of the parameters and to the product owner to know what user uses parameters and collecting the track of the usage of Teacher Student Connect gives advantage.

3. TECHNICALSPECIFICATION

3.1 Hardware Requirements

Table 3.1 Hardware Requirements

Processor	Intel® Core™ i5
System Type	64-bit Operating System
Processor Speed	1.7 GHz minimum per core
Random Access Memory	4.00 GB
Hard Disk Drive	500 GB

Table 3.1 above discusses the system resource settings allotted for the specific device i.e. lists the hardware specification as part of the configuration of the system used to develop the application.

3.2 Software Requirements

Table 3.2 Software Requirements

Operating System	Windows 7,8,10
Language	JAVA, JS,HTML,CSS,BOOTSTRAP
Database	MySQL Workbench 8.0.14
Text Editor	Eclipse IDE for Java EE Developers

Table 3.2 above lists the software specification i.e. the software side of the configuration of the system used to develop the application. The operating system, framework, languages used, databases used and text editor used come under software specification.

3.3 Tools and Technologies Used

This sectiondescribes the tools and technologies that were usedin thesystemdevelopmentandanalysis.

3.3.1 JAVA

Java isa general-purpose programming language that is class-based, object-oriented,and designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.Java applications are typically compiled to "bytecode" that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2018, Java was one of the most popular programming languages in use according to GitHub,particularly for client-server web applications, with a reported 9 million developers.The latest version is Java SE 12, released in March 2019. Since Java 9 is no longer supported, Java 8 will be supported with public updates for personal use up to at least December 2020. Oracle and others "highly recommend that you uninstall older versions of Java"because of serious risks due to unresolved security issues.Oracle extended support for Java 6 ended in December 2018.

3.3.2 JAVASCRIPT

JavaScript is a scripting programming language that is used to make desktop widgets, applications, dynamic websites, PDF documents, developing game, etc. All web browsers support JavaScript. JavaScript is a client side technology, it is mainly used for client side validation and JavaScript language is used to make a webpage more dynamic and can be used again and again on different webpages of a website just by adding tags. JavaScript is an object-based scripting language which offers the features like handling dates and time,Manipulate HTML "layers" including hiding, moving, and allowing the user to drag them around a browser window,Performing simple computations on the client side.It uses the DOM model that provides plenty of prewritten functionality to the various objects on the pages making it a breeze to develop the script to solve the custom purpose. The code is executed on the user’s processor instead of the web server thus saving the bandwidth on the web server.

3.3.3 HTML

HTML (Hypertext Markup Language) is a text-based approach to describe how content contained within a webpage is structured. This markup tells a web browser how to display the text, images and other forms of multimedia on a webpage.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages,inclusion of CSS defines the look and layout of content.

3.3.4 CSS

CSS stands for Cascading Style Sheets with an emphasis placed on "Style." While HTML is used to structure a web document (defining things like headlines and paragraphs, and allowing you to embed images, video, and other media), CSS comes through and specifies your document's style—page layouts, colors, and fonts are all determined with CSS. CSS brings style to your web pages by interacting with HTML elements.CSS code (or Style Sheets) can be external, internal, or inline. External style sheets are saved as .css files.In order to use an external style sheet, your .html files need to include a header section that links to the external style sheet, internal style sheets are CSS instructions written directly into the header of a specific .html page. (This is especially useful if you have a single page on a site that has a unique look.), inline styles are snippets of CSS written directly into HTML code, and applicable only to a single coding instance.

3.3.5 BOOTSTRAP

Bootstrap is a web framework that focuses on simplifying the development of informative web pages. The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

3.3.6MySQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing. MySQL is an important component of an open source enterprise stack called LAMPMySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into separate applications. MySQL enables data to be stored and accessed across multiple storage engines, including InnoDB, CSV, and NDB. MySQL is also capable of replicating data and partitioning tables for better performance and durability.

3.3.7SQL

SQL (Structured Query Language) is a standardized programming language that's used to manage relational databases and perform various operations on the data in them. The uses of SQL include modifying database table and index structures; adding, updating and deleting rows of data; and retrieving subsets of information from within a database for transaction processing and analytics applications. Queries and other SQL operations take the form of commands written as statements. Commonly used SQL statements include select, add, insert, update, delete, create, alter and truncate. SQL commands are divided into several different types, among them data manipulation language (DML) and data definition language (DDL) statements, transaction controls and security measures. The DML vocabulary is used to retrieve and manipulate data, while DDL statements are for defining and modifying database structures. The transaction controls help manage transaction processing, ensuring that transactions are either completed or rolled back if errors or problems occur. The security statements are used to control database access as well as to create user roles and permissions.

3.3.8 ABOUT MVC FRAMEWORK

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

3.3.9MVC Components

- Model
- View
- Controller

3.3.9.1 Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data. Model is a simple class. It is the shape of the data. The model contains business logic. Controller and View can access the model. The model helps to pass the data from the controller to view and view to the controller.

3.3.9.2 View

The view is a user interface. It is used to display the entire data using the model. We are using two type of view engines in view. One is the traditional view engine and another one is the Razor view engine. Traditional view engine is normal ".ASPX" page. The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

3.3.9.3 Controller

The controller is the heart of MVC and it handles the user request. It is a simple class. The controller can access the model and pass data to view with the help of a model. We can pass the data between the controller and view using View Data, Temp Data, and View Bag.The controller is intermediate between model and view. Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

3.3.10 MVC Framework

This project depicts about the outline of model view controller and it has the functional flow of the MVC Framework

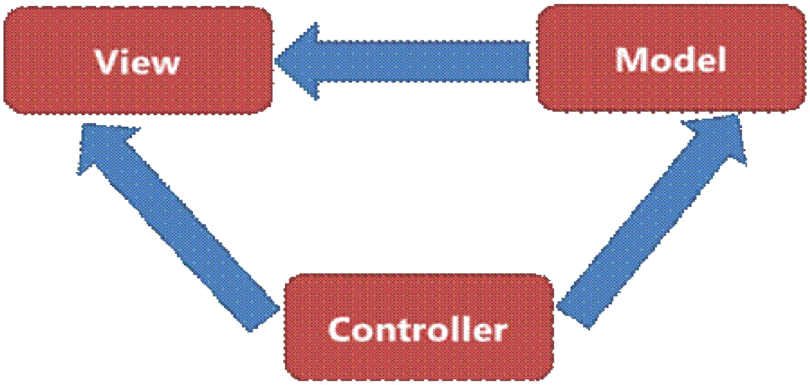


Figure 1. MVC Framework

3.3.11 Basic Work Flow of MVC

MVC workflow starts with the user’s request. Based on the request, first, it goes the controller then goes to the corresponding action method. In the action method, we are calling all layers like business logic layer and data access layer.

Once we reach action method then it goes to the Data access layer. Sometimes while requesting, the request contains some input data that binds with the model, and then goes to the data access layer.

Request reaches data access layer then goes to the corresponding database. It fetches the data from a database based on the request then goes back to the reverse format. After fetching data and binding in the model, it goes to the action method.

Action method returns the result to the corresponding view with the help of model. Now, the user gets the response. Action method returns results in a different format.

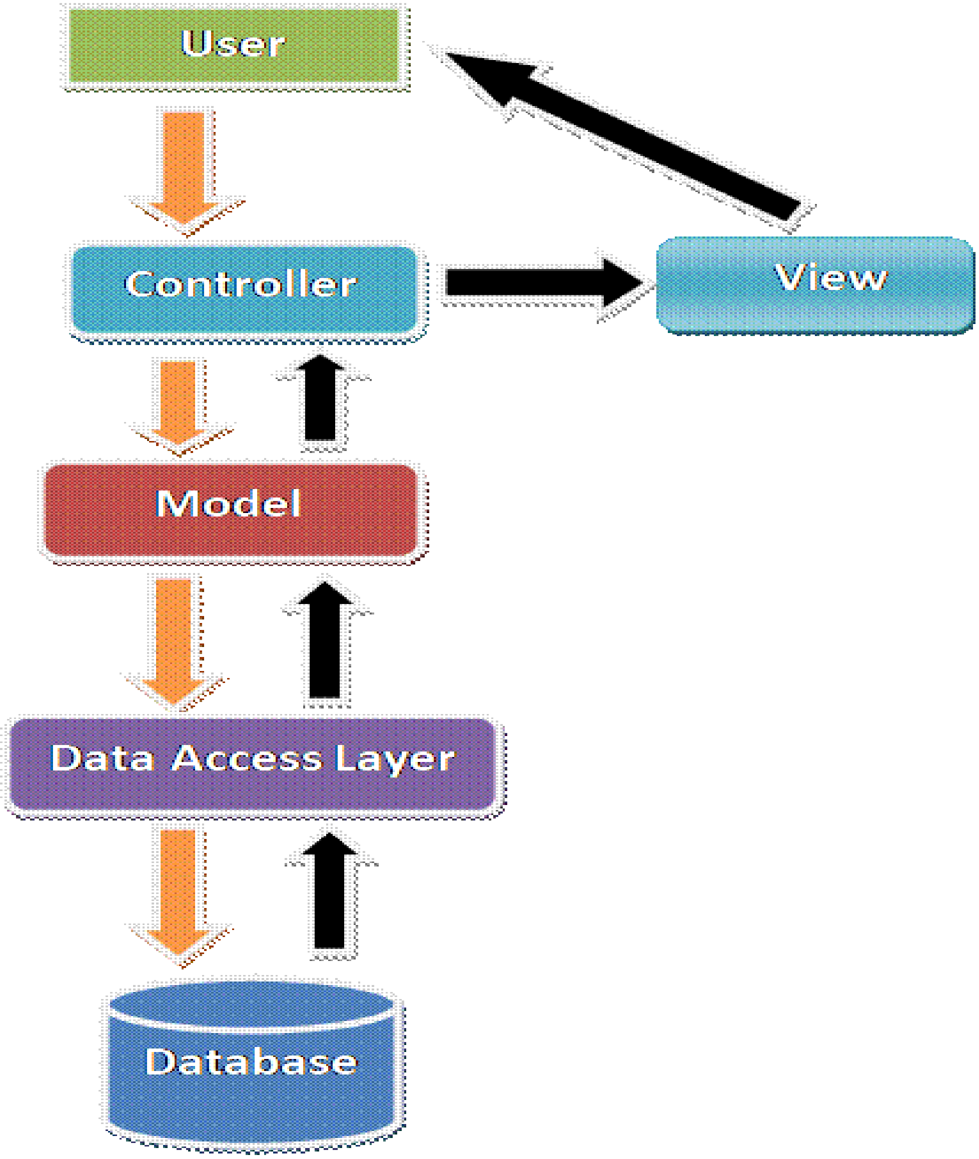


Figure 2. Work Flow of MVC

3.3.12 ADVANTAGES OF MVC:

- Separation of Concern (SOC) is the main advantage. Here we are separating Mode, View, and Controller.
- We can easily maintain MVC application.
- Test Drive Development (TDD) is another main advantage. We can create an application with a unit
- We can write our own test case.
- Split the application and many developers can work at a time without affecting each other.
- MVC application is a default responsive web site and mobile template.We can create our own view engine

3.3.13 ECLIPSE

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via-plug-in Including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python by (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others

The initial codebase originated from IBM Visual Age. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since the introduction of the OSGI implementation (Equinox) in version 3 of Eclipse, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundles.

Eclipse software development kit (SDK) is free and open-source software, released under the terms of the Eclipse Public License, although it is incompatible with the GNU General Public License. It was one of the first IDEs to run under GNU Class path and it runs without problems under IcedTea.

i) Server Platform

Eclipse supports development for Tomcat, Glass Fish and many other servers and is often capable of installing the required server (for development) directly from the IDE. It supports remote debugging, allowing a user to watch variables and step through the code of an application that is running on the attached server.

ii)Web Tool Platform

The Eclipse Web Tools Platform (WTP) project is an extension of the Eclipse platform with tools for developing Web and Java EE applications. It includes source and graphical editors for a variety of languages, wizards and built-in applications to simplify development, and tools and APIs to support deploying, running, and testing apps.

iii)Modeling Platform

The Modeling project contains all the official projects of the Eclipse Foundation focusing on model-based development technologies. All are compatible with the Eclipse Modeling Framework created by IBM.

Those projects are separated in several categories: Model Transformation, Model Development Tools, Concrete Syntax Development, Abstract Syntax Development, Technology and Research, and Amalgam.

3.3.14 MySQL Workbench

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of “My”, the name of co-founders Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook,Twitter, Flickr,and YouTube.

MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system. It is the successor to DB Designer 4 from fabFORCE.net, and replaces the previous package of software, MySQL GUI Tools Bundle.

3.3.15 APACHE TOMCAT SERVER

Apache Tomcat, often referred to as Tomcat Server, is an open-source Java Servlet Container developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, Java Server Pages (JSP), Java EL, and Web Socket, and provides a "pure Java" HTTP web server environment in which Java code can run.

3.3.16 Components of Tomcat

Tomcat 4.x was released with Catalina (a servlet container), Coyote (an HTTP connector) and Jasper (a JSP engine).

3.3.16.1 Catalina

Catalina is Tomcat's servlet container. Catalina implements Sun Microsystems's specifications for servlet and Java Server Pages (JSP). In Tomcat, a Realm element represents a "database" of usernames, passwords, and roles (similar to Unix groups) assigned to those users. Different implementations of Realm allow Catalina to be integrated into environments where such authentication information is already being created and maintained, and then use that information to implement Container Managed Security as described in the Servlet Specification.

3.3.16.2 Coyote

Coyote is a Connector component for Tomcat that supports the HTTP 1.1 protocol as a web server. This allows Catalina, nominally a Java Servlet or JSP container, to also act as a plain web server that serves local files as HTTP documents. Coyote listens for incoming connections to the server on a specific TCP port and forwards the request to the Tomcat Engine to process the request and send back a response to the requesting client. Another Coyote Connector, Coyote JK, listens similarly but instead forwards its requests to another web server, such as Apache, using the JK protocol. This usually offers better performance.

3.3.16.3 Jasper

Jasper is Tomcat's JSP Engine. Jasper parses JSP files to compile them into Java code as servlets (that can be handled by Catalina). At runtime, Jasper detects changes to JSP files and recompiles them. As of version 5, Tomcat uses Jasper 2, which is an implementation of the Sun Microsystems's JSP 2.0 specification. From Jasper to Jasper 2, important features were added:

- JSP Tag library pooling - Each tag markup in JSP file is handled by a tag handler class. Tag handler class objects can be pooled and reused in the whole JSP servlet.

4. DESIGN APPROACH AND DETAILS

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

4.1 INTRODUCTION

System design is the processor art of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. It explains how the functional requirements of the modules have been translated into software elements constituting the system. System design has two phases of development logical and physical design.

During logical design phase the analyst describes inputs(sources), outputs(destinations), databases (data sores) and procedures (data flows) all in a format that meets the user requirements.

4.2 SYSTEM ARCHITECTURE

System architecture is an approach to the design and planning of websites that involves technical, aesthetic and functional criteria. The focus is on the user and on user requirement. The main aim is to provide a framework that captures the entire component required for the design, to make the layering context clear and to position components as the concept that tie architectural system.

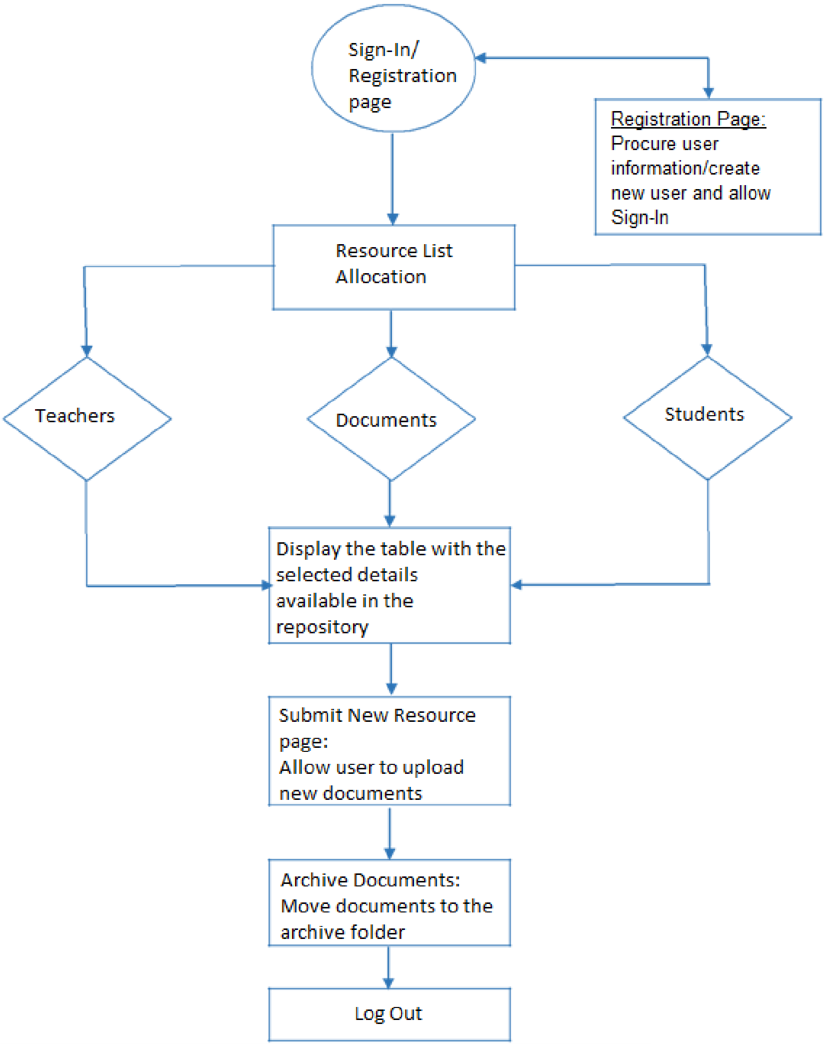

Sign-In/ Registration page
Registration Page: Procure user information/create new user and allow Sign-In
Resource List Allocation
Teachers
Documents
Students
Display the table with the selected details available in the repository
Submit New Resource page: Allow user to upload new documents
Archive Documents: Move documents to the archive folder
Log Out

Figure 3. System Architecture

4.3 UML DIAGRAMS

UML is an acronym that stands for Unified Modelling Language. UML is a modern approach to modelling and documenting software. It is one of the most popular business process modelling techniques. UML has been used as a general-purpose modelling language in the field of software engineering. UML is a standardized modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artefacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

4.3.1 Use Case Diagram

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases.

We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

- Functionalities to be represented as use case
- Actors
- Relationships among the use cases and actors.

4.3.1.1 Login Module.

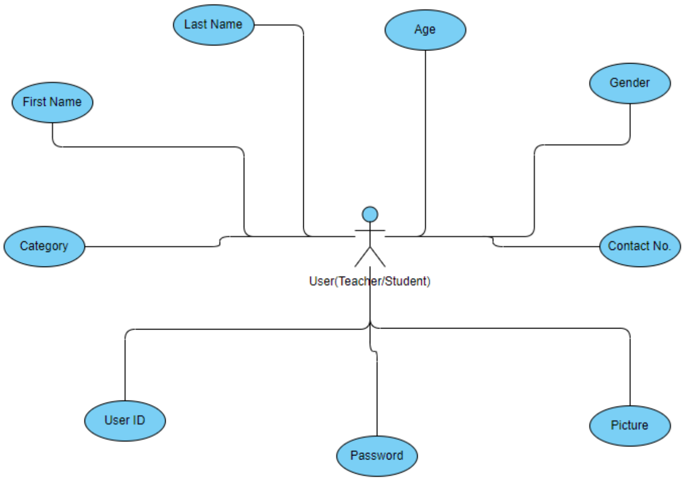


Figure 4. Login Module

4.3.1.2 Teacher Registration Module.

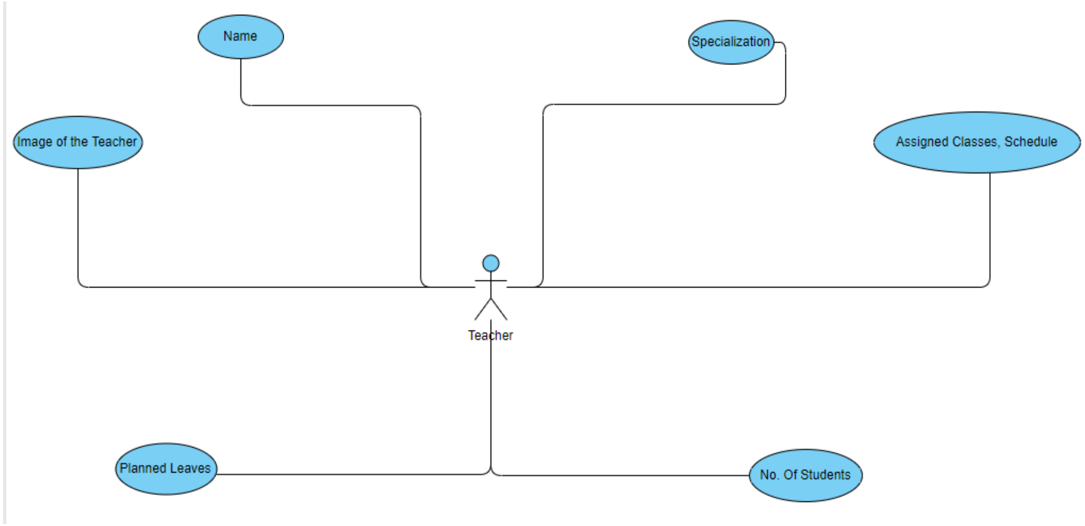


Figure 5. Teacher Registration Module

4.3.1.3 Student Registration Module.

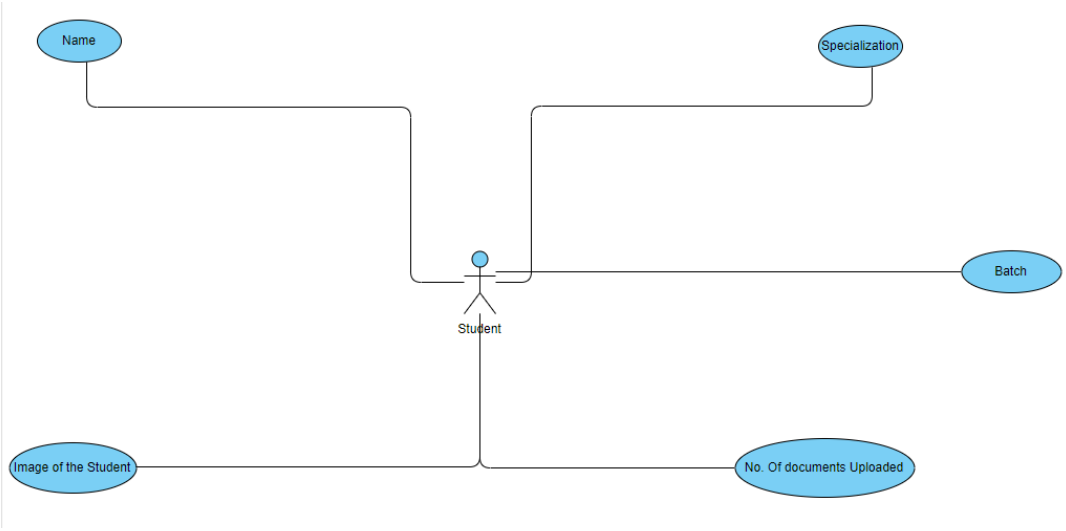


Figure 6. Student Registration Module

4.3.1.4 Document Repository Module.

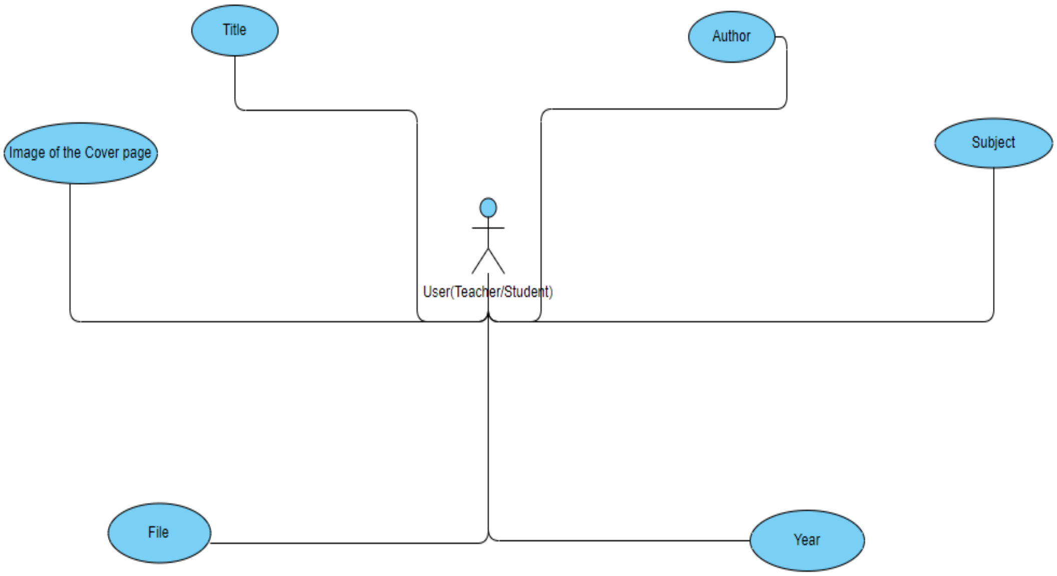


Figure 7. Document Repository Module

4.3.2 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

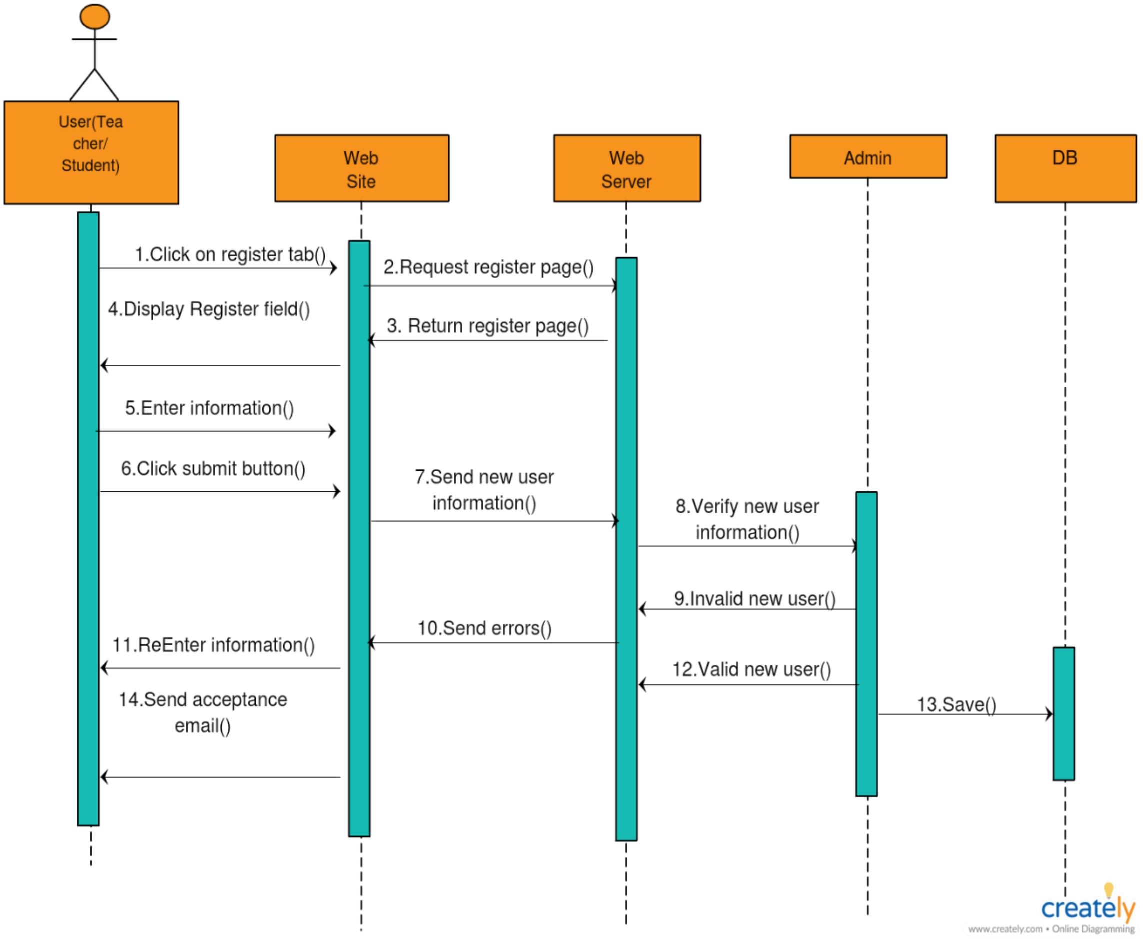

User(Teacher/Student)
Web Site
Web Server
Admin
DB
1.Click on register tab()
2.Request register page()
3. Return register page()
4.Display Register field()
5.Enter information()
6.Click submit button()
7.Send new user information()
8.Verify new user information()
9.Invalid new user()
10.Send errors()
11.ReEnter information()
12.Valid new user()
13.Save()
14.Send acceptance email()
creately
www.creately.com • Online Diagramming

Figure 8. Sequence Diagram

4.3.3 Class Diagram

The class diagram describes the attributes and operations of a class and the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structuraldiagram. Class Diagram helps construct the code for the software application development. Class Diagram illustrates data models for even very complex information systems. It provides an overview of how the application is structured before studying the actual code. This can easily reduce the maintenance time. It helps for better understanding of general schematics of an application.

Essential elements of UML class diagram are:

1. Class Name
2. Attributes
3. Operations

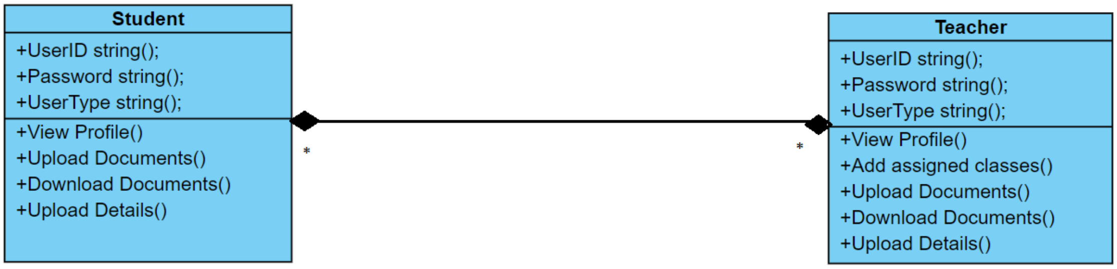


Figure 9. Class Diagram

4.3.4 Collaboration Diagram

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

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

Collaboration diagrams are best suited to the portrayal of simple interactions among relatively small numbers of objects. As the number of objects and messages grows, a collaboration diagram can become difficult to read. Several vendors offer software for creating and editing collaboration diagrams.

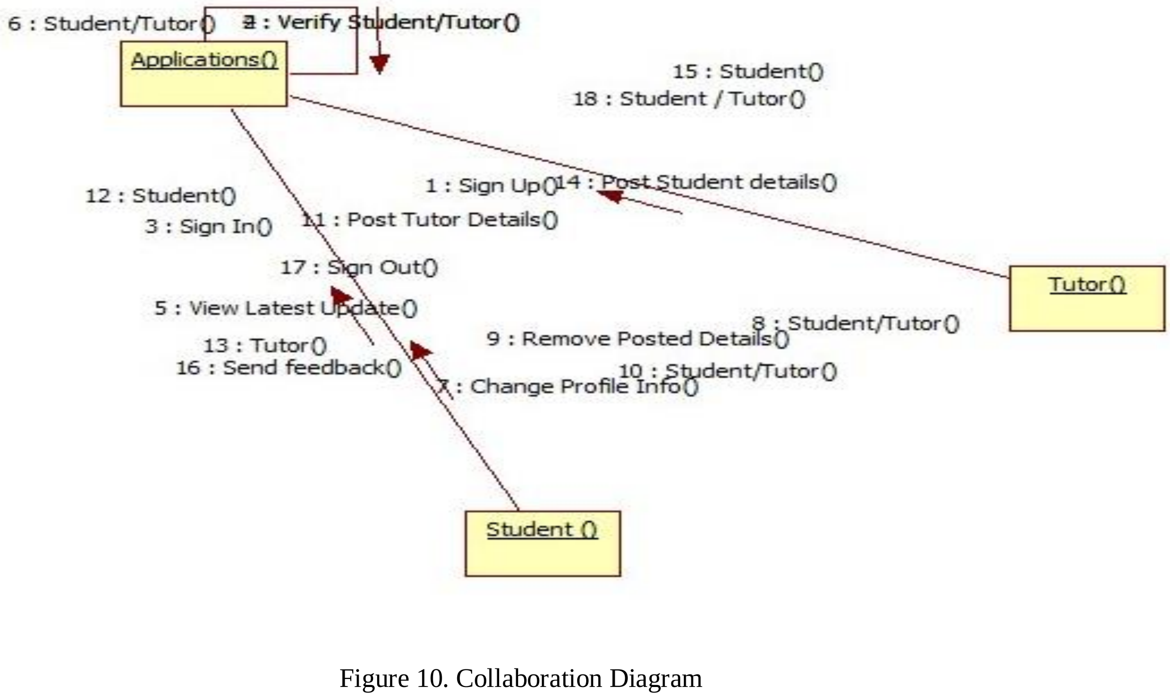


Figure 10. Collaboration Diagram

4.3.5 Activity Diagram

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swim lane, etc.

Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

Before drawing an activity diagram, we should identify the following elements −

- Activities
- Association
- Conditions
- Constraints

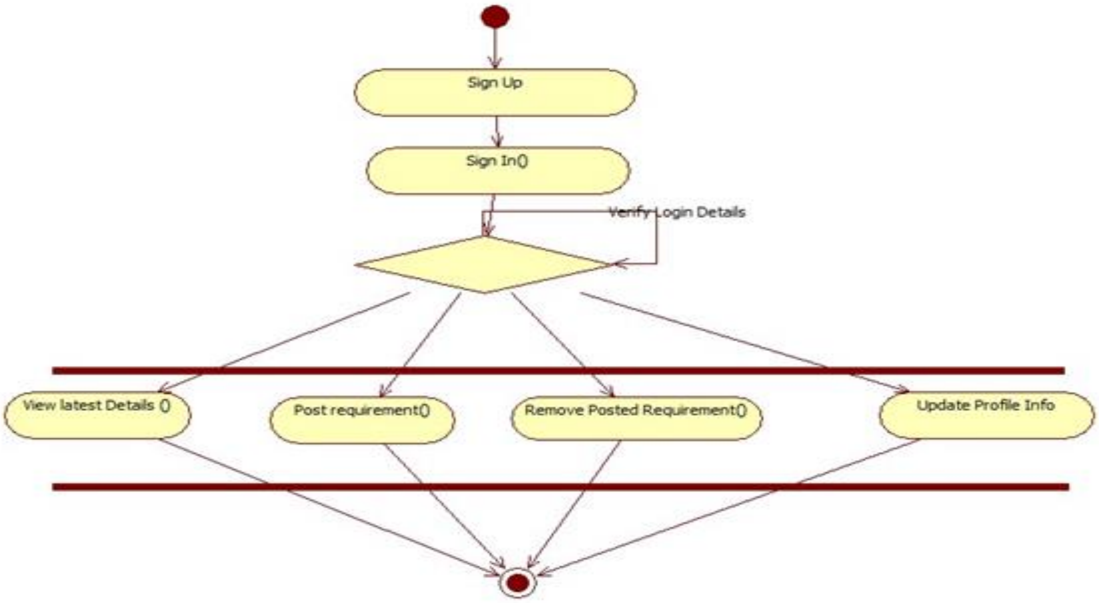


Figure 11. Activity Diagram

5. BUSINESS REQUIREMENTS

5.1 High Level Business Requirements:

S.No.	Business Requirement ID	Short Description	Description in detail	Interacting Business Processes
1	REQ-1	New User Registration	Ability of the portal to procure user data and create new user profiles	
2	REQ-2	User login	Ability of the portal to authenticate and authorize the registered user	
3	REQ-3	Resource List	Displays the complete list of selected resources available in the repository	
4	REQ-4	Advanced Search	Allows the user to search for specific Documents/Teacher/Student	
5	REQ-5	Submit new resource	Allows the Teachers or Students to upload new documents to the repository	
6	REQ-6	Archive Documents	Allows the Teachers to move the documents to archive folder	
7	REQ-7	Help	Page to assist the users to report technical issues	

5.2 Detailed Business Requirements:

Req. #	Rationale	Business	Req. Type	Priority	Originator	BR Traced to Business Requirement/ Usecase
	Categorization	Requirement		**	***	D
1.01	New User Registration	Clicking on the "Register" button should open the Registration form.	UI	Critical	NA	REQ-1
1.02	New User	User to fill the basic	UI		NA	REQ-1

	Registration	attributes: First Name Last Name Age Gender Contact Number Category User Id Password Picture		Critical		
1.03	New User Registration	Clicking ‘Submit’ button should validate the datatype constraints for each field	F	Critical	NA	REQ-1
1.04	New User Registration	User failing to provide information for the mandatory fields be presented with an alert message – ‘Please update the highlighted mandatory field(s).’ Also, highlight the missed out field in red	E	Medium	NA	REQ-1

1.05	New User Registration	Post-successful field validation, save the information in the database	F	Critical	NA	REQ-1
1.06	New User Registration	Upon saving the information in the database, display the message “New user created successfully”	E	Medium	NA	REQ-1
2.01	User login	Clicking “SignIn” link should allow the user to enter the User ID and Password and click “Login” button	F	Critical	NA	REQ-2
2.02	User login	Portal to validate the entered user credentials against the database and allow login for registered users.	F	Critical	NA	REQ-2
3.01	Resource List	Clickling on “Resource List” link	F		NA	REQ-3

		should display the below choices: Document Teacher Student		Critical		
3.02	Resource List	Portal to allow the user to select the required choice from the above	UI	Critical	NA	REQ-3
3.03	Resource List	Selecting “Document” should display table format of the entire list of documents that are stored in the the repository	F	Critical	NA	REQ-3
3.04	Resource List	Allow user to choose sorting order: Sort by Year (Newest First) Sort by Year (Oldest First) Sort by Title (A to Z) Sort by Title (Z to A) Sort by Author (A to	UI		NA	REQ-3

		Z) Sort by Author (Z to A) Sort by Subject (A to Z) Sort by Subject (Z to A)		Critical		
3.05	Resource List	Selecting a particular document from the resultant table should allow the user to preview the document in a new tab	F	Critical	NA	REQ-3
3.06	Resource List	Selecting “Teacher” should display table format of the complete list of Teachers, registered in the the portal	F	Critical	NA	REQ-3
3.07	Resource List	Allow user to choose sorting order: Sort by Name (A to Z) Sort by Name (Z to	UI		NA	REQ-3

		A) Sort by Specialization (A to Z) Sort by Specialization (Z to A)		Critical		
3.08	Resource List	Clicking on a particular Teacher details from the resultant table should allow the user to preview the details in a new tab	F	Critical	NA	REQ-3
3.09	Resource List	Clicking on the “Assigned classes, schedule” hyperlink should open a new tab with the schedule in table format	F	Critical	NA	REQ-3
3.1	Resource List	Selecting “Student” should display table format of the complete list of Students registered in the the portal	F	Critical	NA	REQ-3

3.11	Resource List	Allow user to choose sorting order: Sort by Name (A to Z) Sort by Name (Z to A) Sort by Batch (A to Z) Sort by Batch (Z to A)	UI	Critical	NA	REQ-3
3.12	Resource List	Selecting a particular Student details from the resultant table should allow the user to preview the details in a new tab	F	Critical	NA	REQ-3
4.01	Advanced Search	Clicking “Advanced Search” should display a form with below fields: Keyword Search Title Author Subject Year	UI	Critical	NA	REQ-4

4.02	Advanced Search	The display form should have buttons to “Submit Query” or “Reset” the fields	UI	Critical	NA	REQ-4
4.03	Advanced Search	Clicking on “Submit Query” button should display the required documents and allow user to download the file	UI	Critical	NA	REQ-4
5.01	Submit new Resource	Clicking “Submit new Resource” should display a form with below fields: Subject/Topic Title Author Subject Year Upload/Attach file	UI	Critical	NA	REQ-5
5.02	Submit new Resource	Clicking on “Submit” button should perform backend	F		NA	REQ-5

		validation of the user profile and upload the document		Critical		
6.01	Archive Document	Portal capability to allow Teachers to archive documents	F	Medium	NA	REQ-6
7.01	Help	Page that displays the basic details text about how to search documents	F	Low	NA	REQ-6
7.02	Help	Portal to display a form that allows user to report the technical issues through Email: Issue Description Send Button	F	Low	NA	REQ-6
7.03	Help	Portal to display the Contact number to report issues	F	Low	NA	REQ-6

6. PROJECT DEMONSTRATION

6.1 IMPLEMENTATION PRACTICE

Implementation is the carrying out, execution, or practice of a plan, a method, or any design, idea, model, specification, standard or policy for doing something. As such, implementation is the action that must follow any preliminary thinking in order for something to actually happen. Companies strive to use proven methodologies and enlist professional help to guide them through the implementation of a system but the failure of many implementation processes often stems from the lack of accurate planning in the beginning stages of the project due to inadequate resources or unforeseen problems that arise.

During the implementation process, engineers apply the design properties and/or requirements allocated to a system element to design and produce a detailed description. They then fabricate, code, or build each individual element using specified materials, processes, physical or logical arrangements, standards, technologies, and/or information flows outlined in detailed descriptions (drawings or other design documentation). A system element will be verified against the detailed description of properties and validated against its requirements. Depending on the technologies and systems chosen when a decision is made to produce a system element, the implementation process outcomes may generate constraints to be applied on the architecture of the higher-level system; those constraints are normally identified as derived system requirements and added to the set of system requirements applicable to this higher-level system. The architectural design has to consider those constraints. . Implementation also involves packaging, handling, and storage, depending on the concerned technologies and where or when the system requirement needs to be integrated into a higher-level aggregate.

6.2 MODULES IMPLEMENTATION AND RESULTS

A module is a software component or part of a program that contains one or more routines. One or more independently developed modules make up a program. An enterprise-level software application may contain several different modules, and each module serves unique and separate business operations. These modules may work as basic constructs for the entire software. Designers tend to design modules such that they can be executed and/or compiled separately and independently.

This project consist of several modules, such as

1. Login Module.

2. Teacher Registration Module.

3. Student Registration Module.

4. Document Repository Module.

6.2.1 Login Module

Login module is a module that is completely dedicated for the client user of the system. It includes all the operations, functionalities and specifications that have been stated by the user. This module offers the sophisticated interaction with the system in every possible perspective.

6.2.2 Teacher Registration Module

In this module, Teachers who are new to the portal have to sign-up (i.e., to fill up the details) they are allowed to update the personal details and update scheduled timetable. Teachers can view their students who are registered under them.

6.2.3 Student Registration Module

In this module, Students who are new to the portal have to sign-up (i.e., to fill up the details) they are allowed to update the personal details and view their day-to-day timetable. Students can view the list of teachers and can register under desired faculty based on their availability.

6.2.4 Document Repository Module

In this Document repository module, both the teachers and students access the documents. There are two options in this module, such that one can upload the documents and can download the documents, which are available in the database based on their sorting order by Subject/ Author/Title.

7. TEST RESULTS

7.1 TEST CASES

A test case, in software engineering, is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do. In some settings, it could be a requirement or use case, while in others it could be a heuristic. Test cases are often referred to as test scripts, particularly when written - when they are usually collected into test suites.

A test case is a set of conditions under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

Table 7.1 Test Cases for System Modules

Module/ Component *	Test Case Name *	Expected Result *	Actual Result**	PASS / FAIL**
User Registration	Access Registration Page	User should be able to view and edit in registration page	User should be able to view and edit in registration page	PASS
Login	Login	Logins if the credentials are existing	Logins if the credentials are existing	PASS
		If credentials are not present, invalid message is displayed	If credentials are not present, invalid message is displayed	PASS
Document	Access DOCUMENTS page	Redirects to Document home page	Redirects to Document home page	PASS

		Displays the existing documents and can upload new documents	Displays the existing documents and can upload new documents	PASS
Teacher	Access TEACHER Page	Redirects to Teacher home page	Redirects to Glucose home page	PASS
		Enter the teacher details and can view their allotted schedule	Enter the teacher details and can view their allotted schedule	PASS
		Data is stored in the database and displayed to the Teacher	Data is stored in the database and displayed to the Teacher	PASS
Submit new resource	Access new resource creating Page	check whether it is stored in the database	check whether it is stored in the database	PASS
Student	Access STUDENT page	Redirects to Student home page	Redirects to Student home page	PASS
		Enter the student details	Enter the student details	PASS
		Data is stored in the database and displayed to the Student	Data is stored in the database and displayed to the Student	PASS

8. SUMMARY

This chapter summarizes the whole project, points out the assets of the project and includes the areas of future enhancements.

The project report entitled"Teacher Student Connect" has come to its final stage. The system have been developed with much care that it is free of errors and at the same time, it is efficient and less time consuming. We have tried our level best to make the application as dynamic as possible. The developed system is flexible and changes can be made easily. The entire system is secured. This system will be approved and implemented soon in live.