

# Software Project Final Report

***Submitted by***

1. Netra Ramesh (2862120)
2. Bindu Manjunath Raju (2862245)
3. Aradhana Nandhikeswaran (2861686)

Table of Contents

# Introduction

* 1. About the project
  2. Project description
     1. Objectives
  3. Modules
     1. Professor admin module
     2. Student user module

# Requirement Specifications

* 1. Hardware Requirements
  2. Software Requirements
  3. Uses cases
  4. Graphical use case model

2.4.1 Description of each use case:

* 1. Functional requirements
  2. Non-functional requirements

# Project Management Plan

* 1. Project Deliverables

# Architecture

* 1. Architectural style(s) used
  2. Architectural model (includes components and their interactions)
  3. Architecture diagram

# Design Specification

* 1. Design goals and objectives

5.1.2 Statement of scope

5.1.3 Software context

5.1.4 Major constraints

* 1. Data Design

5.2.1 Database description

* 1. Component level design

5.3.2 Architecture diagrams

5.4 Description for components

5.4.1 Component and description

* 1. Design Models
  2. User interface design
  3. Interface design rules
  4. Restrictions, limitations and constraints

# Test Management

* 1. Test Objectives:
  2. Scope
  3. Test Approach

6.3.1 Levels of Testing

6.3.2 Test Cycles

* 1. Test Cases
  2. Traceability of test cases to use cases
  3. Test results and Assessments

# Conclusions

* 1. Outcomes of the project (
  2. Lessons learned
  3. Future enhancements

# List of Figures

|  |  |
| --- | --- |
| Figures | Description |
| 1. | Use-case diagram |
| 2. | Architecture diagram |
| 3. (a) | Create new class |
| (b) | Create assignment |
| (c) | Student access |
| 4. | Sequence diagram -1 |
| 5. | Sequence diagram -2 |
| 6. | Activity diagram |
| 7. | Login portal |
| 8. | Student signup |
| 9. | Faculty signup |
| 10. | Create assignment |
| 11. | Assignment upload |
| 12. | Anonymous feedback |
| 13. | Reward assignment |

**List of Tables**

|  |  |
| --- | --- |
| 1. | Table 1-Usecase 1 |
| 2. | Table 2-Usecase 2 |
| 3. | Table 3-Usecase 3 |
| 4. | Table 4-Usecase 4 |
| 5. | Table 5-Usecase 5 |
| 6. | Table 6-Usecase 6 |
| 7. | Table 7-Usecase 7 |
| 8. | Table 8-Usecase 8 |
| 9. | Table 9-Usecase 9 |
| 10. | Table 10-Usecase 10 |
| 11. | Table 11-Usecase 11 |
| 12. | Table 12-Usecase 12 |
| 13. | Table 13-Usecase 13 |
| 14. | Table 14-Usecase 14 |
| 15. | Table 15- Project Deliverables |
| 16. | Table 16- Test case 1 |
| 17. | Table 17- Test case 2 |
| 18. | Table 18- Test case 3 |
| 19. | Table 19- Test case 4 |
| 20. | Table 20- Test case 5 |
| 21. | Table 21- Test case 6 |
| 22. | Table 22- Test case 7 |
| 23. | Table 23- Test report case 1 |
| 24. | Table 24- Test report case 2 |
| 25. | Table 25- Test report case 3 |
| 26. | Table 26- Test report case 4 |
| 27. | Table 27- Test report case 5 |

# Introduction

* 1. **About Project**

**Edu-Learn** is a web-based application, a learning platform that offers students and faculties to upload, download and view study materials shared. This is a modern solution to exchange ideas and notes by professors to students. Finding and governing contemporary ideas and thoughts of getting everything online are quickly changing as the world is built with current technologies. The other highlights of this website are that student users can offer anonymous feedback, notifying student when faculty downloads students’ work.

# Project Description:

The scope of the project is wide because it does not target a specific organization. This project develops general- purpose software that can be used by any business organization. In addition, it provides facilities for users. Also, the software provides a large amount of summarized data.

Titled "Edu-Learn", the goal of this project is to provide a user-friendly and intuitive GUI so that users can easily upload their study materials. Student module can also update note details and delete uploaded notes, give anonymous feedback. This Edu-learn site project allows faculties to create classes, assignments, students and faculty can upload/download technical subject notes, Students can submit their assignments. The project deployed his two main modules, a student module and professor module.

# Objectives:

The internet has grown increasingly popular as the rest of the world adopts cutting-edge technology. A quality website is now seen as a reflection of a company. We may establish virtual classrooms using the internet, participate in online exams, display research and articles, and more. The goal of this "Edu Learn Site" project is to provide a user-friendly and easy-to-understand graphical interface for users to easily upload topic notes. User can also update details of notes and delete study materials or work uploaded by them.

The main points are:

* + - * Simplify the management of all notes.
      * Create Classes
      * Feature to create and submit assignments
      * Provide anonymous feedback
      * Manage profiles

# Modules:

* + 1. **Professor admin module:**

In this module, professor can create classes, review, update grades. Professor can also update student comments on assignments in the comments section and can also assign grades to a specific assignment.

* Manage (View/Delete) Notes
* Accept or Reject Notes
* Change Password
* Create Assignments
* Download assignments
* Reward assignments

# Student user module:

In this module, student can join classes, view assignments, upload assignments and provide anonymous feedbacks.

* Download Notes
* Join classes
* Change Password
* Upload assignments
* Student users can offer anonymous feedback

# Requirement Specifications:

* 1. **HARDWARE REQUIREMENTS:**

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

* RAM : 2 GB (OR) Higher
* Hard disk or SSD : More than 100 GB
* Processor : Intel i5 Processor

# SOFTWARE REQUIREMENTS:

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

* + Technology: Angular Framework

Angular CLI is a command-line interface tool that automates the application development process by initializing new Angular applications and maintaining them directly from a command shell

* + IDE: Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps

* + Data Base Server: JSON server

JSON Server is a library that connects to a SQL database instance, stores JSON documents in tables and queries JSON documents through a JSON query language

* + Operating System: Microsoft Windows/Linux

# Use cases:

* Register
* Login
* Create class
* Create Assignment
* Upload Study material
* View submitted work
* Grading
  + View anonymous feedback
  + Join class
  + Submit assignment
  + Download study material
  + View grades
  + Send anonymous feedback

# Graphical use case model

Diagram

Description automatically generated

Fig.1 Use-case diagram

# Description of each use case:

|  |  |
| --- | --- |
| Use Case ID | Use case 1 |
| Use Case Name | Register |
| Description | Professor can Register into the website by entering the credentials. |
| Actors | Professor and student |
| Preconditions | User needs to open the website. |
| Data | Username, Password, Mail ID, Name, Student ID |
| Stimulus | Professor and student has to enter their Name, Email, Password, Confirm Password, Address |
| Response | User will be registered into the application |
| Steps | 1. Register user |
| Comments | If user exists, then the user is requested to login instead. |

Table.1- Use case 1

|  |  |
| --- | --- |
| Use Case ID | Use case 2 |
| Use Case Name | Login |
| Description | Professor and student can Login into the website by entering the credentials. |
| Actors | Professor and student |
| Preconditions | User needs to open the website |
| Data | Username, Password |
| Stimulus | Professor and student have to enter the username and password |
| Response | User will be logged on to the website |
| Steps | 1. Register 2. Login |
| Comments | If username and password is wrong, user cannot login. |

Table.2 -Use case 2

|  |  |
| --- | --- |
| Use Case ID | Use case 3 |
| Use Case Name | Create Class |
| Description | Professor creates the class |
| Actors | Professor |
| Preconditions | Professor needs to open the website. |
| Data | Class name, Branch, Class Code |
| Stimulus | Class code |
| Response | Class will be created |
| Steps | 1. Create class 2. Enter class name 3. Enter Branch 4. Enter class code |
| Comments | Professor can create as many classes as they need. |

Table.3 -Use case 3

|  |  |
| --- | --- |
| Use Case ID | Use case 4 |
| Use Case Name | Create Assignment |
| Description | Professor can create assignment for their respective classes |
| Actors | Professor |
| Preconditions | Professor has to have created the class |
| Data | Assignment, Description |
| Stimulus | Assignment template |
| Response | User will be registered into the application |
| Steps | 1. Create Class 2. Create assignment 3. Enter description |
| Comments | Professor can issue many assignments as required. |

Table.4- Use case 4

|  |  |
| --- | --- |
| Use Case ID | Use case 5 |
| Use Case Name | Upload Study materials |
| Description | Professor can upload study notes for respective classes |
| Actors | Professor |
| Preconditions | Study notes should be available on the computer as they upload and should be accessible to the website |
| Data | Study materials |
| Stimulus | Professor can make the materials accessible for their students |
| Response | Study notes will be uploaded |
| Steps | 1. Choose class 2. Choose upload file 3. Select the available file on computer upload it. |
| Comments | The notes will be made available for students |

Table.5 -Use case 5

|  |  |
| --- | --- |
| Use Case ID | Use case 6 |
| Use Case Name | View submitted work |
| Description | Professor can view the assignments submitted by the students |
| Actors | Professor |
| Preconditions | Students should have joined and submitted completed work |
| Data | Completed assignment |
| Stimulus | Submitted work will be available on the website |
| Response | Download and view the work |
| Steps | 1. Download assignment |
| Comments | Reject assignment if the requirements are not filled |

Table.6 -Use case 6

|  |  |
| --- | --- |
| Use Case ID | Use case 7 |
| Use Case Name | Grade and give remarks |
| Description | Professor can Grade the assignment and give remarks |
| Actors | Professor |
| Preconditions | Professor should download the assignment |
| Data | Assignment submitted by the student |
| Stimulus | Downloaded work done |
| Response | Grade and give remarks |
| Steps | 1. Grade the assignment 2. Give remarks |
| Comments | Faculty can grade give remarks |

Table.7- Use case 7

|  |  |
| --- | --- |
| Use Case ID | Use case 8 |
| Use Case Name | View anonymous feedback |
| Description | Professor can view the anonymous feedback given by the students |
| Actors | Professor |
| Preconditions | Professor should be logged into the website |
| Data | Feedback offered by students |
| Stimulus | Feedback given |
| Response | View feedback and improvise if necessary |
| Steps | 1. Open class 2. View feedback |
| Comments | View feedback |

Table.8 -Use case 8

|  |  |
| --- | --- |
| Use Case ID | Use case 9 |
| Use Case Name | Join class |
| Description | Students can join classes for their respective departments |
| Actors | Student |
| Preconditions | Student has to be registered in the website |
| Data | Class Code and branch name |
| Stimulus | Class code, Branch and Course name |
| Response | Class joined |
| Steps | 1. Enter class code 2. Join class |
| Comments | Can join class only once |

Table.9 -Use case 9

|  |  |
| --- | --- |
| Use Case ID | Use case 10 |
| Use Case Name | Submit assignment |
| Description | Student has to submit assignments for respective subjects |
| Actors | Student |
| Preconditions | Student has to submit completed work for the respective classes |
| Data | Homework assignment material |
| Stimulus | Assignment should be available on the computer |
| Response | Work submitted |
| Steps | 1. Complete assignment 2. Submit assignment |
| Comments | Assignment submitted. |

Table.10- Use case 10

|  |  |
| --- | --- |
| Use Case ID | Use case 11 |
| Use Case Name | Download Study materials |
| Description | Student can download materials offered by teachers |
| Actors | Student |
| Preconditions | Student has to join the respective class |
| Data | Study notes |
| Stimulus | Material uploaded by the professor |
| Response | Study notes available |
| Steps | 1. Download materials |
| Comments | Study material downloaded |

Table.11 -Use case 11

|  |  |
| --- | --- |
| Use Case ID | Use case 12 |
| Use Case Name | View grades |
| Description | Students can view their grades for every subject |
| Actors | Student |
| Preconditions | Student has to enroll in the class |
| Data | Grades given by professor |
| Stimulus | Student has to enroll in class and view their grades |
| Response | View the grade offered |
| Steps | 1. View grades |
| Comments | The grades cannot be viewed if not given by the faculty |

Table.12- Use case 12

|  |  |
| --- | --- |
| Use Case ID | Use case 13 |
| Use Case Name | Notified when professor views the assignment |
| Description | Student gets notified when professor views the assignment |
| Actors | Student |
| Preconditions | Assignment has to be submitted |
| Data | Notified if the work done is viewed |
| Stimulus | Assignment submission |
| Response | Status of the submitted work |
| Steps | 1. View grades 2. Status of the assignment after submission |
| Comments | If the faculty has not downloaded the work student will be notified as not viewed |

Table.13 -Use case 13

|  |  |
| --- | --- |
| Use Case ID | Use case 14 |
| Use Case Name | Send anonymous feedback |
| Description | Student can send anonymous feedback to the professor |
| Actors | Student |
| Preconditions | Student has to join the respective class |
| Data | Anonymous feedback |
| Stimulus | Feedback form |
| Response | Feedback submitted |
| Steps | 1. Offer feedback to the professor |
| Comments | Student cannot offer feedback if not enrolled in class. |

Table.14 -Use case 14

# FUNCTIONAL REQUIREMENTS:

* + Student should be able to register and login to the website
  + Faculty should be able to register and login to the website
  + Faculty should be able to create new classes
  + Faculty should be able to create assignment with description
  + Faculty should be able to download the submitted assignment.
  + Faculty should be able to reward the assignment
  + Student should be able to join classes
  + Student should be able to view assignment created by faculty
  + Student should be able to provide anonymous feedback.
  + Student should be able to download study materials.

# NON-FUNCTIONAL REQUIREMENTS:

* + Understandability: The User interface has to be easy to understand and user friendly.
  + Speed: The system should be able to respond fast as the user should not have to wait for long time for the page to load.
  + Security: The information should be secured from unauthorised access

# Project Management Plan:

* 1. **Project Deliverables:**

|  |  |
| --- | --- |
| Project Deliverables | Date |
| Project Plan | 09-27-2022 |
| Software requirement specification | 10-11-2022 |
| Software Design Specification | 10-18-2022 |
| Initial version Software | 11-01-2022 |
| Test Plan | 11-08-2022 |
| Final Software Product | 12-08-2022 |
| Project Presentation and Demonstration | 12-06-2022 |

Table.14 project deliverables

# Architecture:

**4.1 Architecture Style used:**

In order to create this e-learning system, we employed layered architecture. Three interconnected layers make up the system. User interface layer, application service layer, and database layer are the three levels.

# 4.2 Architecture model:

The architectural model has three levels:

# User interface layer:

On the user's system, this software is functional. The user may see the services the app offers by using this feature. The user may choose choices, set up classes, log in, submit assignments, give feedback, etc. The application layer and user interface communicate with one other over the internet services.

# Application Layer:

The system operations and business logic are handled in the most crucial layer, the application service layer, which is also known as the application layer. This layer houses the logic of the whole system. The system modules and the user interface layer both have access to the application service interfaces in between function calls. According to the top layer's service request, the application service layer also makes database modifications.

# Database layer:

The last layer is the database layer which holds data and information like user details, assignments, documents etc.

# 4.3 Architecture diagram

Diagram

Description automatically generated

Fig2 Architecture diagram

# Design Specification

* 1. **Design goals and objectives:**

The goal of this project is to give users a simple and user- friendly GUI so they may post notes, download documents, create classes, provide anonymous feedback, and manage all the class activities. Additionally, the administrator has the ability to review user-uploaded notes and decide whether to approve or reject them. The main points are:

* + - Simplified Management of all
    - Uploads of notes and assignments.
    - Reward assignments
    - Join classes
    - Create Classes
    - Create Assignments
    - Anonymous feedback
    - Notifying user when faculty downloads his documents

# Statement of scope:

* + - 1. The user(student) needs to know if faculty has viewed his documents
      2. Enhancements learning platforms requires enhancement like, anonymous feedback system,

# Software context:

The product will eventually be used concurrently by many people. Concurrent connections will thus cause problems during implementation. Additionally, this is a trial product that, if it is successful, may also be applied elsewhere. This raises concerns regarding the support for a greater user base in the future.

# Major constraints:

Time: We have little over weeks to finish all the documentation, software development, and upgrades. We have quite a lot of ideas, but we lack the time to put them into execution.

# Data Design:

**5.2.1 Database description:**

User Profile:

User\_ID, User\_Name, Password, Email, Contact\_Number, Role.

Assignment Entities:

Assignment\_Name, Description, Class\_Code, Posted\_Date, Submission\_date, File\_Path.

Notes Entities:

Document\_Name, Class\_Code, Count\_of\_Students\_viewed, File\_path

Submission Records Entities:

Class\_code, Assignment\_Name, Submission\_Date, Grade, Comment, Viewed by\_Faculty

# Component level design:

**5.3.2 Architecture diagrams**

* + - Creating new class



Fig.3(a) Creating new class

* + - Creating course content and assignments



Fig.3(b) Creating assignments

* + - Student access

Chart, box and whisker chart

Description automatically generated

Fig.3(c) Creating student access

# Description for Components

**5.4.1 Component and description**

* + - Signup:

Action: The signup page appears. The user needs to enter name, University ID number, email, role(student/faculty), password to register and create a new account.

* + - Login:

Action: The login page appears. The user will enter username in username field,password in password field and click on login. User will be logged in if it’s a valid username andpassword pair.

* + - Create New class:

Actions: For the Faculty module, create class button appears in order to create a new class. On clicking create class, Create class page with fields appears. This page contains Class\_name, branch and code to create a class.

* + - Create Assignment:

Actions: The faculty enters the class, clicks on create assignment button to create a new assignment.

* + - Uploading assignment:

Actions: The Student enters the class and clicks on upload assignment. On clicking upload assignment, Upload assignment page appears with Name, Description and filed field to upload the document.

* + - Viewing and grading the uploaded assignment document:

Actions: The faculty views the document and click on reward to grade the assignment submitted.

# Design Models

* + - **Sequence Diagram**

A picture containing chart

Description automatically generated

Fig 4: Sequence diagram

Diagram

Description automatically generated

Fig 5: Sequence diagram 2

# Activity diagram:

Diagram

Description automatically generated

Fig 6: Activity diagram

# User interface design

The new interface will be simple to use and will appeal to system users. A graphical user interface with menus at its base will be included. It will be up to the users to make the right decisions or enter the required data. The user interface design is as follows:

* + - Login Page:

Graphical user interface, website

Description automatically generated

Fig 7: login portal

* + - Student Sign up:

Graphical user interface

Description automatically generated

Fig 7: Student sign-up

* + - Faculty Sign up:

Graphical user interface

Description automatically generated

Fig 8: Faculty Sign-up

* + - Create assignment

Background pattern

Description automatically generated

Fig 9 : Activity diagram

* + - Upload assignment

Graphical user interface

Description automatically generated with low confidence

Fig 10: assignment upload

* + - Give anonymous feedback

Background pattern

Description automatically generated

Fig 11: Anonymous feedback

* + - Reward assignment

Graphical user interface, application, Word, PowerPoint

Description automatically generated

Fig 12: Reward assignment

* + - View anonymous feedback

Graphical user interface, application

Description automatically generated

Fig 13: view anonymous feedback

# 5.7 Interface design rules

* + - Designing interfaces between software modules.
    - Designing interfaces between humans (or users) and computers.
    - Designing interfaces between software and non-human producers and consumers of information.
    - Easy to Learn, Easy to Read, Easy Navigation Between Interfaces

# 5.8 Restrictions, limitations, and constraints

* The main restriction or constraint for our project so far is time, as we only have around three months to complete it. It is crucial that we observe the. We invest time in each stage of the software development process. We might have offered many other features, such as an online assistance menu, but time denies us the ability to do so.
* Design expertise is another constraint. Although it does not affect the project as much as time does,it does prevent us from adding more to the projects.

# Test Management

* 1. **Test Objectives:**

Testing is one of the most important aspects of computer programming triggers; without working programming, the system would never produce the output for which it was designed. Testing works best when user development is asked to help identify all errors and bugs. The sample data is used for testing purposes. It is the quality of the data used, not the quantity, that is important in testing. Before executing live operation commands, testing ensures that the system was correctly and efficiently configured.

The primary goal of testing is to discover a slew of error in a systematic and time efficient manner. Formally, testing is the process of running a program with the goal of finding errors.

* + - A successful test is one that reveals a previously unknown error.
    - A good test case is one that has a good chance of detecting an error, if one exists.
    - The test is insufficient for detecting potentially present errors.
    - The software mostly confirms to quality and dependability standards. Main features to be tested:
      * Notifying student when faculty downloads his/her work.
      * Join classes.
      * Anonymous feedback
      * Upload notes
      * View and download notes

# Scope:

This phase will cover all important requirements. In the testing phase, team members work on:

1. Constructing a manual test with all required stages
2. Verifying and be able to inspect it while doing the test
3. Add the outcomes and pertinent information
4. Reviews outcomes.

# Test Approach:

* + 1. **Levels of testing:**

# Code Testing:

This investigation the program’s logic. For example, the logic for updating various sample data, as well as the logic for working with sample files and directories, was tested and verified.

# Specification Testing:

Starting with what the software should do and how it should operate under various circumstances, this specification begins to be put into action. All the modules have test cases for various scenarios and combination of criteria.

# Unit Testing:

Each module is individually tested during unit testing, then the results are integrated with the total system. Unit testing concentrates verification efforts on the module’s tiniest unit of software design. Additionally known as module testing, this is. Each system module is tested independently. This testing is done right together with the code. Each module is found to function satisfactorily in the testing phase with relation to the module’s anticipated output. There are also certain field validation tests. For instance, the correctness of the data entered is verified by alerting the user. Finding system errors is quite simple.

Each modules can be tested using the following two strategies:

* + Black box testing
  + White box testing

# Test Cycles:

* There will be functional testing cycles. All of the script will be run once every cycle.
* The goal of the first cycle is to find any blocking issues, critical flaws, and the majority of high-level flaws. To access all of the scripts, it is anticipated that a workaround will be used.
* The second cycle’s goals are to find any remaining high and medium faults, eliminate the workaround from the first cycle, fix scripting errors, and get overall performance.

# Test Cases:

Test cases for functionalities:

|  |  |
| --- | --- |
| ID | EL001 |
| Test Input | Clicking on the student’s work |
| Expected Output | Download the file, change status in student’s portal |
| Description | When the faculty click on the student’s work, the file gets downloaded and the notification will be sent to the student. |

Table 16 : Test case 1

|  |  |
| --- | --- |
| ID | EL002 |
| Test Input | Click on anonymous feedback |
| Expected Output | The feedback field should be displayed |
| Description | When the student clicks on anonymous feedback, the feedback field should be displayed where he/she can write their feedback |

Table 17 : Test case 2

|  |  |
| --- | --- |
| ID | EL003 |
| Test Input | Enter valid credentials and click on login |
| Expected Output | Login should be successful |
| Description | When the user enters valid credentials and clicks on login, the user should be able to login to the website successfully. |

Table 18 :Test case 3

|  |  |
| --- | --- |
| ID | EL004 |
| Test Input | Click on create class |
| Expected Output | Create class page should be displayed with all the fields |
| Description | When faculty clicks on create class button, create class page with all the fields should be displayed. |

Table 19 :Test case 4

|  |  |
| --- | --- |
| ID | EL005 |
| Test Input | Click on view assignments. |
| Expected Output | View assignment page should be displayed. |
| Description | When user clicks on view assignments button, view assignment page should be displayed. |

Table 20 :Test case 5

|  |  |
| --- | --- |
| ID | EL006 |
| Test Input | Click on give anonymous feedback and give feedback |
| Expected Output | Feedback should be listed on faculty’s feedback portal |
| Description | When student gives anonymous feedback , the Faculty should receive feedback from student. |

Table 21 :Test case 6

|  |  |
| --- | --- |
| ID | EL007 |
| Test Input | Submit assignment |
| Expected Output | Assignment submitted successfully confirmation popup should be  displayed. |
| Description | When student uploads the assignments and click on submit assignment, successful confirmation pop up should be displayed. |

Table 22 :Test case 7

# Traceability of test cases to use cases:

Up to 90% of the test cases and the previously indicated use cases match. All the test cases mentioned are developed and tested. Anyone can use this application with any required prior knowledge about the application.

# Test results and Assessments:

|  |  |
| --- | --- |
| Testcase ID | 001 |
| Test Case description | Login to the application |
| Steps | 1.Launch the application 2.Enter valid login details  3.Click on login |
| Data | Username, password |
| Expected Output | Login to the website |
| Actual Result | Login to the website |

Table 23 :Test report case 1

|  |  |
| --- | --- |
| Testcase ID | 002 |
| Test Case description | Create Class |
| Steps | 1.Login the application as faculty 2.Click on create class  3.Enter the details and create |
| Data | Course, Branch, class code |
| Expected Output | A new class should be created |
| Actual Result | A new class is created. |

Table 24 :Test report case 2

|  |  |
| --- | --- |
| Testcase ID | 003 |
| Test Case description | Join class |
| Steps | 1.Login the application as student 2.Enter the class code  3.Click on join class |
| Data | class code |
| Expected Output | Student should be able to join the class. |
| Actual Result | Student can join the class. |

Table 25 :Test report case 3

|  |  |
| --- | --- |
| Testcase ID | 004 |
| Test Case description | Give anonymous feedback |
| Steps | 1.Login to the application as student 2. Select the class  3.Click on the give anonymous feedback button 4. Provide feedback   1. logout 2. login to the application as faculty 7. click on view feedback |
| Data | Provide feedback |
| Expected Output | Feedback should be listed in faculty’s feedback portal |
| Actual Result | Feedback is listed on faculty’s portal. |

Table 26: Test report case 4

|  |  |
| --- | --- |
| Testcase ID | 005 |
| Test Case description | Download assignment |
| Steps | 1.Login to the application as student 2. Select the class  3.Click on view assignment 4. Submit assignment  5. logout   1. Login as faculty 2. click on view assignment 8. Download the assignment |
| Data | Click on download assignment |
| Expected Output | Faculty should be able to download the file |
| Actual Result | Faculty can download the file |

Table 27: Test report case 5

# Conclusions

* 1. **Outcomes of the project**

Graphical user interface, website

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generatedBackground pattern

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generated

Background pattern

Description automatically generatedGraphical user interface, application

Description automatically generatedBackground pattern

Description automatically generated with medium confidence

# Lessons learned

We have gained knowledge about software engineering during the course of this project. Various software ideas, its architecture and types of architectural models, design, various representations of a single software, and most importantly, time management and the process by which a software is created and the stages involved in creating a software. We also learnt about how to develop UML diagram using various tools.

# Future enhancements

This website can be adapted for any level of education. This application can further be enhanced with an effective user interface and various features like random automated group assignment, multiple attempts to submit assignments can be implemented to improve the overall experience.

30