MULTIMEDIA UNIVERSITY

Software Requirements Specification

for

Legendary Management System

Part 2

Group No.: **L (Legendary Industry)**

Lee Guang Shen	**1171103543**	**Project Leader**	**Sign Up and Login Management**
Mang Yu Jie	**1181101662**	**Programming Leader**	**Training Materials Management**
Low Min Xuan	**1181100653**	**Documentation Manager**	**Training Course Management**
Leim Jing Han	**1171103793**	**Quality Manager**	**User Feedback**

Table of Content

1.0 Problem Statements

- Have a communication barrier between trainers and participants.
- Physical systems are limited to the region where the participants are located.
- Existing training systems are having a complicating and confusing interface.

2.0 User Stories

- As a user, I want to have an easy user interface, so that the program is easy for me to use.
- As a trainee, I want to have a good training course so that I can improve myself.
- As a CEO, I want to have a good training management system so that I can maintain the effectiveness and understand the ability of my employees.
- As a trainer, I want to view the feedback from the trainees so that I can improve the course.
- As a Manager, I want the system to be able to add and delete the trainees.

3.0 Objectives

- To create, manage and carry out educational and training programs.
- To provide an easy access to information anywhere anytime without limitation.
- To allow the trainers to organise their training efficiently.

4.0 Elicitation Techniques

- User Story
- Literature Review

5.0 Software Requirements Specification (SRS)

5.1 Functional Requirements

5.1.1 Registration Management

R1: The system shall provide a login menu for the trainers and trainees.

R2: The system may be able to encrypt and decrypt the password of the user.

5.1.2 Training Materials Management

R3: The system should provide a function for the trainers to upload the training materials.

R4: The system should provide a function for the trainees to view the training materials.

5.1.3 Training Course Management

R5: The system shall let the trainers create courses.

R6: The system shall let the trainees to enroll in training courses.

R7: The system shall check the training registration and do not let the trainees register the training courses at the same time.

5.1.5 User Feedback

R10: The system shall provide a text box for users to fill up the feedback of the course.

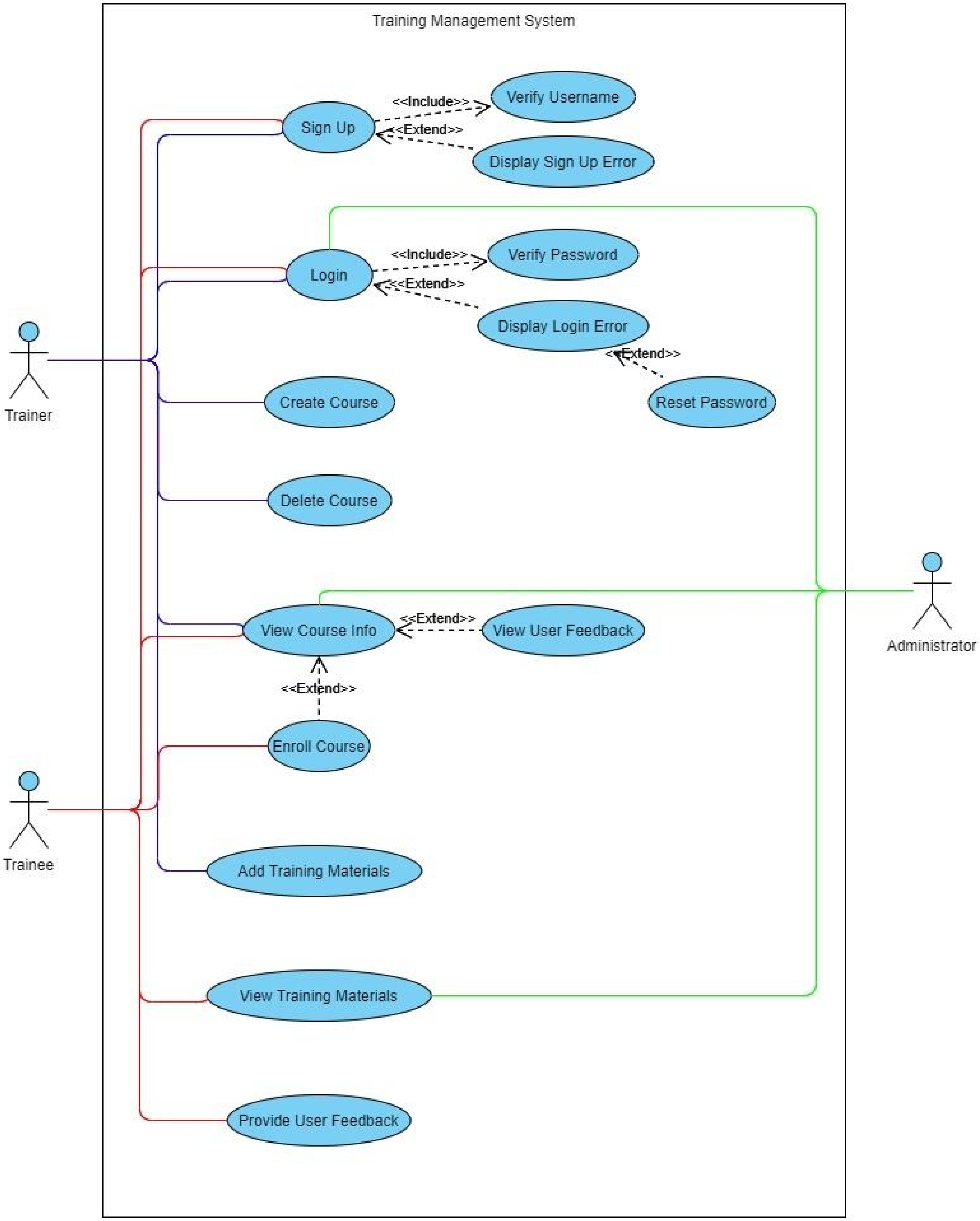
5.2 Non Functional Requirements

- An appropriate error message has to be shown to the user, if the system is unable to process certain requests.

6.0 Process Model

- Evolutionary Models - Prototyping

7.0 Use Case Diagram



8.0 Use Case Diagram Table (UCDT)

8.1 Registration Management

Use Case Name	**Sign Up Management**
Description	**Users are able to sign up to TMS**
Precondition	User must exist
Postcondition	User is successfully signed up
Main Flow	1. Users create their Username, Nickname, Email and Password. 2. The system will go through the database to check whether the Username is taken already or not. 3. Once the Username provided is unique and not yet been used, the user is allowed to use the username. 4. Account is created and user is successfully signed up
Alternate Flow	2a. Username is taken
Exception Flow	3a. If the Username provided is already taken, the system will display an error message to the User and require the User to create another Username.

Use Case Name	**Login Management**
Description	**Users are able to log in to TMS**
Precondition	User must exist
Postcondition	User is logged in
Main Flow	1. User enters their Username and Password. 2. The system will go through the database to find the matching username. 3. System validates the combination of the Username and Password by matching the input of the user and value in the database. 4. Once the Username and Password are matched, user is allowed to log in to the TMS 5. User is logged in
Exception Flow	4a. Once the Username and Password are mismatched, the system will display an error message to the user 4b. The user has to enter the Username and Password again

8.2 Training Materials Management

Use Case Name	**Upload Training Materials Management**
Description	**Trainee/Trainer (Users) are able to access the materials in TMS**
Precondition	Users must login as a trainer
Postcondition	Trainers are able to upload the materials in TMS.
Main Flow	1. User login into the TMS. 2. User clicks on the “View Course” button. 3. User clicks on the “Upload Material” button. 4. User could fill in the details of materials to the TMS 5. User click the “Upload” button 6. System uploads the details to the database.
Exception Flow	6a. Trainer upload fail due to the network problem

Use Case Name	**View Training Materials Management**
Description	**Trainee/Trainer (Users) are able to access the materials in TMS**
Precondition	Users must be enrolled in the specific course
Postcondition	Users are able to view the materials in TMS.
Main Flow	1. User login into the TMS. 2. User clicks on the “Course” button. 3. User clicks on the “View Training Materials” button. 4. User could view the materials from TMS
Exception Flow	4a. User view training materials fail due to the network problem

8.3 Training Course Management

Use Case ID	**Create Course Management**
Description	**System is able to let the trainer create courses in TMS.**
Precondition	Trainers have signed up for an account and logged in TMS.
Postcondition	Trainer successfully created the course in the TMS.
Main Flow	1. Trainer log in to TMS. 2. Trainer clicks on the “My Course” button. 3. Trainer clicks on the “Add Course” button. 4. Trainer enters the details of the course that to be created. 5. Trainer clicks on the “Submit” button when done. 6. The newly added course and its details will be stored in the system’s database. 7. Trainer successfully created the course in the TMS.
Alternate Flow	2a. Trainer can view the courses that have been created. 2b. Trainer is able to delete their course. 5a. If the course is created successfully, the system will display a notice to notify the trainer.

Use Case ID	**Enroll Course Management**
Description	**System is able to let the trainees enroll courses in TMS.**
Precondition (optional)	Trainees have signed up for an account and logged in TMS.
Postcondition	Trainees successfully enroll courses in TMS.
Main Flow	1. Trainees log in to TMS. 2. Trainees click on the “ Course” button to view available courses in TMS. 3. Trainees choose a course to enroll. 4. Trainees successfully enrolled in a course.
Alternate Flow (optional)	3a.Trainee can view course details before enrolling a course.

8.5 User Feedback Management

Use Case Name	**Submit user feedback**
Description	**Trainees are able to view the progress of the course they enrolled in.**
Precondition	Users enrolled in at least 1 course.
Postcondition	Users give feedback based on the course user enrolled.
Main Flow	Users select give feedback. Users fill the box and click the submit button to submit the feedback.
Alternate Flow	

Use Case Name	**View User feedback**
Description	**Administrator/Trainer (Users) are able to view the progress of the course they created.**
Precondition	Users must exist.
Postcondition	Users can view the feedback when clicking into course details.
Main Flow	1. User click view course details. 2. Then, select view user feedback. 3. System will show all the feedback to the user.

9.0 Business Rules

1. Each administrator can only own one and only one account.
   Each account can only be owned by one and only one administrator.

2. Each trainer can only register one and only one account.
   Each account can only be registered by one and only one trainer.

3. Each trainee can only register one and only one account.
   Each account can only be registered by one and only one trainee.

4. Each trainer may upload more than one training material.
   Each training material can only be uploaded by one trainer.

5. Each trainee may view more than one training material.
   Each training material may be viewed by more than one trainee.

6. Each trainee may submit many feedbacks.
   Each feedback can only be submitted by one trainee.

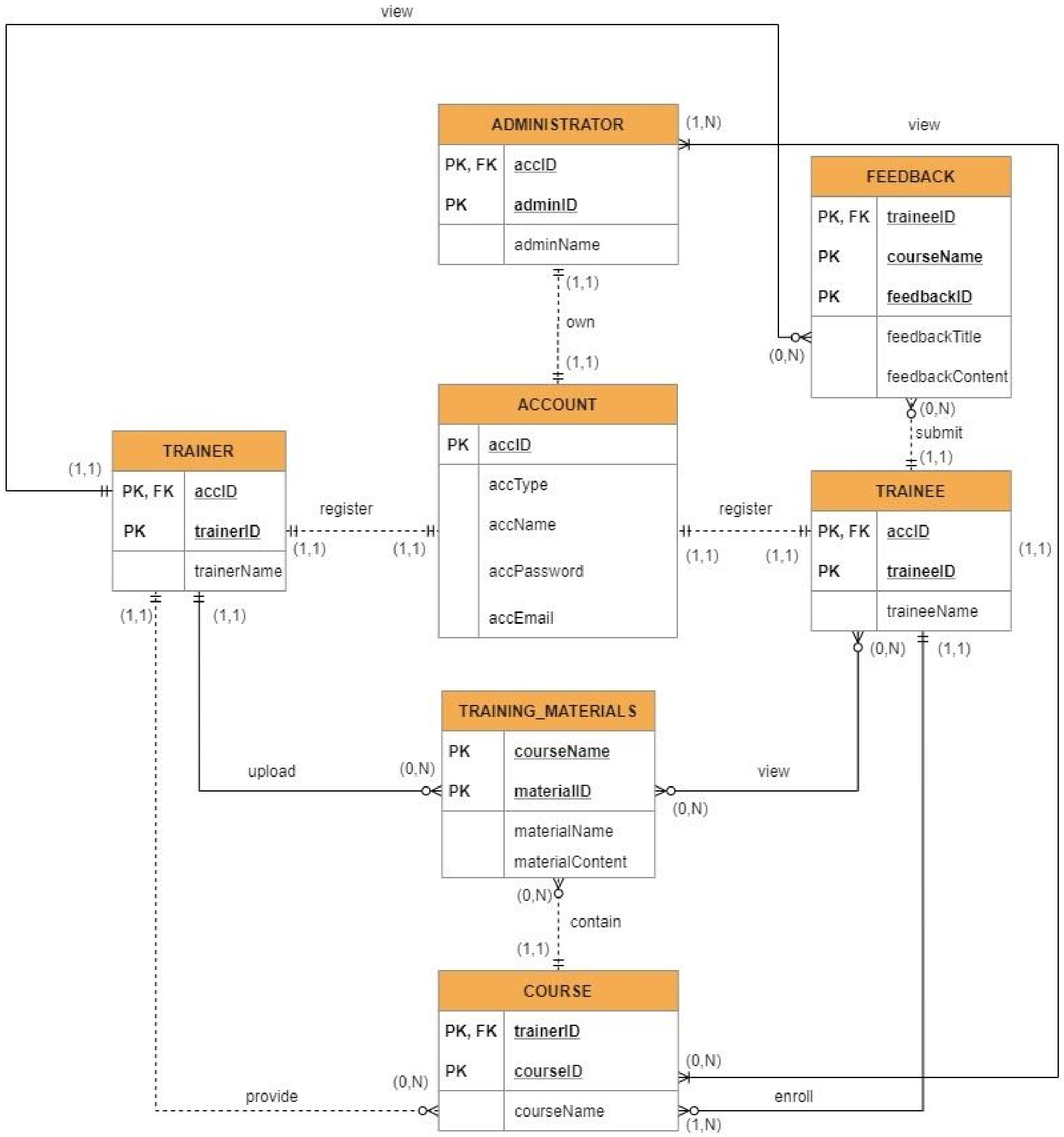
7. Each trainer may provide three courses.
   Each course can be provided by one and only one trainer.

8. Each trainee may enroll in more than one course.
   Each course can be enrolled by more than one trainee.

9. Each administrator can view all the course information available.
   The course information can be viewed by more than one administrator.

10. Each course may contain more than one training material.
    Each training material may be uploaded to one and only one course.

10.0 ERD Diagram



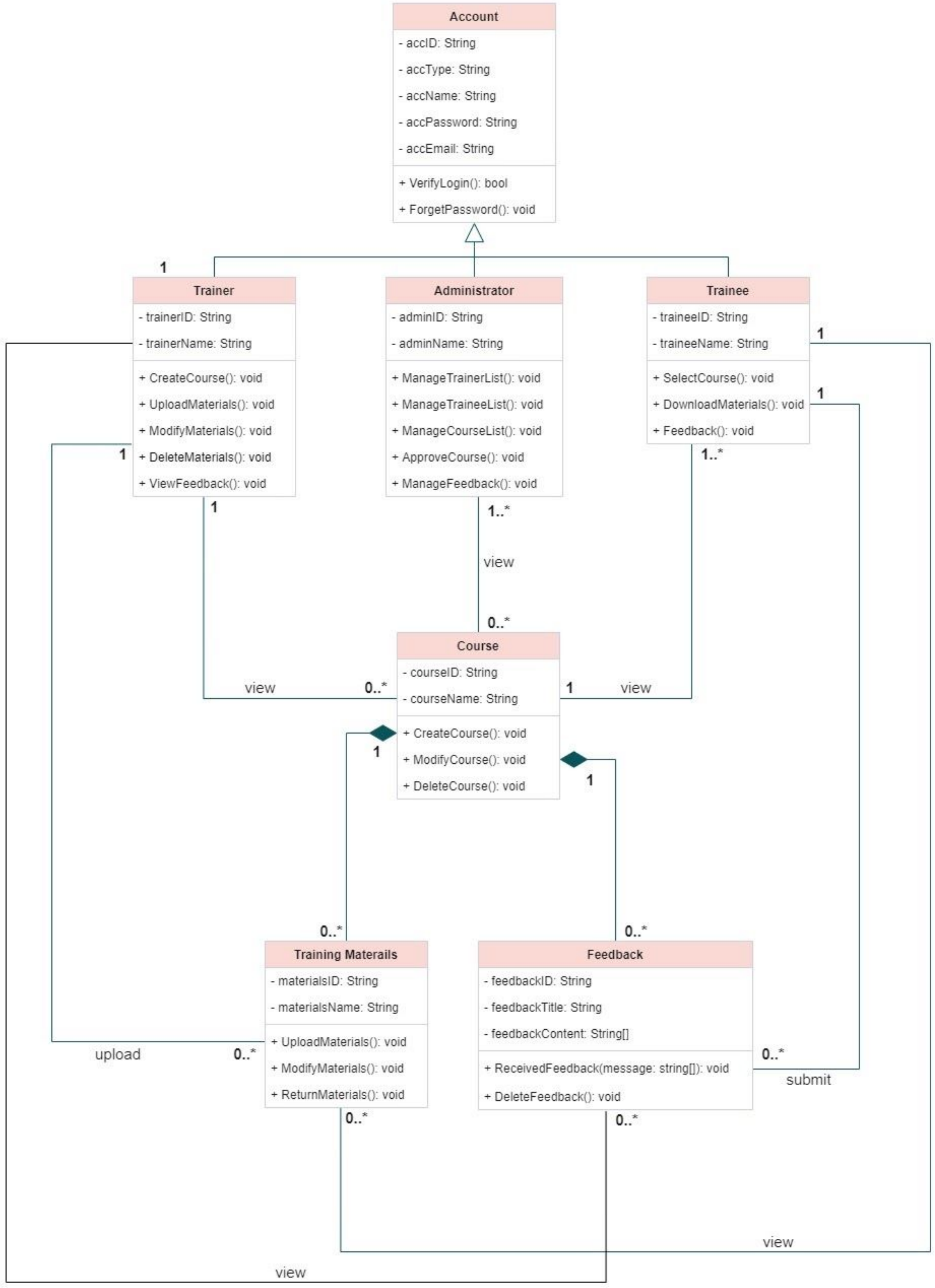
11.0 Data Dictionary

Entity Name	Entity Description	Attribute Name	Attribute Description	Date Type	Field Size	PK/FK
Account	An account that will be owned by a user.	accID	Unique identification of the account of the user.	Varchar	15	PK
		accType	The type of the account.	Varchar	8	
		accName	The name of the user.	Varchar	50	
		accPassword	The password of the account.	Varchar	20	
		accEmail	User's email binded to the account.	Varchar	50	
Administrator	A user who will use the system.	accID	Unique identification of the account of the user.	Varchar	15	PK, FK
		adminID	Unique identification of the user.	Int	-	PK
		adminName	The name of the user.	Varchar	50	
Trainer	A user who will use the system.	accID	Unique identification of the account of the user.	Varchar	15	PK, FK
		trainerID	Unique identification of the user.	Int	-	PK
		trainerName	The name of the user.	Varchar	50	

Trainee	A user who will use the system.	accID	Unique identification of the account of the user.	Varchar	15	PK, FK
		traineeID	Unique identification of the user.	Int	-	PK
		traineeName	The name of the user.	Varchar	50	
Course	Course to be provided by the Trainers	trainerID	Unique identification of the user.	Int	-	PK, FK
		courseID	Unique identification of the course.	Varchar	-	PK
		courseName	The name of the course.	Varchar	50	
Training_Materials	Training materials of the training courses.	courseName	Unique identification of the course.	Varchar	-	PK
		materialID	Unique identification of the training materials.	Varchar	-	PK
		materialName	The name of the training material.	Varchar	50	
		materialContent	The content of the training material.	Varchar	5000	

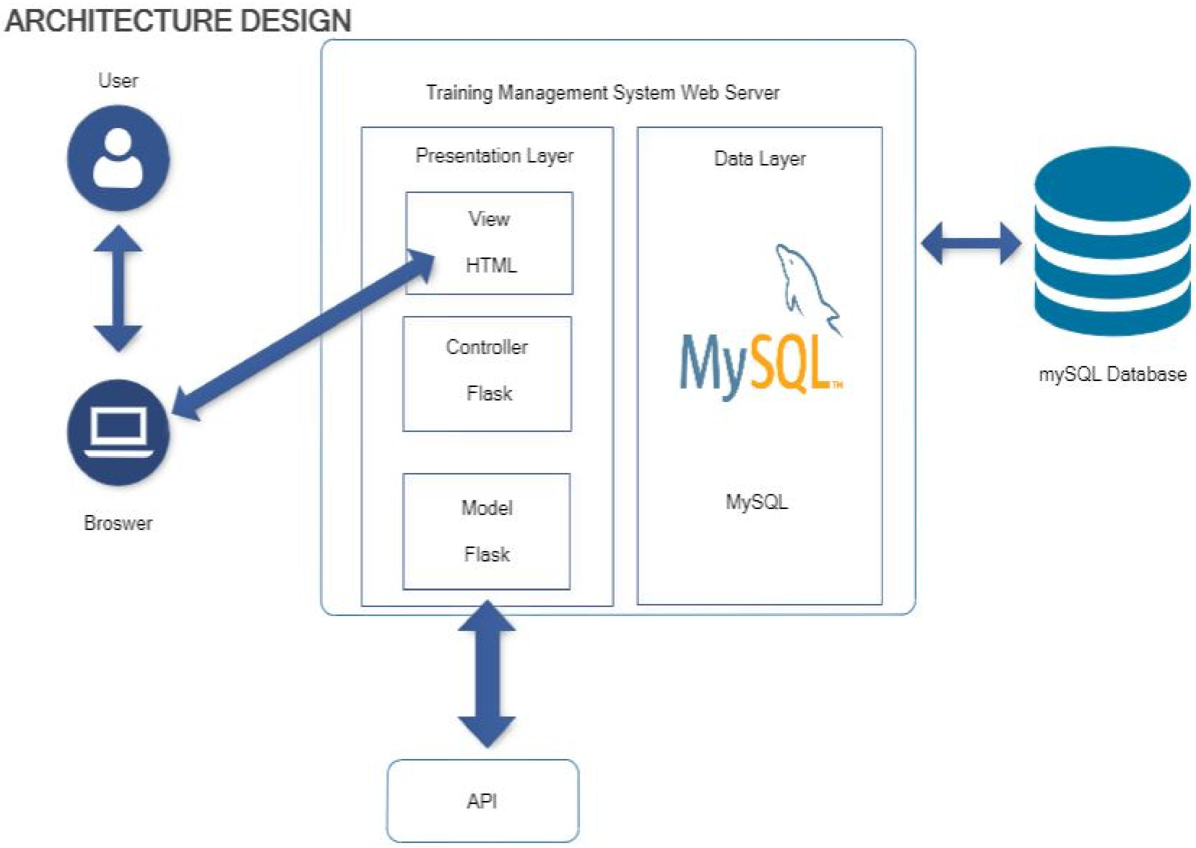
Feedback	The user feedback from the trainee.	traineeID	Unique identification of the user.	Int		PK, FK
		courseName	The name of the course.	Varchar	50	PK
		feedbackID	Unique identification of the feedback.	Varchar	-	PK
		feedbackTitle	The title of the feedback	Varchar	50	
		feedbackContent	The content of feedback	Varchar	500	

12.0 Class Diagram



13.0 Architecture Diagram

13.1 Software Architecture Diagram



Based on our team's architecture design, the user will access the web browser which will request access to the web server. In the Training System Management web server, it consists of a presentation layer which divides into View, Controller and Model. Controller and Model are built with Flask, which is a micro web framework written in Python and it will invoke the Application Program Interface(API). View is built up with HTML Besides, the data layer is using MySQL to store data and it's connected with a database to store, analyze and retrieve data.

14.0 User Interface Design

14.1 Trainer Interface

14.1.1 Login and Sign Up

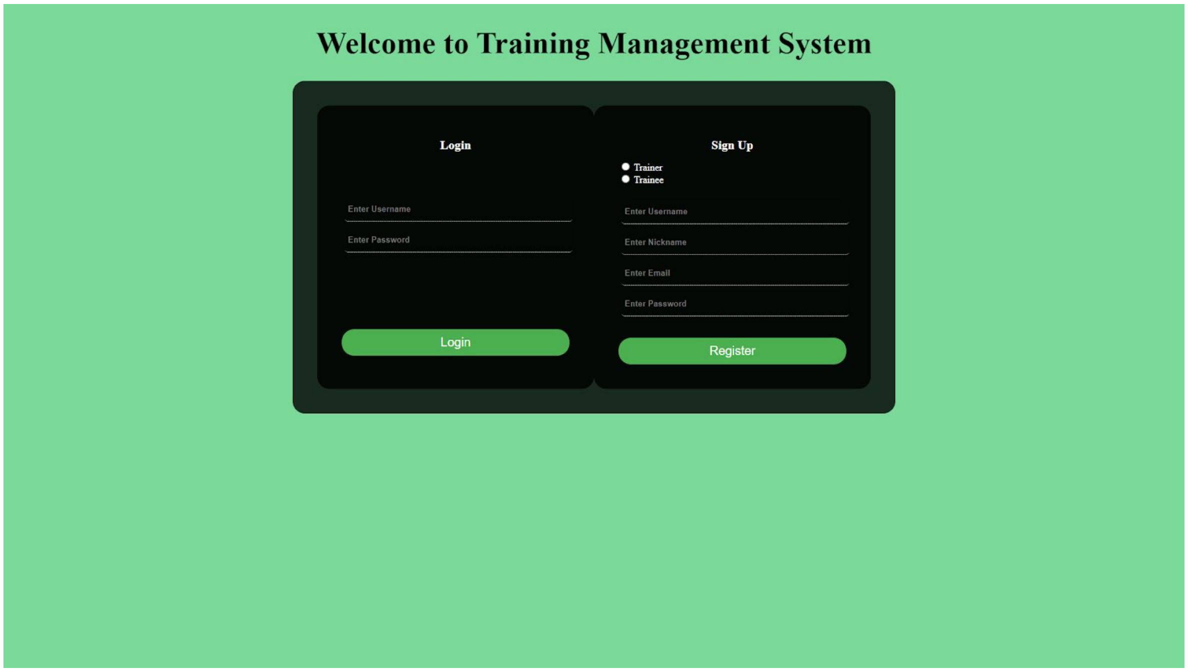


Diagram 1: Login and SignUp UI

The Login and Signup page is shown in Diagram 1 above. Trainers are able to login at the Login Session by entering their Username and Password that has been created earlier. If they don't have an account, they can register a new account by providing the required information at the Sign Up Session.

14.1.2 Homepage

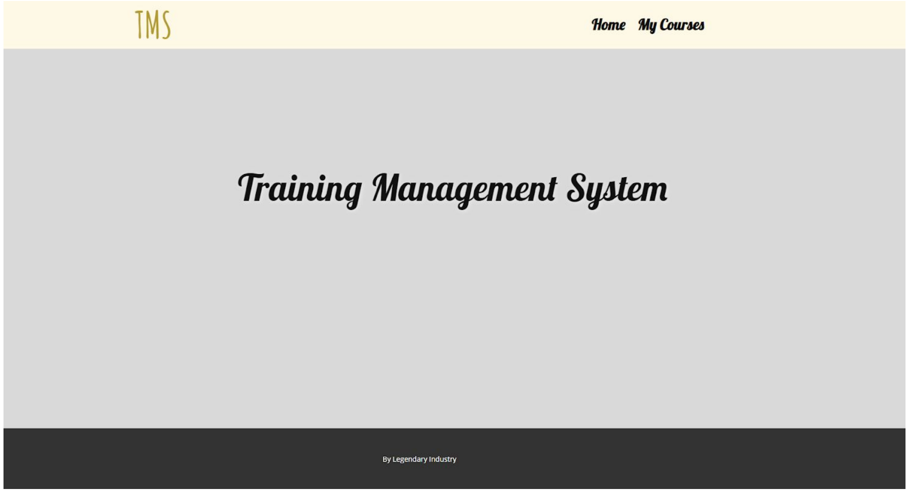
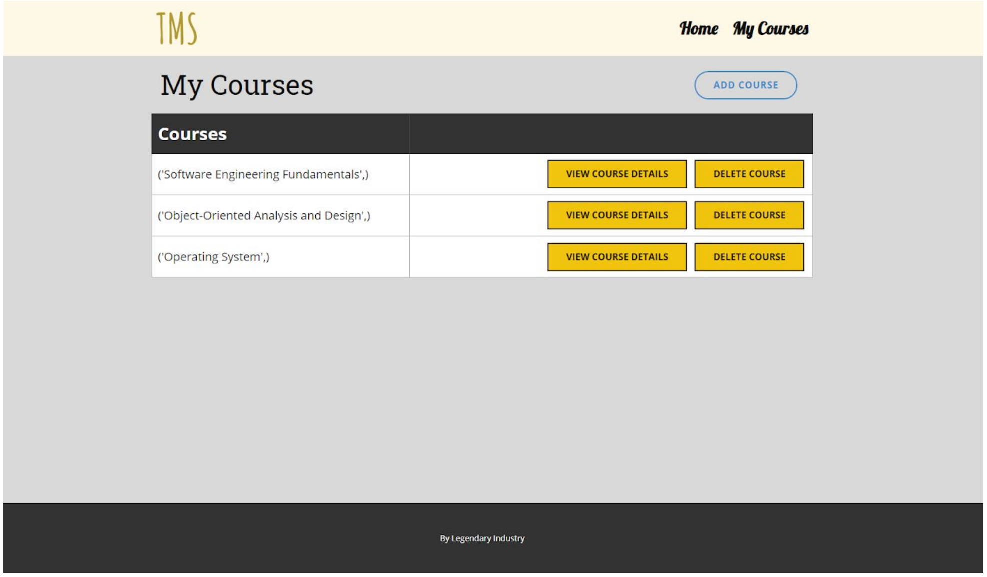


Diagram 2: Trainer's Homepage

After logging into the system, the Homepage is shown which contains a navigation bar on the right corner. It has "Home" to redirect back to this Homepage later on, and "My Courses" to navigate to their course to be created.

14.1.3 My Courses



Courses	
('Software Engineering Fundamentals',)	VIEW COURSE DETAILS DELETE COURSE
('Object-Oriented Analysis and Design',)	VIEW COURSE DETAILS DELETE COURSE
('Operating System',)	VIEW COURSE DETAILS DELETE COURSE

Diagram 3: Trainer's Course Page

After clicking on the "My Courses" at the navigation bar above, trainers are able to create courses by clicking on the "Add Course" button. An example of the course added "Software Engineering Fundamentals", "Object-Oriented Analysis and Design" and "Operating System" are shown in the Diagram 3 above.

14.1.4 Add Course

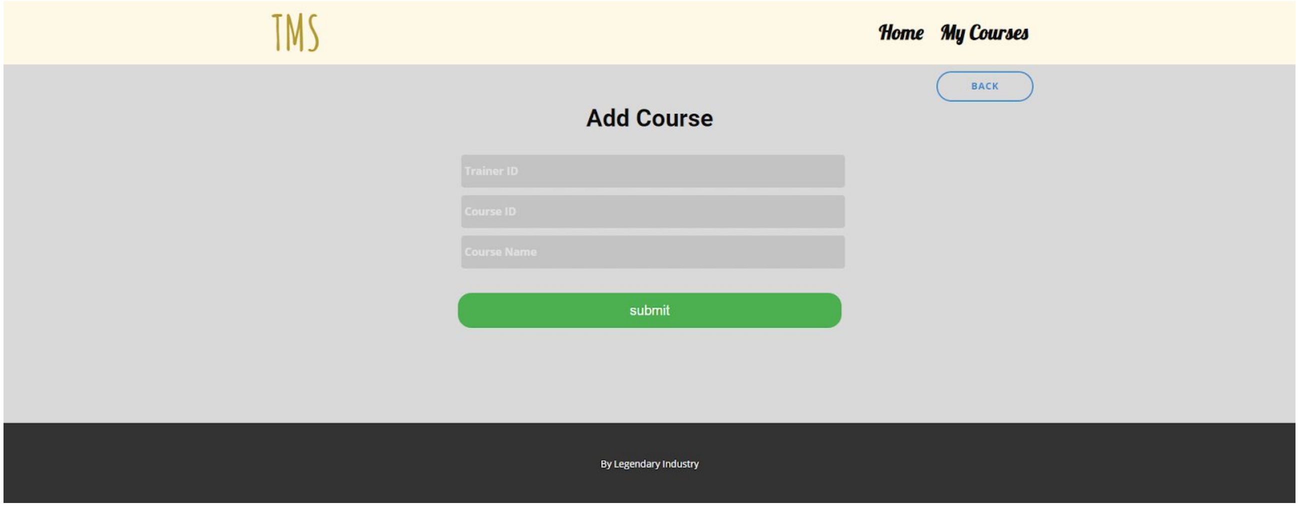


Diagram 4: Add Course Page

After clicking the “Add Course” button on Diagram 3, it will direct the trainer to the Add Course Page in Diagram 4. Trainers are able to fill in their TrainerID, CourseID and Course Name here.

14.1.5 View Course

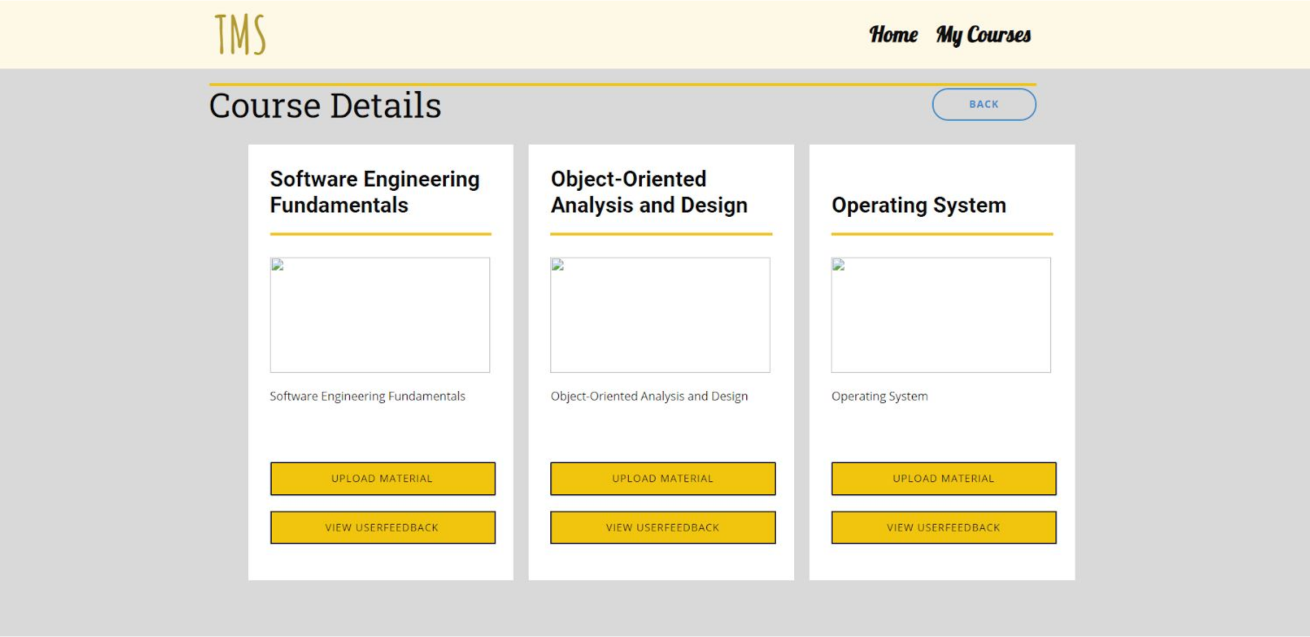


Diagram 5: View Course Page

When the course is successfully created, the trainer will be directed back to the Course Details that has an overview of all the courses created. The trainer then can click the “Upload Material” button shown in Diagram 5, to upload the corresponding training materials and “View UserFeedback” to view the feedback that has been provided by the trainees.

14.1.6 Upload Materials

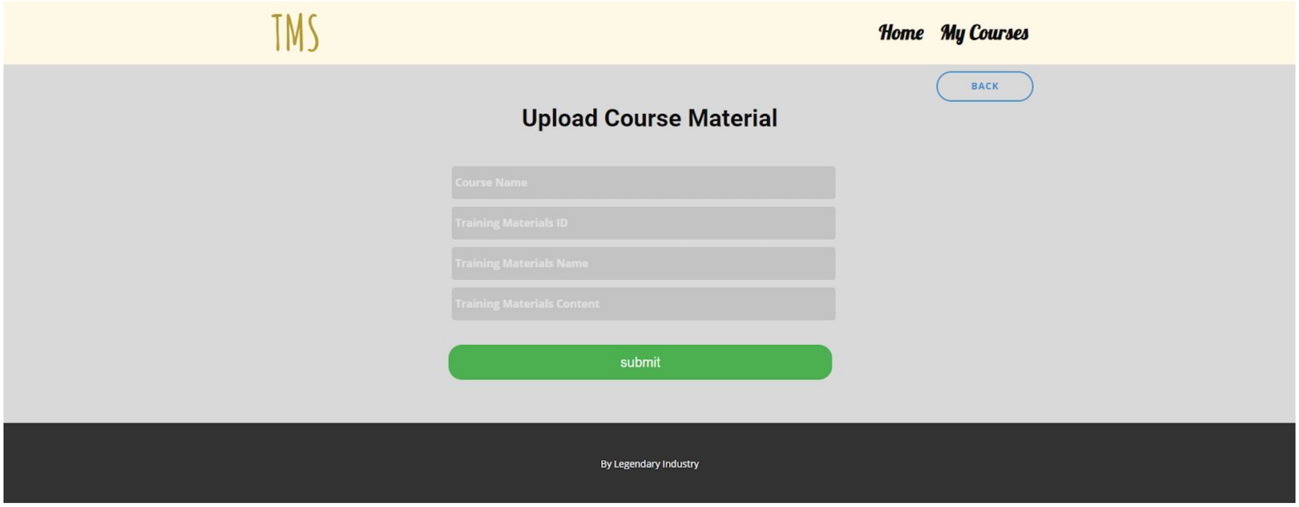
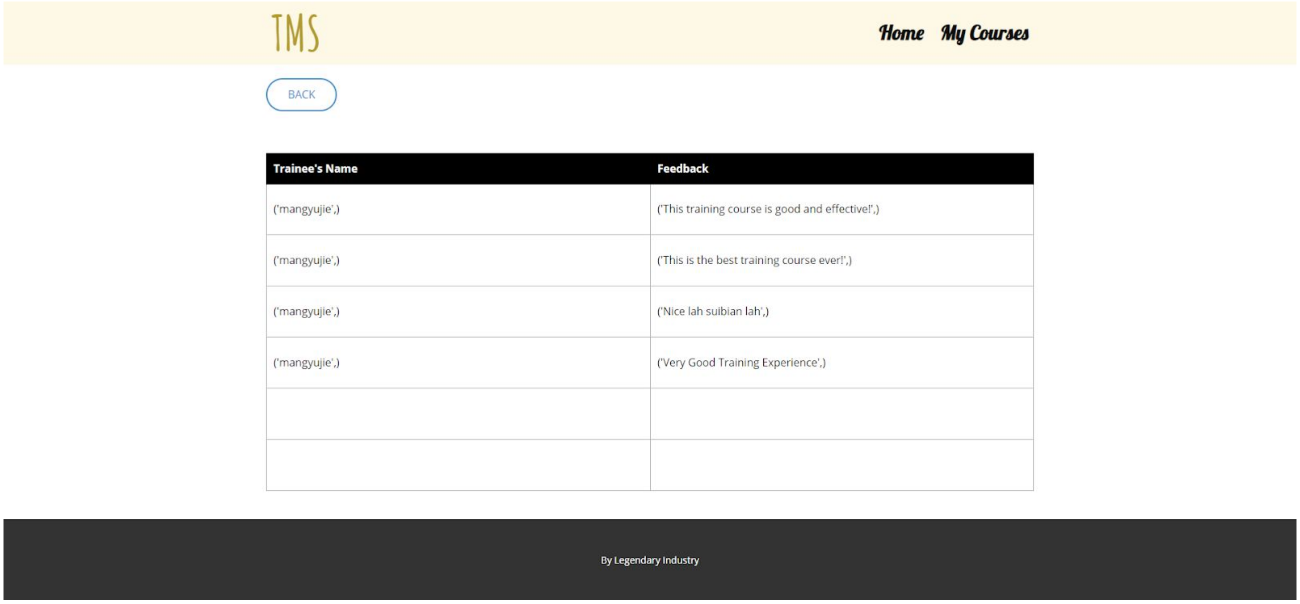


Diagram 6: Upload Materials Page

After clicking the “Upload Material” button shown in Diagram 5, trainers are able to name their training materials and upload it here in Diagram 6.

14.1.7 User Feedback



Trainee's Name	Feedback
('mangyujie',)	('This training course is good and effective!',)
('mangyujie',)	('This is the best training course ever!',)
('mangyujie',)	('Nice lah suibian lah',)
('mangyujie',)	('Very Good Training Experience',)

Diagram 7: User Feedback Page

From Diagram 5, when the trainer clicks the “View User Feedback” button, it will direct to the User Feedback Page as shown in Diagram 7.

14.2 Trainee Interface

14.2.1 Login and Sign Up

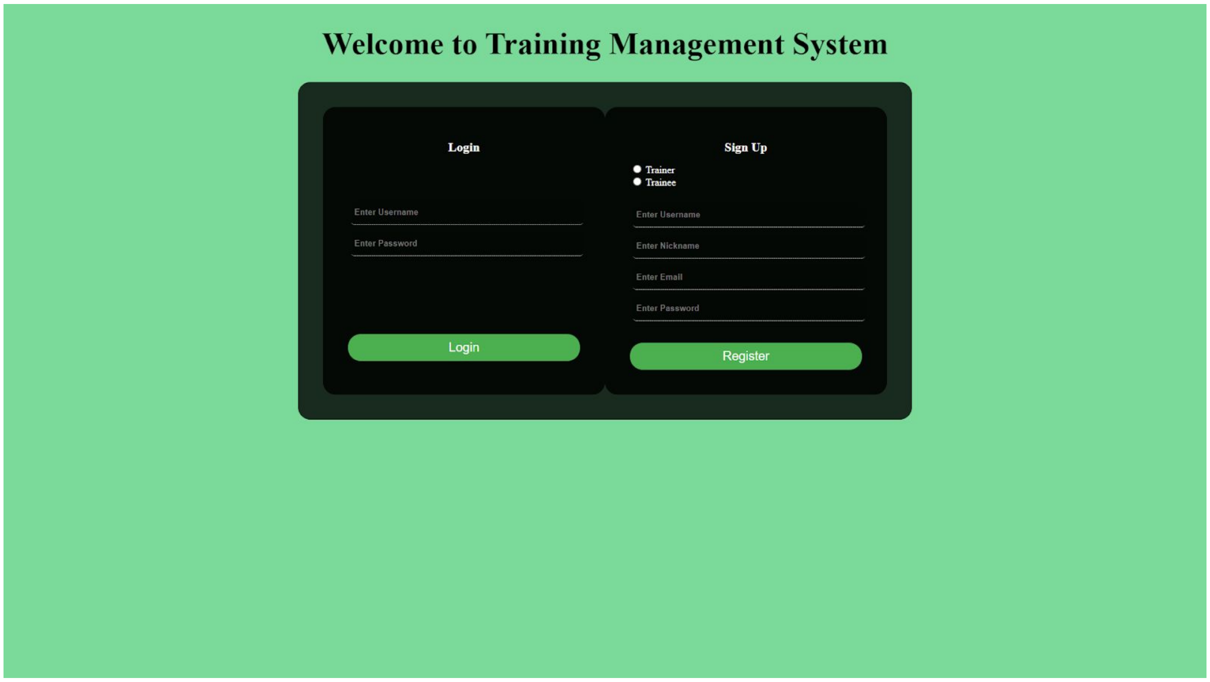


Diagram 8: Login and Sign Up UI

Similar to Trainer’s Interface, the Login and Signup page is shown in Diagram 1 above. Trainees are able to login at the Login Session by entering their Username and Password that has been created earlier. If they don't have an account, they can register a new account by providing the required information at the Sign Up Session.

14.2.2 Homepage

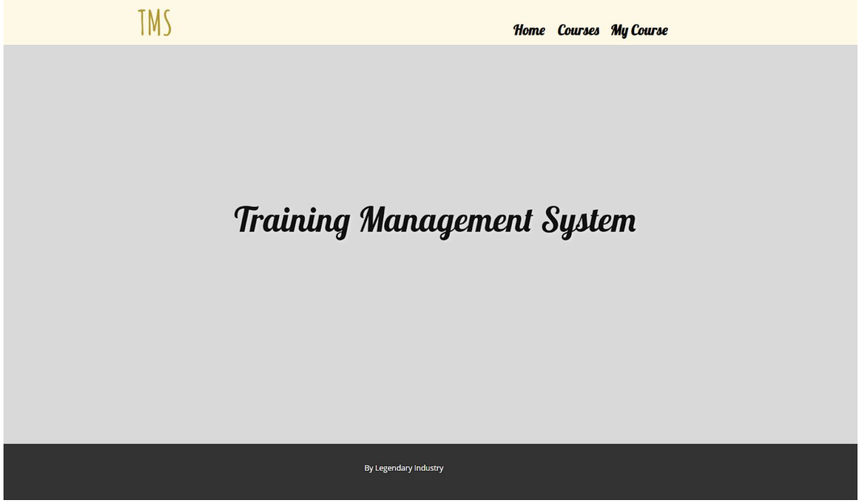


Diagram 9: Trainee's Homepage

After logging into the system, the Homepage is shown which contains a navigation bar on the right corner. It has "Home" to redirect back to this Homepage later on, "Courses" to navigate to the page that shows the courses available and ready to be enrolled and "My Courses" to navigate to the courses that they have enrolled.

14.2.3 Courses

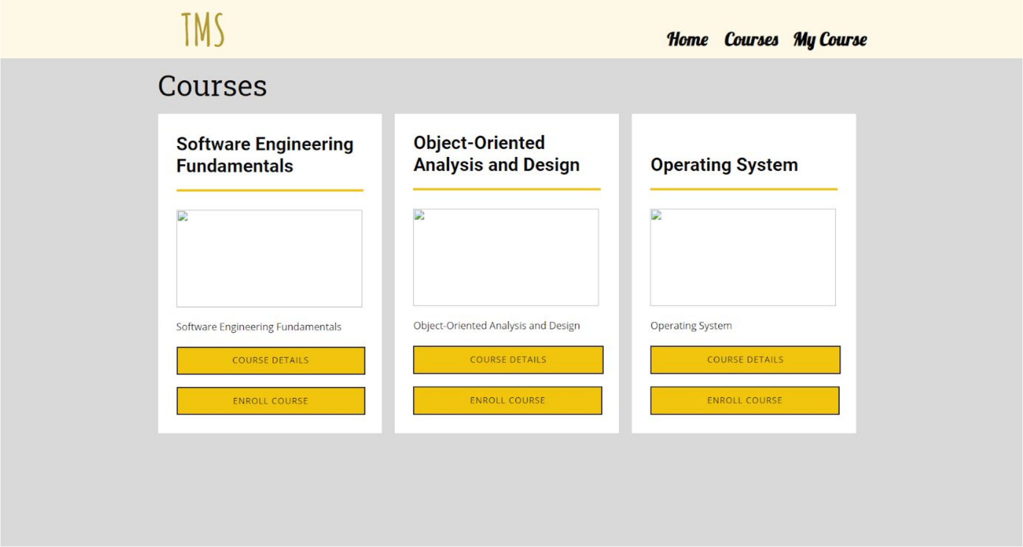
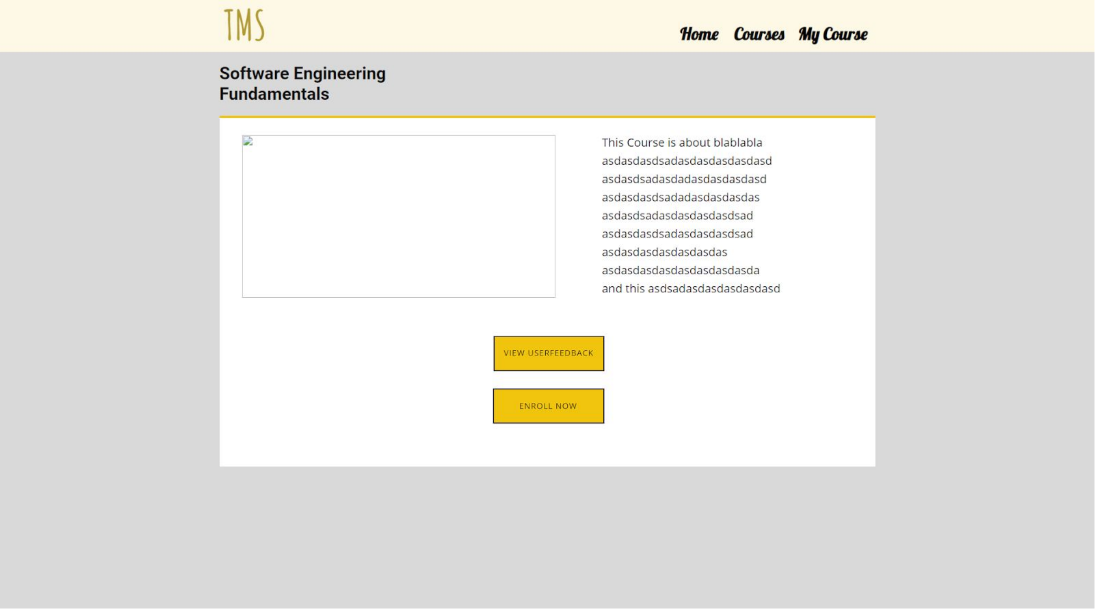


Diagram 10: View Courses Available

After clicking on the "Courses" button at the navigation bar, trainees are directed to the page as shown in Diagram 9. It shows the available courses to be enrolled in, they can either click the "Course Details" to show the details of the courses, or click "Enroll Course" to enroll the corresponding courses.

14.2.4 Course Details



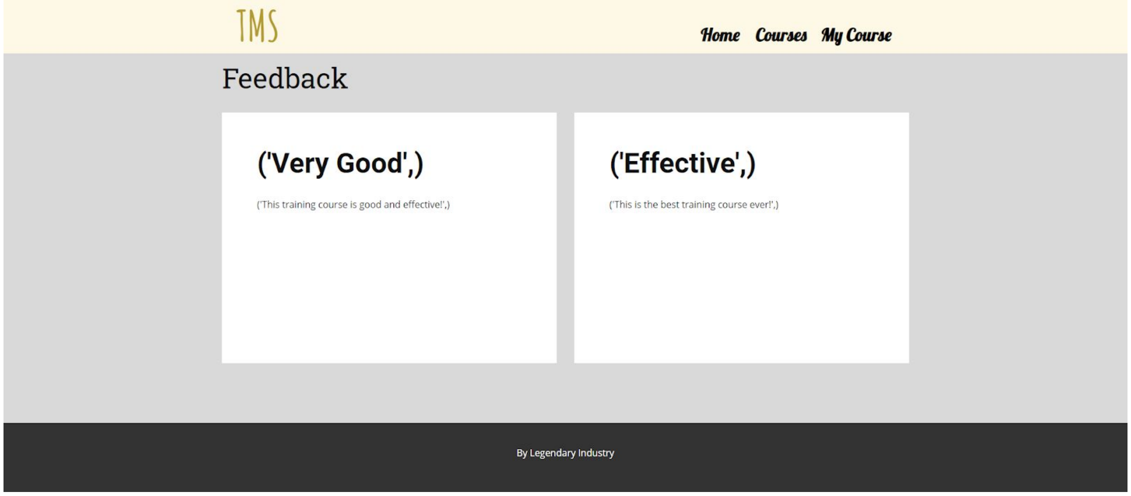
Software Engineering
Fundamentals

This Course is about blablabla
asdasdasdsadasdasdasdasdasd
asdasdsadasdadasdasdasdasd
asdasdasdsadadasdasdasdas
asdasdsadasdasdasdasdsad
asdasdasdsadasdasdasdsad
asdasdasdasdasdasdas
asdasdasdasdasdasdasdasda
and this asdsadasdasdasdasdasd

Diagram 11: Course Details Page

When the "Course Details" button is clicked in Diagram 10, the trainee will be able to see the details of the corresponding courses. If the trainee clicks the "View UserFeedback" button, it will direct them to Diagram 12, "Enroll Now" button to Diagram 13.

14.2.5 View Feedback



Feedback

('Very Good',)

('This training course is good and effective!',)

('Effective',)

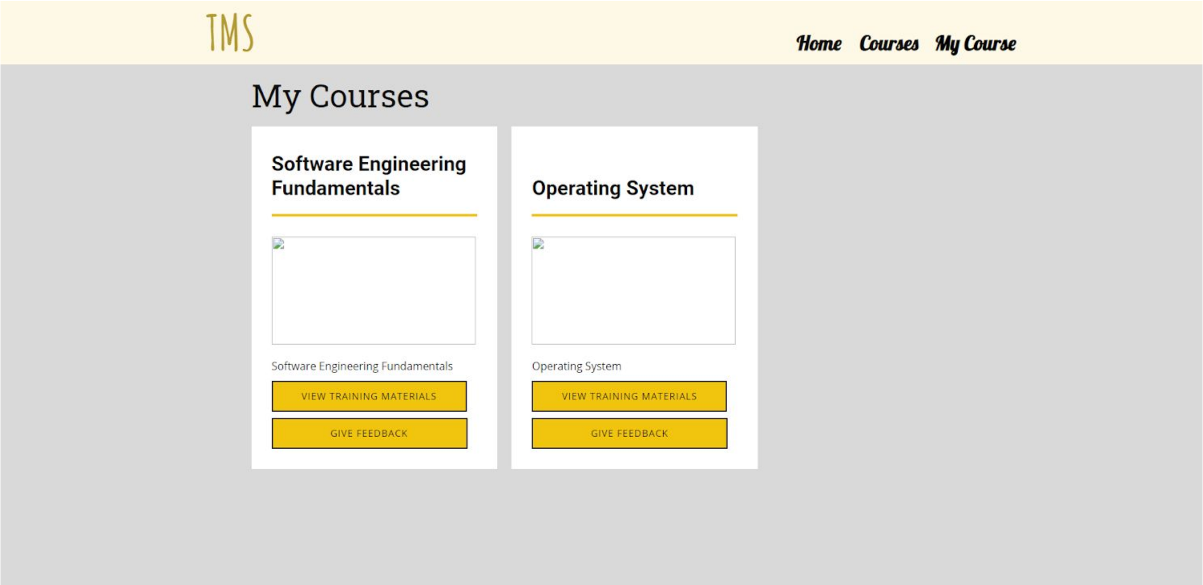
('This is the best training course ever!',)

By Legendary Industry

Diagram 12: View User Feedback Page

If the trainee clicks the "View Userfeedback" button in Diagram 11, it will show the user feedback that has been written by the previous trainees that have taken the course. New trainees can read them to help them to decide whether to enroll in the course or not.

14.2.6 My Courses



My Courses

Software Engineering Fundamentals

Software Engineering Fundamentals

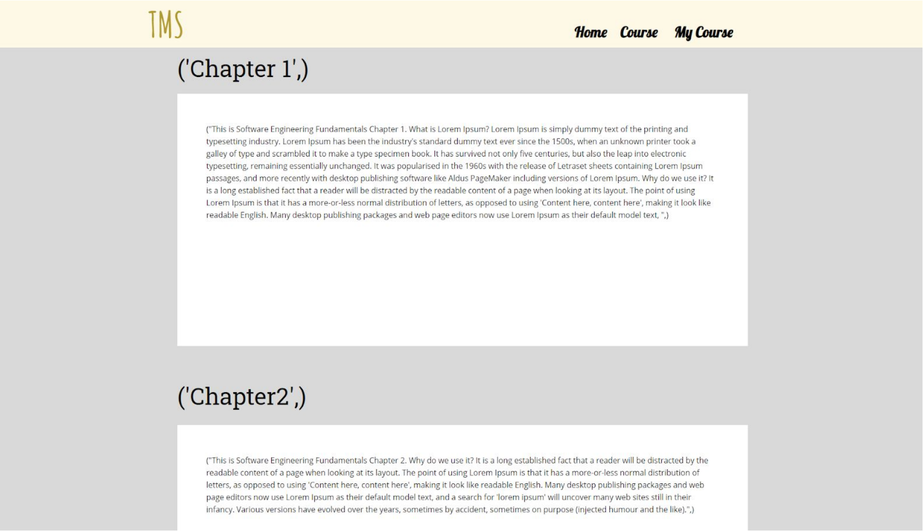
Operating System

Operating System

Diagram 13: View Trainee Enrolled Courses

Moreover, when the "My Course" button at the navigation bar is clicked, it will direct the trainee to the page as shown in Diagram 13. This is the page that is showing the enrolled course by the trainee. The trainee can view the training materials and provide their feedback towards the corresponding training course.

14.2.7 Training Materials



('Chapter 1',)

("This is Software Engineering Fundamentals Chapter 1. What is Lorem Ipsum? Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Why do we use it? It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, ",)

('Chapter2',)

("This is Software Engineering Fundamentals Chapter 2. Why do we use it? It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).",)

Diagram 14: Trainee's Course Training Materials

However, if the trainee clicks the "View Training Materials" button in Diagram 13, all the available training materials that have been uploaded by the trainers can be downloaded here as shown in Diagram 14.

14.2.8 Provide Feedback

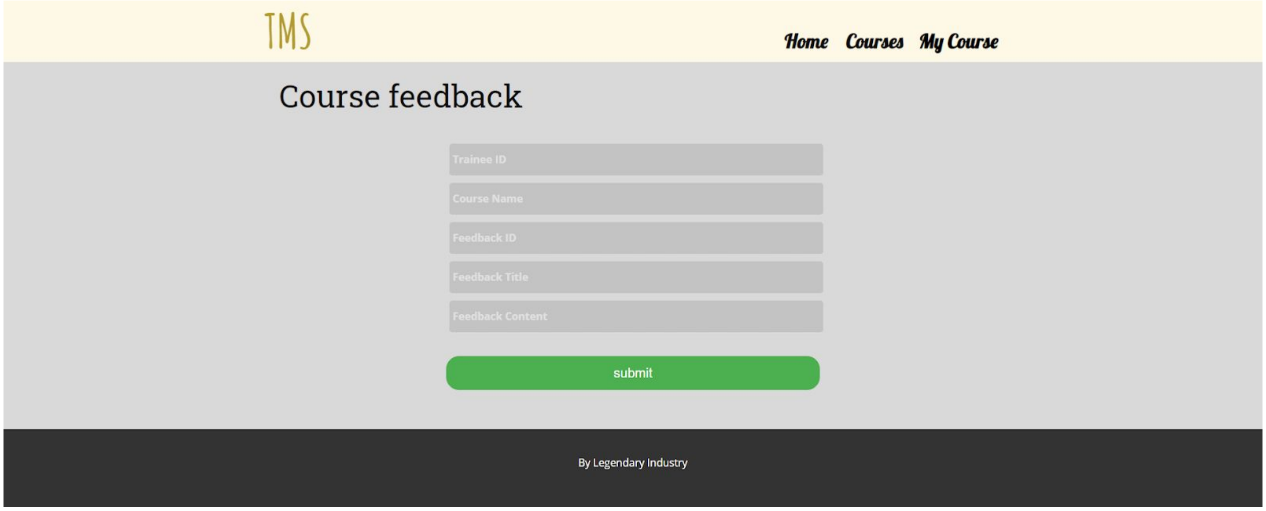


Diagram 15: Provide Feedback Page

Trainees that have enrolled to the course can provide their feedback of the course through this page.

14.3 Administrator Interface

14.3.1 Login and Sign Up

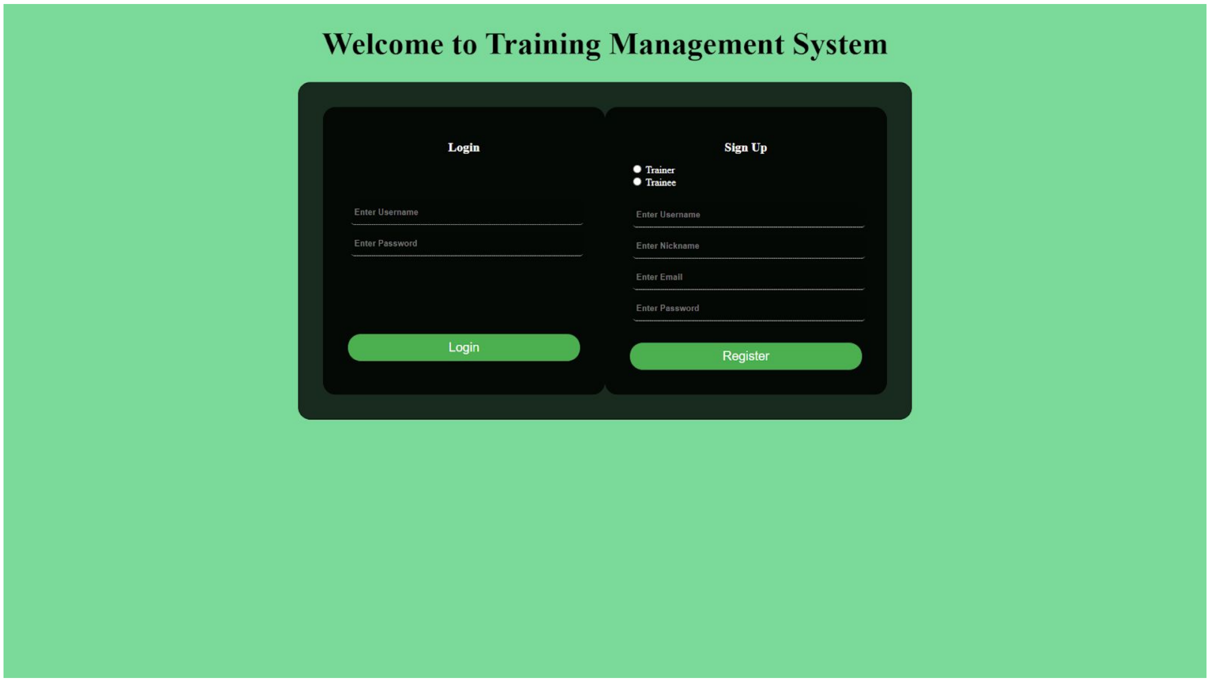


Diagram 16: Login and Signup UI

Similar to Trainer and Trainee Interface, the Login and Signup page is shown in Diagram 16 above. Administrators are able to login at the Login Session by entering their Username and Password that has been pre-set. Nobody is allowed to create an Administrator account.

14.3.2 Homepage

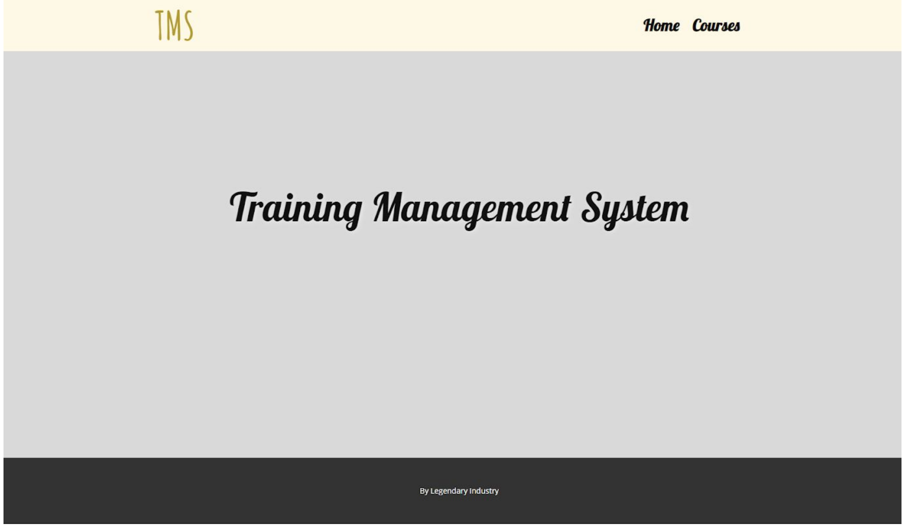


Diagram 17: Administrator Home Page

After logging into the system, the Homepage is shown which contains a navigation bar on the right corner. It has "Home" to redirect back to this Homepage later on, "Courses" to navigate to the page of all courses of the trainers in the system.

14.3.3 Course Details

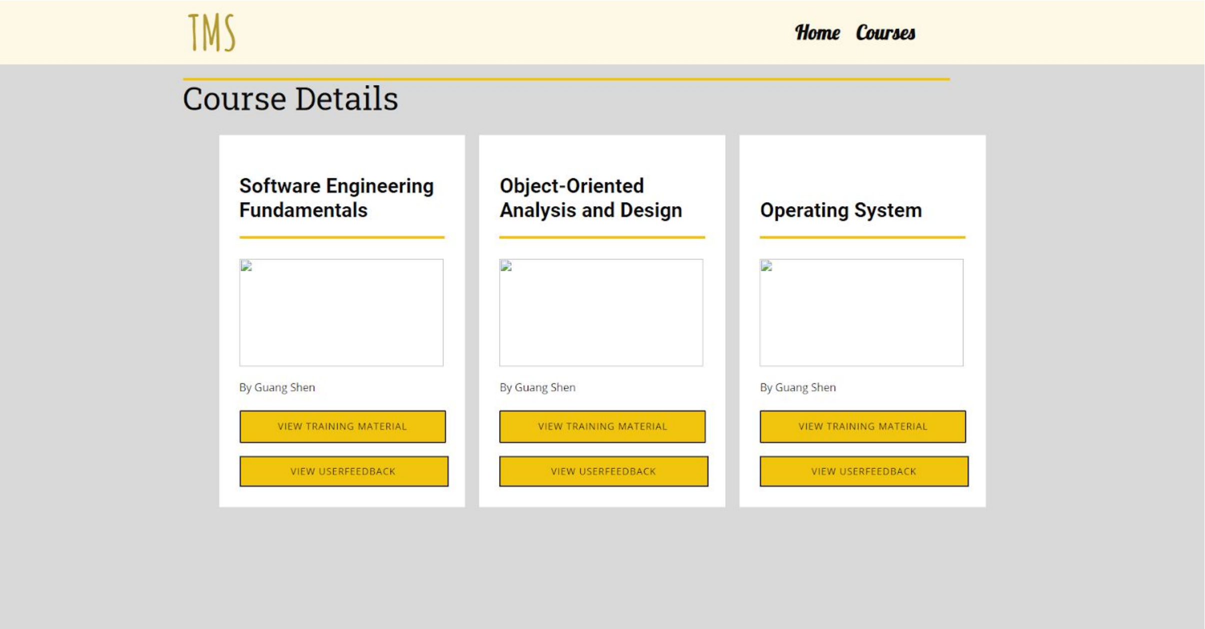
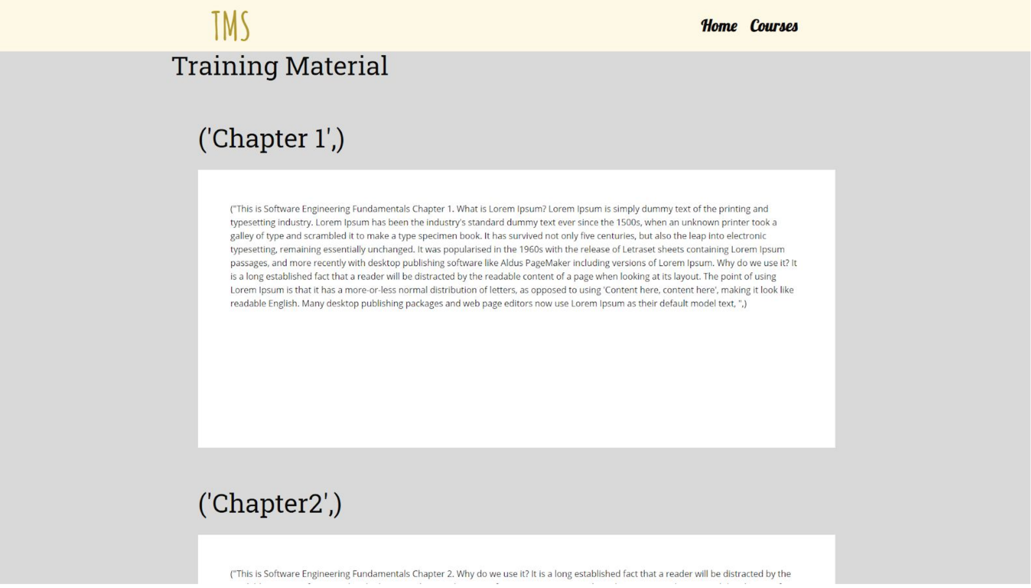


Diagram 18: Courses Page

After clicking on the "Courses" in Diagram 17, it will direct to this Course Details Page shown in Diagram 18. It can "View Training Material" and "View UserFeedback".

14.3.4 Training Materials



Training Material

('Chapter 1',)

("This is Software Engineering Fundamentals Chapter 1. What is Lorem Ipsum? Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Why do we use it? It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, ",)

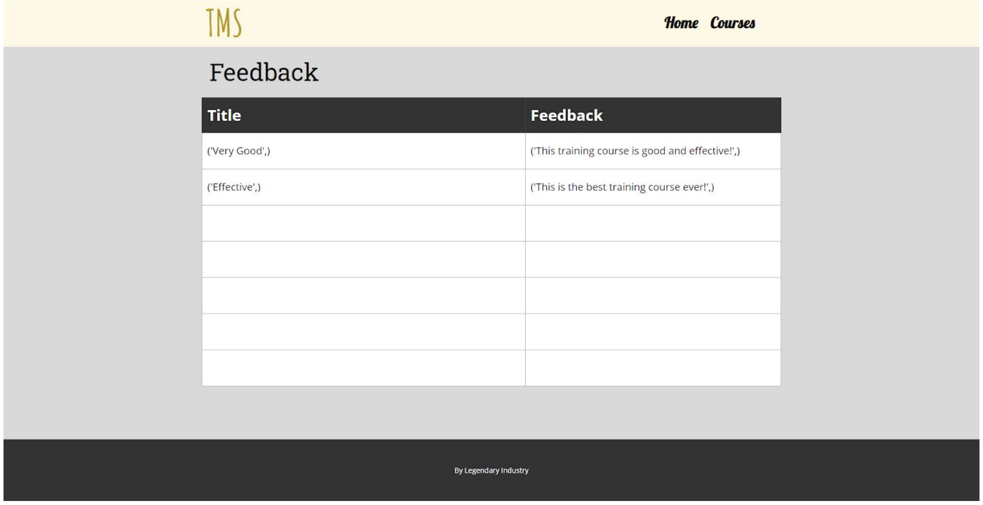
('Chapter2',)

("This is Software Engineering Fundamentals Chapter 2. Why do we use it? It is a long established fact that a reader will be distracted by the

Diagram 19: Training Materials Page

When the “View Training Material” button is clicked in Diagram 18, it will direct to this Training Materials Page as shown in Diagram 19.

14.3.6 User Feedback



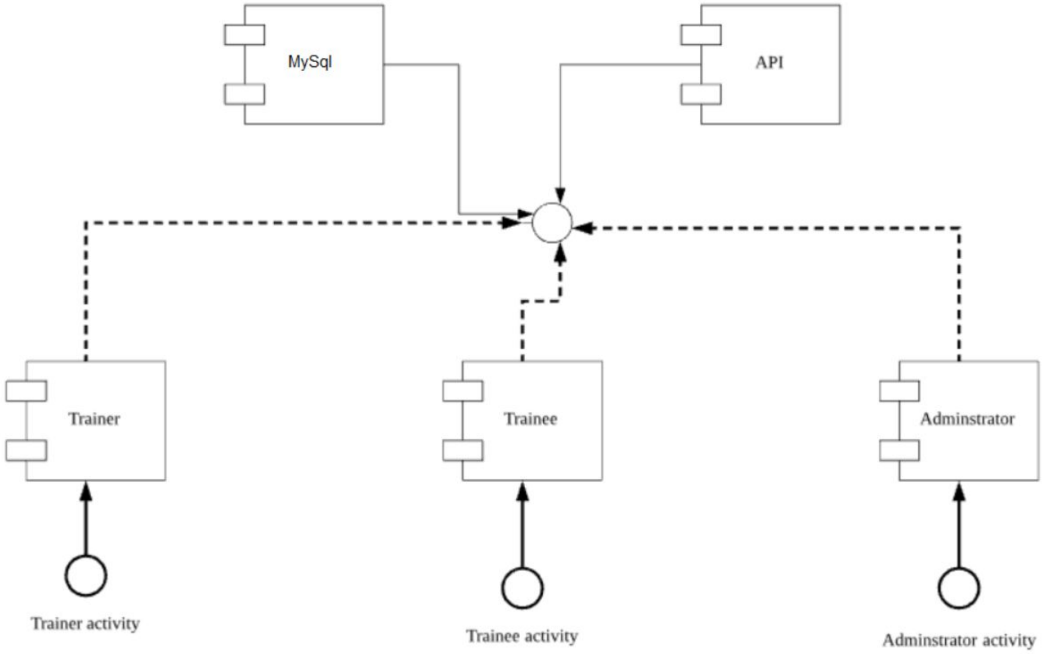
Feedback

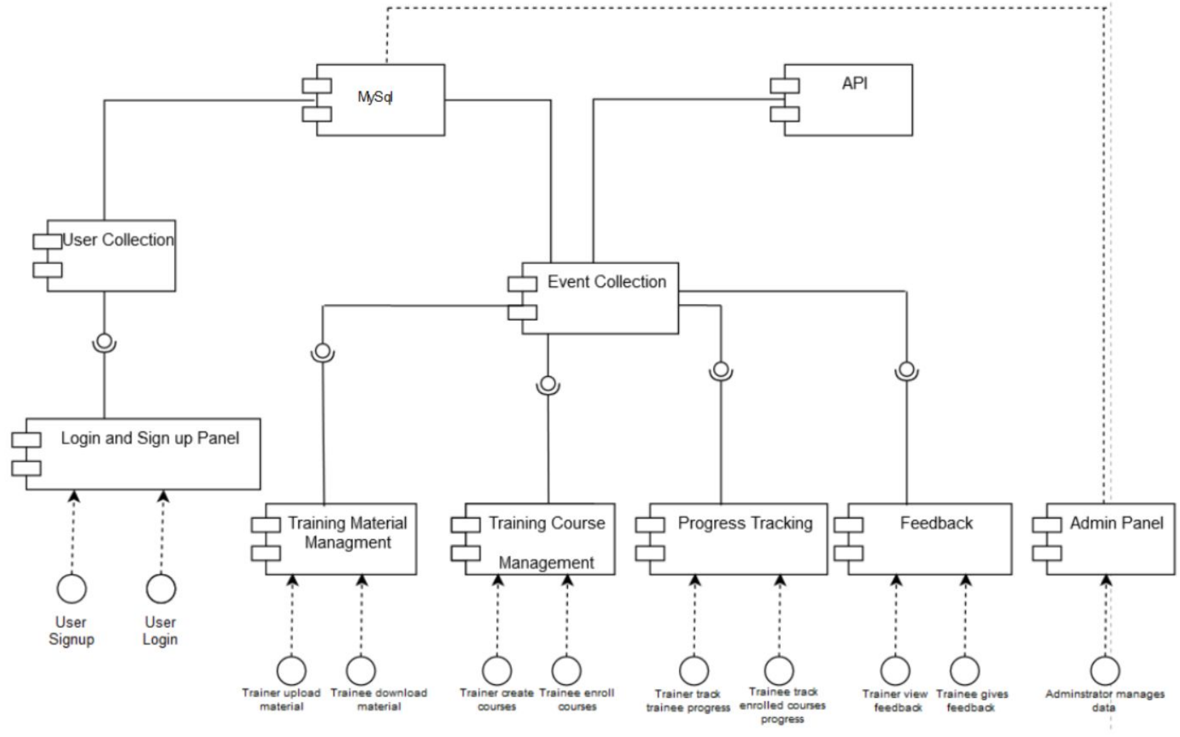
Title	Feedback
('Very Good',)	('This training course is good and effective!',)
('Effective',)	('This is the best training course ever!',)

Diagram 20: User Feedback Page

When the “View UserFeedback” button is clicked in Diagram 18, it will direct to this User Feedback Page as shown in Diagram 20.

15.0 Component Diagram





Component diagram above shows the interactions between components. There are mainly six system component interfaces for all users(student, teachers and administrators) to access which includes login and signup panel, training material management, training course management, progress tracking, feedback and admin panel. These components then interact with MySql database and API. Admin panel is connected to the MySql database which enables admin to manage data.

16.0 Algorithm Design Model Pseudocode

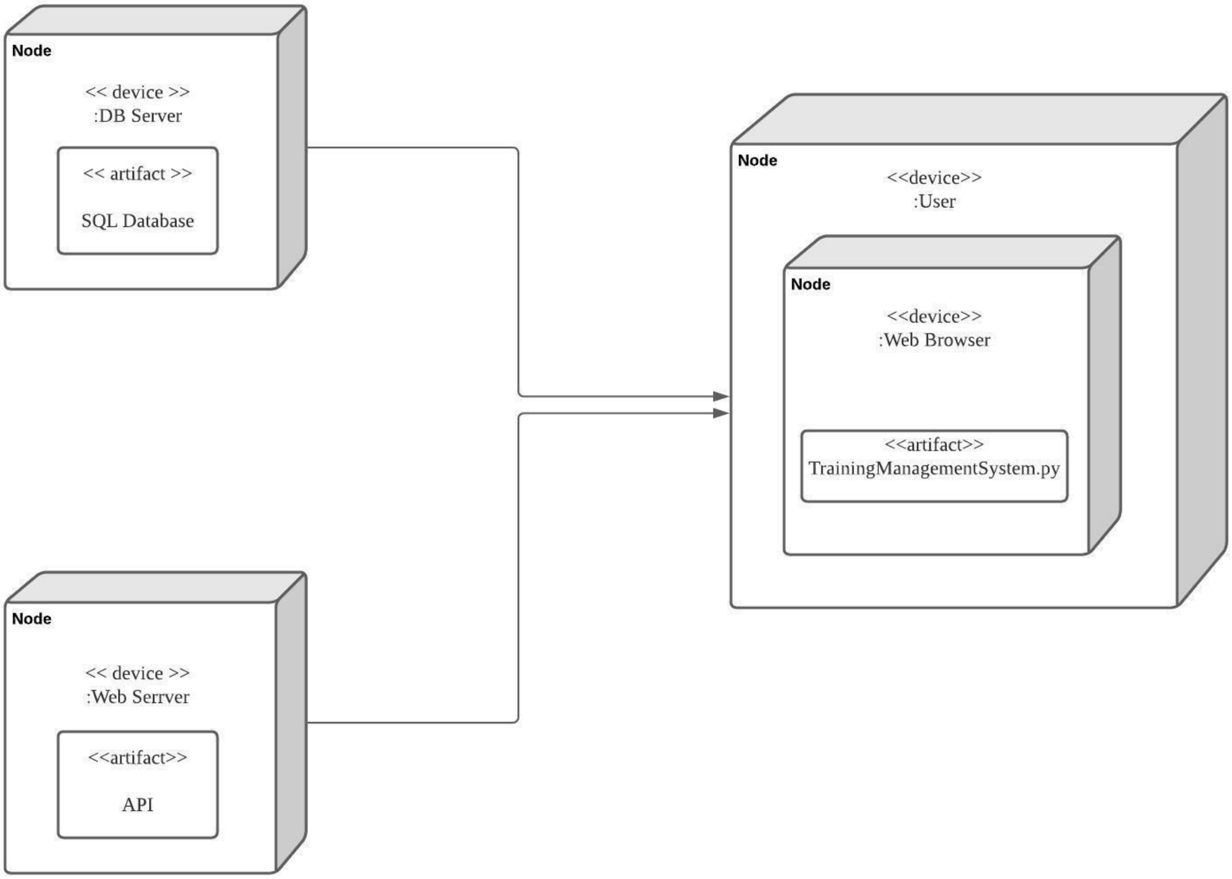
```
Program Start

Import Flask
Connect to MySQL Database

DEFINE FUNCTION firstpage():
   RETURN render_template("tms_home.html")

DEFINE FUNCTION main():
   SET selection TO the button clicked by the users

   IF selection EQUALS 'Login':
      GET username
      GET password
                  VALIDATE username & password with mySQL Database

      IF account:
                           IF accountType EQUALS ('trainee',):
            RETURN render_template("Trainee/Trainee_Home.html")

                           ELSEIF accountType EQUALS ('trainer',):
                                    RETURN render_template("Trainer/Trainer_Home.html")

                           ELSE:
                                    RETURN render_template("Admin/Admin_Home.html")

                  ELSE:
         SET error TO "Invalid Credentials! Please Try Again!"
         RETURN render_template("login.html", error=error)

   ELSEIF request.form['button'] EQUALS 'Register':
      GET accountType
      GET username
      GET nickname
      GET email

      IF accountType EQUALS 'trainee':
                           INSERT accountType, username, nickname, email into the database
         RETURN render_template("TraineeSite/Trainee_Home.html")
      ELSE:
                           INSERT accountType, username, nickname, email into the database
         RETURN render_template("TrainerSite/Trainer_Home.html")
```

```
DEFINE FUNCTION trainerhome():
   RETURN render_template("Trainer/Trainer_Home.html")

DEFINE FUNCTION trainermycourses():
   c = [" ", " ", " "]

   GET courseName from database

   course EQUALS courseName

   FOR x IN RANGE(LEN(course)):
      c[x] EQUALS course[x]

   RETURN render_template("Trainer/Trainer_MyCourse.html", c1=c[0], c2=c[1], c3=c[2])

DEFINE FUNCTION trainerAddCourse():
   RETURN render_template('Trainer/Trainer_AddCourse.html')

DEFINE FUNCTION traineraddcourse():

   IF submit:
      GET trainerID
      GET courseID
      GET courseName

      INSERT trainerID, courseID, courseName into database

      RETURN redirect(url_for('trainermycourses'))

DEFINE FUNCTION trainercoursedetails():
   RETURN render_template('Trainer/Trainer_CourseDetails.html')

DEFINE FUNCTION trainercoursedetails2():
   RETURN render_template('Trainer/Trainer_CourseDetails2.html')

DEFINE FUNCTION userfeedbacksef():
   GET courseName from database

   FOR j IN RANGE(LEN(course)):
      IF course[j] EQUALS ('Software Engineering Fundamentals',):
         s = [" ", " ", " ", " ", " ", " "]
         f = [" ", " ", " ", " ", " ", " "]
```

```
            GET traineeID from database WHERE courseName = 'Software Engineering Fundamentals' AND EQUALS trainee

            FOR i IN RANGE(LEN(trainee)):
                s[i] EQUALS trainee[i]

            GET feedbackContent from database WHERE courseName = 'Software Engineering Fundamentals' AND EQUALS feedback

            FOR i IN RANGE(LEN(feedback)):
                f[i] EQUALS feedback[i]

            RETURN render_template('Trainer/Trainer_UserFeedback_SEF.html', s1=s[0], s2=s[1], s3=s[2], s4=s[3], s5=s[4], s6=s[5], f1=f[0], f2=f[1], f3=f[2], f4=f[3], f5=f[4], f6=f[5])

DEFINE FUNCTION userfeedbackooad():
    GET courseName from database

    course EQUALS courseName

    FOR j IN RANGE(LEN(course)):
        IF course[j] EQUALS ('Object-Oriented Analysis and Design',):
            s = [" ", " ", " ", " ", " ", " "]
            f = [" ", " ", " ", " ", " ", " "]

            GET traineeID from database WHERE courseName = 'Object-Oriented Analysis and Design' AND EQUALS trainee

            FOR i IN RANGE(LEN(trainee)):
                s[i] EQUALS trainee[i]

            GER feedbackContent from database WHERE courseName = 'Object-Oriented Analysis and Design' AND EQUALS feedback

            FOR i IN RANGE(LEN(feedback)):
                f[i] = feedback[i]

            RETURN render_template('Trainer/Trainer_UserFeedback_OOAD.html', s1=s[0], s2=s[1], s3=s[2], s4=s[3], s5=s[4], s6=s[5], f1=f[0], f2=f[1], f3=f[2], f4=f[3], f5=f[4], f6=f[5])

DEFINE FUNCTION userfeedbackos():
    RETURN render_template('Trainer/Trainer_UserFeedback_OS.html')

DEFINE FUNCTION uploadCourseMaterial():
    RETURN render_template('Trainer/Trainer_Upload_CourseMaterial2.html')

DEFINE FUNCTION uploadcoursematerial():
    IF submit:
        GET courseName
        GET materialID
        GET materialName
        GET materialContent

        INSERT courseName, materialID, materialName, materialContent INTO database

        RETURN redirect(url_for('trainercoursedetails'))

DEFINE FUNCTION uploadCourseMaterial2():
    RETURN render_template('Trainer/Trainer_Upload_CourseMaterial2.html')

DEFINE FUNCTION uploadcoursematerial2():
    IF submit:
        GET courseName
        GET materialID
        GET materialName
        GET materialContent

        INSERT courseName, materialID, materialName, materialContent INTO database

        RETURN redirect(url_for('trainercoursedetails2'))

DEFINE FUNCTION traineehome():
    RETURN render_template("Trainee/Trainee_Home.html")

DEFINE FUNCTION traineecourses():
    RETURN render_template("Trainee/Trainee_Courses.html")

DEFINE FUNCTION traineemycourse():
    RETURN render_template("Trainee/Trainee_MyCourse.html")

DEFINE FUNCTION traineecoursedetailsSEF():
    RETURN render_template('Trainee/Trainee_CourseDetails_SEF.html')

DEFINE FUNCTION traineecoursedetailsOOAD():
    RETURN render_template('Trainee/Trainee_CourseDetails_OOAD.html')

DEFINE FUNCTION traineecoursedetailsOS():
    RETURN render_template('Trainee/Trainee_CourseDetails_OS.html')
```

```
DEFINE FUNCTION traineefeedbacksef():
    GET courseName from database

    course EQUALS courseName

    t = [" ", " ", " "]
    f = [" ", " ", " "]

    FOR j IN RANGE(LEN(course)):
        IF course[j] == ('Software Engineering Fundamentals',):
            GET feedbackTitle from database WHERE courseName = 'Software Engineering Fundamentals' AND EQUALS title

            FOR i IN RANGE(LEN(title)):
                t[i] EQUALS title[i]

            GET feedbackContent from database WHERE courseName = 'Software Engineering Fundamentals' AND EQUALS feedback

            FOR i IN RANGE(LEN(feedback)):
                f[i] EQUALS feedback[i]

            RETURN render_template('Trainee/Trainee_Userfeedback_SEF.html', t1=title[0], t2=title[1], f1=feedback[0], f2=feedback[1])

DEFINE FUNCTION traineefeedbackooad():
    GET courseName from database

    courseName EQUALS course

    t = [" ", " ", " "]
    f = [" ", " ", " "]

    FOR j IN RANGE(LEN(course)):
        if course[j] EQUALS ('Object-Oriented Analysis and Design',):

            GET feedbackTitle FROM database WHERE courseName = 'Object-Oriented Analysis and Design' AND EQUALS title

            FOR i IN RANGE(LEN(title)):
                t[i] EQUALS title[i]
            GET feedbackContent FROM database WHERE courseName = 'Object-Oriented Analysis and Design' AND EQUALS feedback

            FOR i IN RANGE(LEN(feedback)):
                f[i] EQUALS feedback[i]

            RETURN render_template('Trainee/Trainee_Userfeedback_OOAD.html', t1=title[0], t2=title[1], f1=feedback[0], f2=feedback[1])

DEFINE FUNCTION traineefeedbackos():
    RETURN render_template('Trainee/Trainee_Userfeedback_OS.html')

DEFINE FUNCTION traineegiveuserfeedback():
    RETURN render_template('Trainee/Trainee_GiveUserfeedback.html')

DEFINE FUNCTION traineeGiveUserFeedback():
    IF submit:
        GET traineeID
        GET courseName
        GET feedbackID
        GET feedbackTitle
        GET feedbackContent

        INSERT traineeID, courseName, feedbackID, feedbackTitle, feedbackContent INTO database

        RETURN redirect(url_for('traineemycourse'))

DEFINE FUNCTION traineeviewtrainingmaterialssef():
    GET courseName FROM database

    course EQUALS courseName

    IF course:
        t = [" ", " ", " ", " ", " ", " "]
        c = [" ", " ", " ", " ", " ", " "]

        SELECT materialName FROM database WHERE courseName = 'Software Engineering Fundamentals' AND EQUALS material

        FOR i IN RANGE(LEN(material)):
            t[i] EQUALS material[i]

        SELECT materialContent FROM database WHERE courseName = 'Software Engineering Fundamentals' AND EQUALS content
        FOR i IN RANGE(LEN(content)):
            c[i] EQUALS content[i]

        RETURN render_template('Trainee/Trainee_View_TrainingMaterials_SEF.html', t1=t[0], t2=t[1], t3=t[2], t4=t[3], c1=c[0], c2=c[1], c3=c[2], c4=c[3])

DEFINE FUNCTION traineeviewtrainingmaterialsooad():
    GET courseName FROM database

    course EQUALS courseName

    IF course:
        t = [" ", " ", " ", " ", " ", " "]
        c = [" ", " ", " ", " ", " ", " "]

        GET materialName FROM database WHERE courseName = 'Object-Oriented Analysis and Design' AND EQUALS material

        FOR i IN RANGE(LEN(material)):
            t[i] EQUALS material[i]

        GET materialContent FROM database WHERE courseName = 'Object-Oriented Analysis and Design' AND EQUALS content

        FOR i IN RANGE(LEN(content)):
            c[i] EQUALS content[i]

        RETURN render_template('Trainee/Trainee_View_TrainingMaterials_OOAD.html', t1=t[0], t2=t[1], t3=t[2], t4=t[3], c1=c[0], c2=c[1], c3=c[2], c4=c[3])

DEFINE FUNCTION traineeviewtrainingmaterialsos():
    GET courseName FROM database

    course EQUALS courseName

    IF course:
        t = [" ", " ", " ", " ", " ", " "]
        c = [" ", " ", " ", " ", " ", " "]

        GET materialName FROM database WHERE courseName = 'Operating System' AND EQUALS material

        FOR i IN RANGE(LEN(material)):
            t[i] EQUALS material[i]

        GET materialContent FROM database WHERE courseName = 'Operating System' AND EQUALS content

        FOR i IN RANGE(LEN(content)):
            c[i] EQUALS content[i]

        RETURN render_template('Trainee/Trainee_View_TrainingMaterials_OS.html', t1=t[0], t2=t[1], t3=t[2], t4=t[3], c1=c[0], c2=c[1], c3=c[2], c4=c[3])

DEFINE FUNCTION adminhome():
    RETURN render_template("Admin/Admin_Home.html")

DEFINE FUNCTION admincourses():
    RETURN render_template("Admin/Admin_Courses.html")

DEFINE FUCNTION admin_user_feedback_SEF():
    GET courseName FROM database

    course EQUALS courseName

    FOR j IN RANGE(LEN(course)):
        IF course[j] EQUALS ('Software Engineering Fundamentals',):
            t = [" ", " ", " ", " ", " ", " ", " "]
            f = [" ", " ", " ", " ", " ", " ", " "]

            GET feedbackTitle FROM database WHERE courseName = 'Software Engineering Fundamentals' AND EQUALS title

            FOR i IN RANGE(LEN(title)):
                t[i] EQUALS title[i]

            GET feedbackContent FROM database WHERE courseName = 'Software Engineering Fundamentals' AND EQUALS feedback

            FOR i IN RANGE(LEN(feedback)):
                f[i] EQUALS feedback[i]

            RETURN render_template('Admin/Admin_UserFeedback_SEF.html', t1=t[0], t2=t[1], t3=t[2], t4=t[3], t5=t[4], t6=t[5], t7=t[6], f1=f[0], f2=f[1], f3=f[2], f4=f[3], f5=f[4], f6=f[5], f7=f[6])


DEFINE FUNCTION admin_user_feedback_OOAD():
    GET courseName FROM database

    course EQUALS courseName

    FOR j IN RANGE(LEN(course)):
        IF course[j] EQUALS ('Object-Oriented Analysis and Design',):
            t = [" ", " ", " ", " ", " ", " ", " "]
            f = [" ", " ", " ", " ", " ", " ", " "]

            GET feedbackTitle FROM database WHERE courseName = 'Object-Oriented Analysis and Design' AND EQUALS title

            FOR i IN RANGE(LEN(title)):
                t[i] EQUALS title[i]

            GET feedbackContent FROM database WHERE courseName = 'Object-Oriented Analysis and Design' AND EQUALS feedback

            FOR i IN RANGE(LEN(feedback)):
                f[i] EQUALS feedback[i]

                                    RETURN render_template('Admin/Admin_UserFeedback_OOAD.html', t1=t[0], t2=t[1], t3=t[2], t4=t[3], t5=t[4], t6=t[5], t7=t[6], f1=f[0], f2=f[1], f3=f[2], f4=f[3], f5=f[4], f6=f[5], f7=f[6])

DEFINE FUNCTION admin_user_feedback_OS():
    GET courseName FROM database

    course EQUALS courseName

    FOR j IN RANGE(LEN(course)):
        IF course[j] EQUALS ('Operating System',):
            t = [" ", " ", " ", " ", " ", " ", " "]
            f = [" ", " ", " ", " ", " ", " ", " "]

            GET feedbackTitle FROM database WHERE courseName = 'Operating System' AND EQUALS title

            FOR i IN RANGE(LEN(title)):
                t[i] EQUALS title[i]

            GET feedbackContent FROM database WHERE courseName = 'Operating System' AND EQUALS feedback
            FOR i IN RANGE(LEN(feedback)):
                f[i] EQUALS feedback[i]

            RETURN render_template('Admin/Admin_UserFeedback_OS.html', t1=t[0], t2=t[1], t3=t[2], t4=t[3], t5=t[4], t6=t[5], t7=t[6], f1=f[0], f2=f[1], f3=f[2], f4=f[3], f5=f[4], f6=f[5], f7=f[6])

DEFINE FUNCTION adminviewtrainingmaterialssef():
    GET courseName FROM training_materials

    course EQUALS courseName

    IF course:
        t = [" ", " ", " ", " ", " ", " "]
        c = [" ", " ", " ", " ", " ", " "]

        GET materialName FROM database WHERE courseName = 'Software Engineering Fundamentals' AND EQUALS material

        FOR i IN RANGE(LEN(material)):
            t[i] EQUALS material[i]

        GET materialContent FROM database WHERE courseName = 'Software Engineering Fundamentals' AND EQUALS content

        FOR i IN RANGE(LEN(content)):
            c[i] EQUALS content[i]

        RETURN render_template('Admin/Admin_View_TrainingMaterials_SEF.html', t1=t[0], t2=t[1], t3=t[2], t4=t[3], c1=c[0], c2=c[1], c3=c[2], c4=c[3])



DEFINE FUNCTION adminviewtrainingmaterialsooad():
    GET courseName FROM database

    course EQUALS courseName

    IF course:
        t = [" ", " ", " ", " ", " ", " "]
        c = [" ", " ", " ", " ", " ", " "]

        GET materialName FROM database WHERE courseName = 'Object-Oriented Analysis and Design' AND EQUALS material
        FOR i IN RANGE(LEN(material)):
            t[i] EQUALS material[i]

        GET materialContent FROM database WHERE courseName = 'Object-Oriented Analysis and Design' AND EQUALS content

        FOR i IN RANGE(LEN(content)):
            c[i] EQUALS content[i]

        RETURN render_template('Admin/Admin_View_TrainingMaterials_OOAD.html', t1=t[0], t2=t[1], t3=t[2], t4=t[3], c1=c[0], c2=c[1], c3=c[2], c4=c[3])


DEFINE FUNCTION adminviewtrainingmaterialsos():
    GET courseName FROM database

    course EQUALS courseName

    IF course:
        t = [" ", " ", " ", " ", " ", " "]
        c = [" ", " ", " ", " ", " ", " "]

        GET materialName FROM database WHERE courseName = 'Operating System' AND EQUALS material

        FOR i IN RANGE(LEN(material)):
            t[i] EQUALS material[i]

        GET materialContent FROM database WHERE courseName = 'Operating System' AND EQUALS content

        FOR i in RANGE(LEN(content)):
            c[i] EQUALS content[i]

        RETURN render_template('Admin/Admin_View_TrainingMaterials_OS.html', t1=t[0], t2=t[1], t3=t[2], t4=t[3], c1=c[0], c2=c[1], c3=c[2], c4=c[3])

IF __name__ == "__main__":
    app.run(debug=True)
```

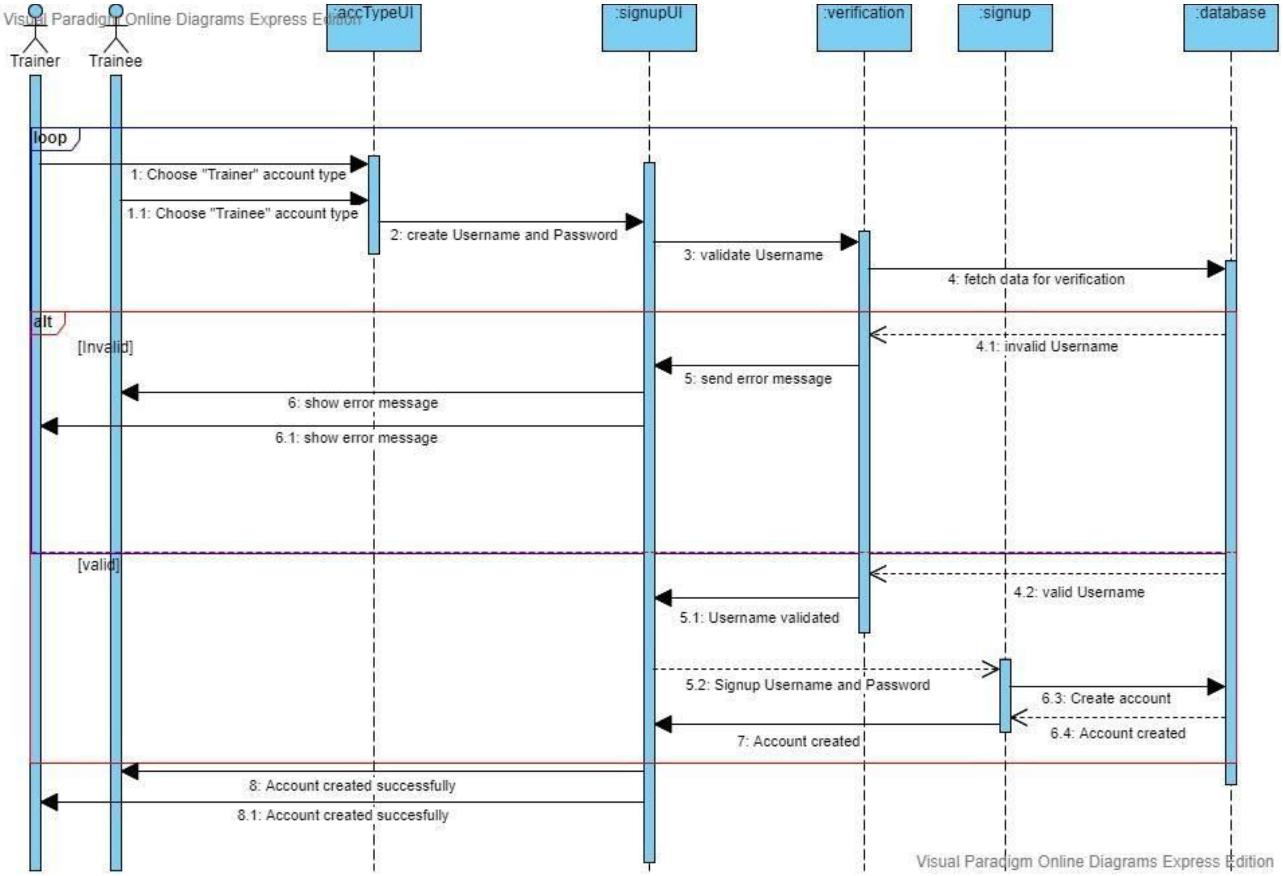
17.0 Deployment Diagram



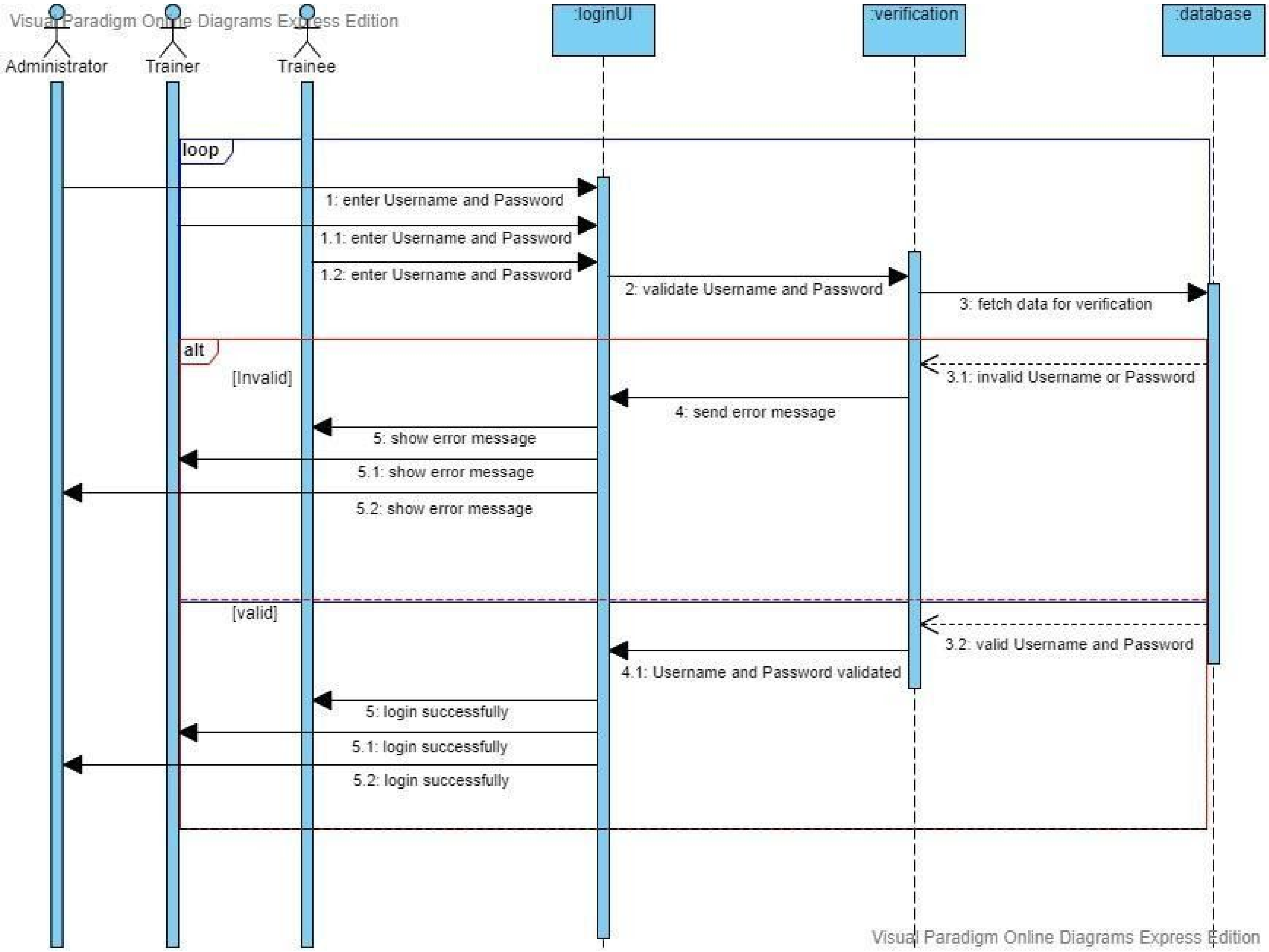
The deployment diagram describes the servers and the user device road path which is involved in the project. It is separated into a user device, DB server, web server and shows how these components interact with each other.

18.0 Sequence Diagram

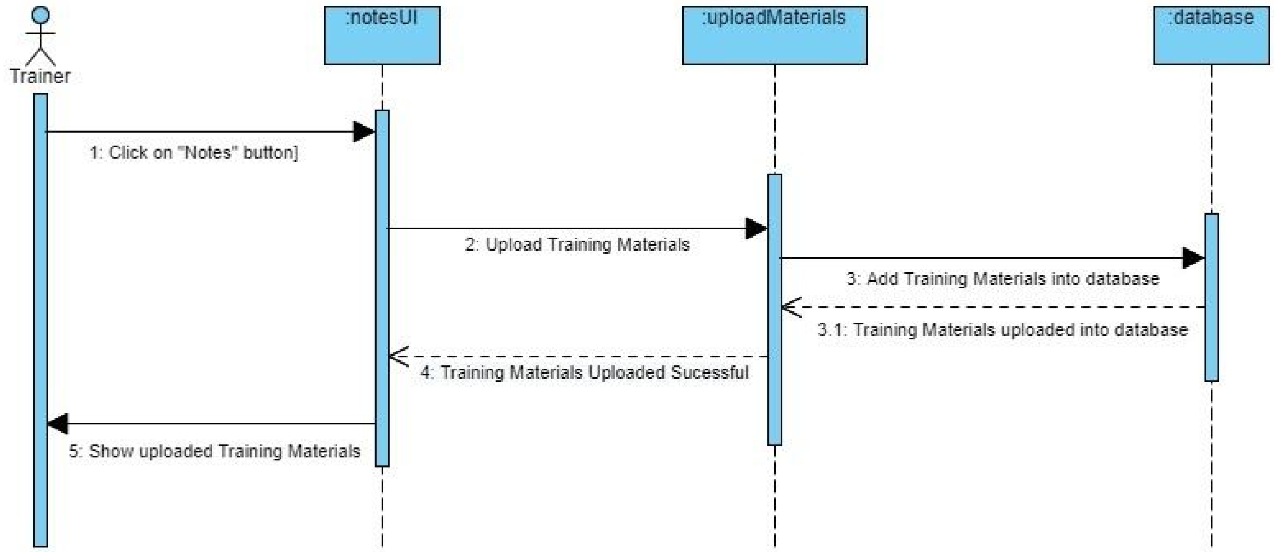
18.1 Sign up Management


Visual Paradigm Online Diagrams Express Edition
Trainer
Trainee
:accTypeUI
:signupUI
:verification
:signup
:database
loop
1: Choose "Trainer" account type
1.1: Choose "Trainee" account type
2: create Username and Password
3: validate Username
4: fetch data for verification
alt
[Invalid]
4.1: invalid Username
5: send error message
6: show error message
6.1: show error message
[valid]
4.2: valid Username
5.1: Username validated
5.2: Signup Username and Password
6.3: Create account
6.4: Account created
7: Account created
8: Account created successfully
8.1: Account created succesfully
Visual Paradigm Online Diagrams Express Edition

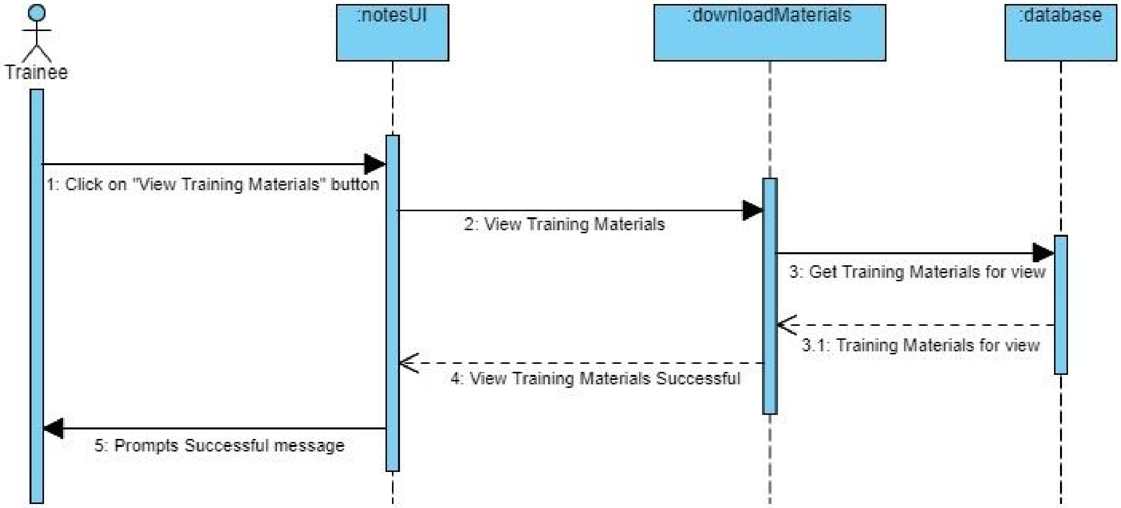
18.2 Login Management


Visual Paradigm Online Diagrams Express Edition
Administrator
Trainer
Trainee
:loginUI
:verification
:database
loop
1: enter Username and Password
1.1: enter Username and Password
1.2: enter Username and Password
2: validate Username and Password
3: fetch data for verification
alt
[Invalid]
3.1: invalid Username or Password
4: send error message
5: show error message
5.1: show error message
5.2: show error message
[valid]
3.2: valid Username and Password
4.1: Username and Password validated
5: login successfully
5.1: login successfully
5.2: login successfully
Visual Paradigm Online Diagrams Express Edition

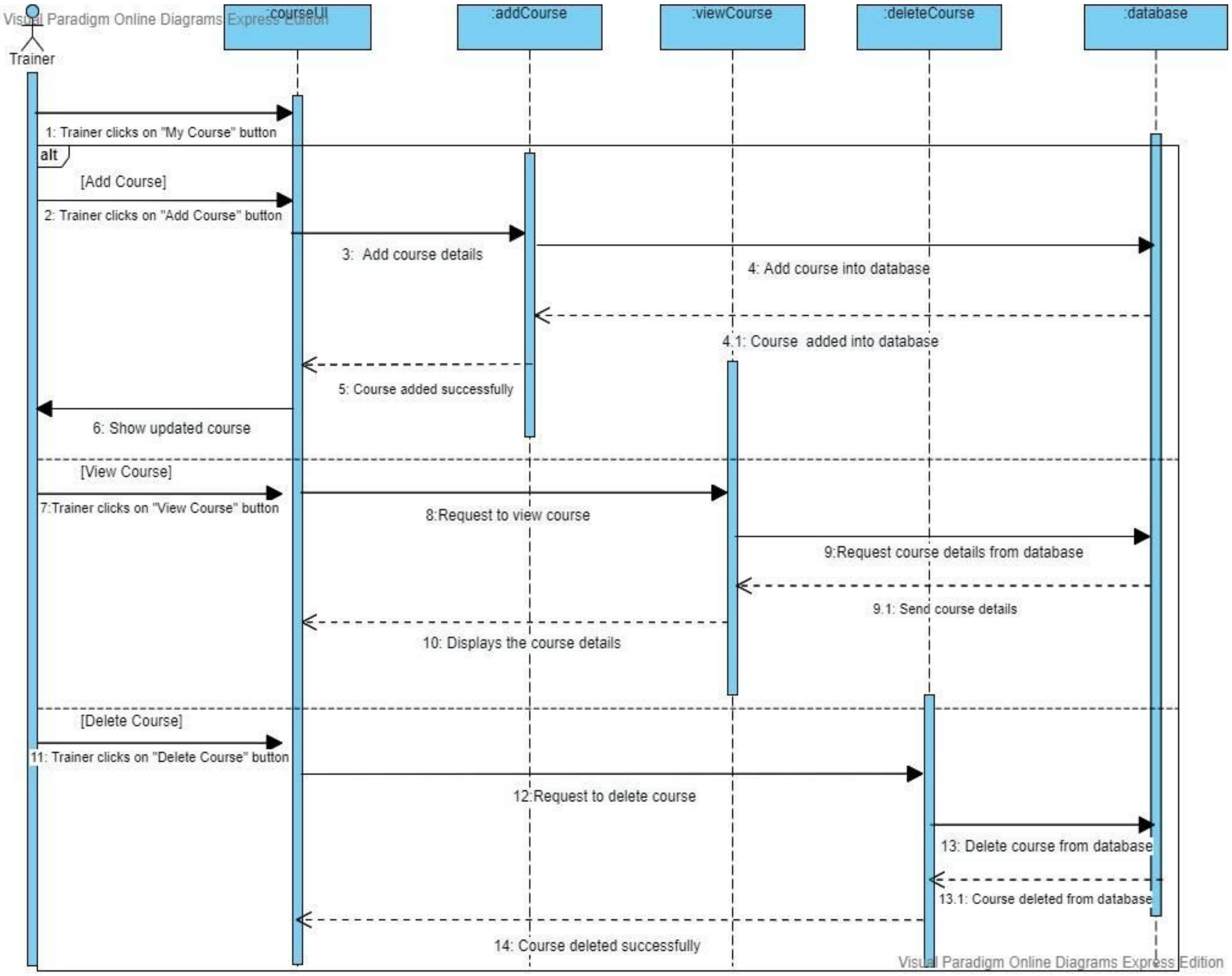
18.3 Upload Training Materials Management


Trainer
:notesUI
:uploadMaterials
:database
1: Click on "Notes" button]
2: Upload Training Materials
3: Add Training Materials into database
3.1: Training Materials uploaded into database
4: Training Materials Uploaded Sucessful
5: Show uploaded Training Materials

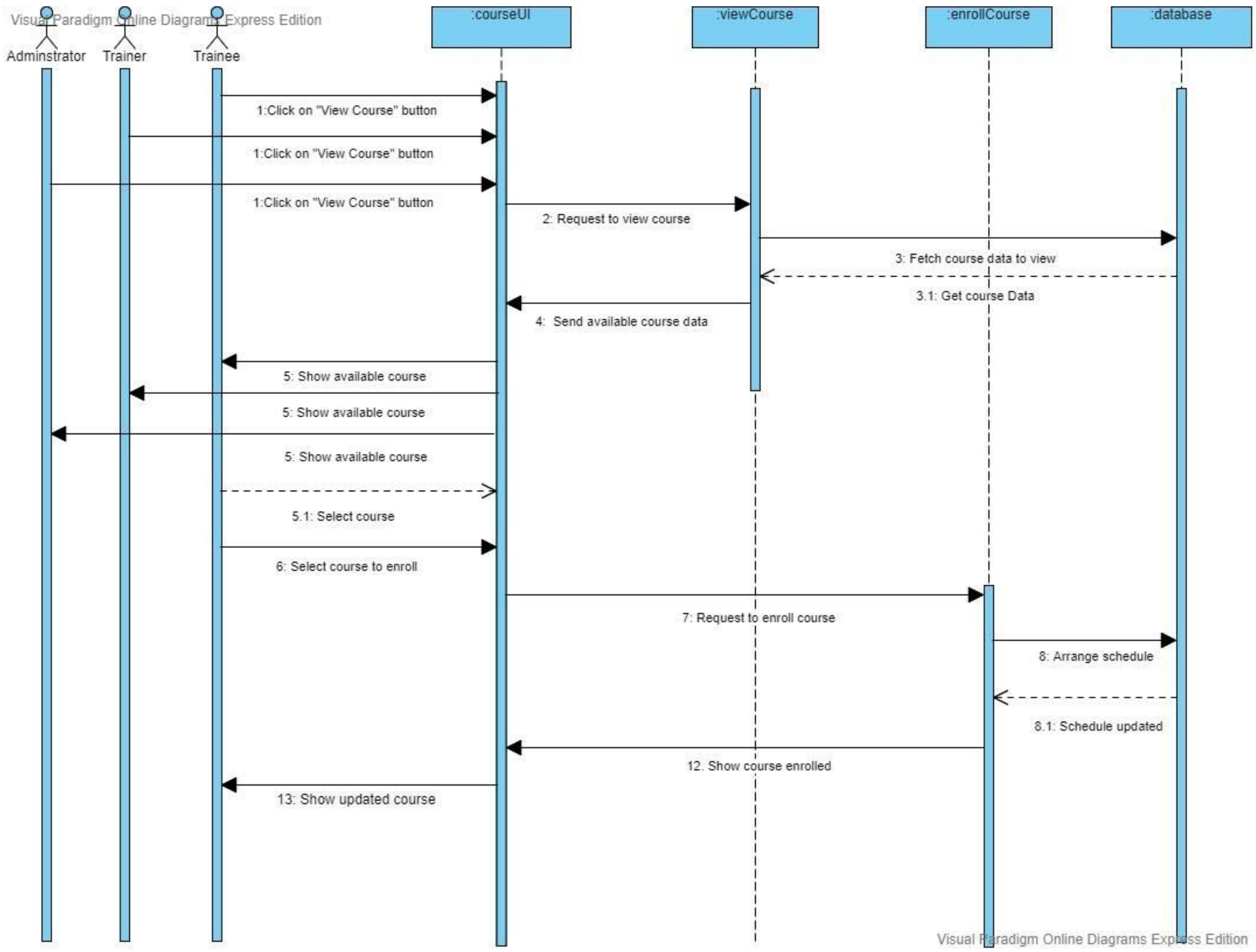
18.4 View Training Materials Management


Trainee
:notesUI
:downloadMaterials
:database
1: Click on "View Training Materials" button
2: View Training Materials
3: Get Training Materials for view
3.1: Training Materials for view
4: View Training Materials Successful
5: Prompts Successful message

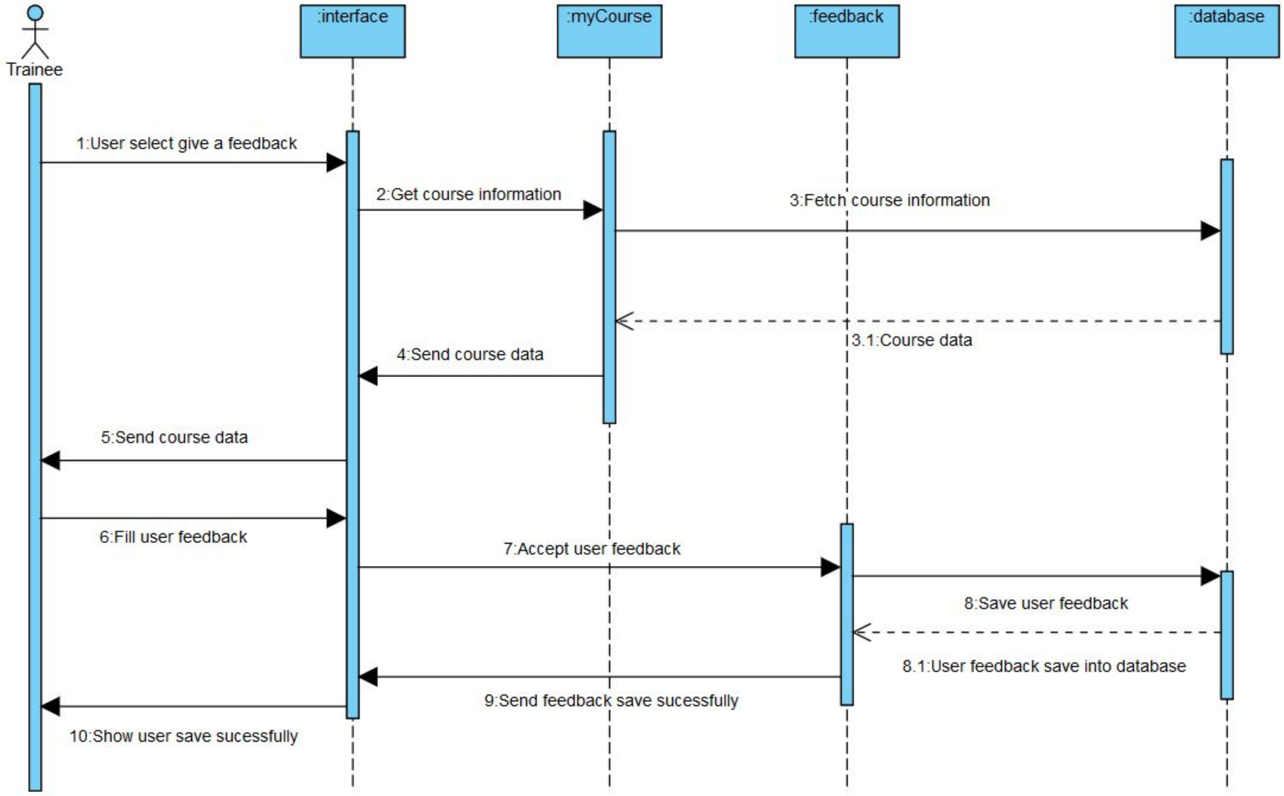
18.5 Create Course Management


Visual Paradigm Online Diagrams Express Edition
Trainer
:courseUI
:addCourse
:viewCourse
:deleteCourse
:database
1: Trainer clicks on "My Course" button
alt
[Add Course]
2: Trainer clicks on "Add Course" button
3: Add course details
4: Add course into database
4.1: Course added into database
5: Course added successfully
6: Show updated course
[View Course]
7:Trainer clicks on "View Course" button
8:Request to view course
9:Request course details from database
9.1: Send course details
10: Displays the course details
[Delete Course]
11: Trainer clicks on "Delete Course" button
12:Request to delete course
13: Delete course from database
13.1: Course deleted from database
14: Course deleted successfully
Visual Paradigm Online Diagrams Express Edition

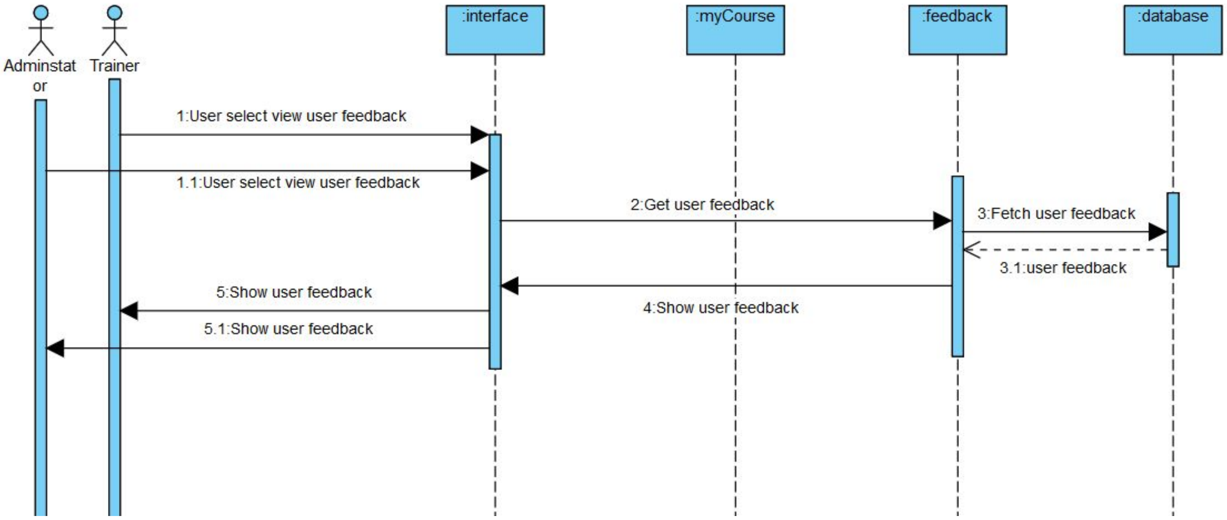
18.6 Enroll Course Management


Visual Paradigm Online Diagrams Express Edition
Adminstrator
Trainer
Trainee
:courseUI
:viewCourse
:enrollCourse
:database
1:Click on "View Course" button
1:Click on "View Course" button
1:Click on "View Course" button
2: Request to view course
3: Fetch course data to view
3.1: Get course Data
4: Send available course data
5: Show available course
5: Show available course
5: Show available course
5.1: Select course
6: Select course to enroll
7: Request to enroll course
8: Arrange schedule
8.1: Schedule updated
12. Show course enrolled
13: Show updated course
Visual Paradigm Online Diagrams Express Edition

18.7 Give User Feedback Management



18.8 Show User Feedback Management



19.0 Black Box Testing

19.1 Trainer Login

Test Case ID: TC1

Test Case Name: Trainer Login

No	Actions	Input (Valid/Invalid)	Expected Output	Actual Output	Pass/Fail	Remarks
1	User enters invalid trainerID and password	Invalid trainerID = ‘leeguangshen’ password = ‘passabc’	Error message ‘Incorrect Username or Password, Please Try Again.’	Error message ‘Incorrect Username or Password, Please Try Again.’	Pass	Invalid trainerID
		Invalid trainerID = ‘guangshenlee’ password = ‘abcdef’	Error message ‘Incorrect Username or Password, Please Try Again.’	Error message ‘Incorrect Username or Password, Please Try Again.’	Pass	Invalid Password
2	User enters valid trainerID and password	Valid trainerID = ‘leeguangshen’ password = ‘abcdef’	Login successfully	Login successfully	Pass	

19.2 Trainer Signup

Test Case ID: TC2

Test Case Name: Trainer Signup

No	Actions	Input (Valid/Invalid)	Expected Output	Actual Output	Pass/Fail	Remarks
1	User enters invaid email format	Invalid Email: “jh.com”	Operation Error	Operation Error	Pass	Invalid email format
2	User enters	Valid Email: “jh@gmail.com”	Login successfully	Login successfully	Pass	

19.3 Trainer Create Course

Test Case ID: TC3

Test Case : Trainer create course

No	Actions	Input (valid/invalid)	Expected Output	Actual Output	Pass/Fail	Remarks
1	User enters a blank course name	Invalid Course Name: “ ”	Error Message “Please fill out this field.”	Error Message “Please fill out this field.”	Pass	Blank course name
2	User enters a blank course ID	Invalid course ID :“ ”	Error Message “Please fill out this field.”	Error Message “Please fill out this field.”	Pass	Blank course ID
3	User enters course name and course descriptions.	Valid Course:“ OOAD” Course description: “Object-Oriented Analysis and Design”	Course created successfully	Course created successfully	Pass	

19.4 Trainee Enroll Course

Test Case ID: TC4

Test Case : Trainee enroll course

No	Actions	Input (valid/invalid)	Expected Output	Actual Output	Pass/Fail	Remarks
1	User choose a course to enroll	valid course:“OOAD”	Course enrolled successfully	Course enrolled successfully	Pass	

19.5 Trainer Upload Training Materials

Test Case ID: TC5

Test Case Name: Trainer upload training materials

No	Actions	Input (valid/invalid)	Expected Output	Actual Output	Pass/Fail	Remarks
1	User enter a blank training materials name	Invalid Training Materials Name: “ ”	Error Message “Please fill out this field.”	Error Message “Please fill out this field.”	Pass	Invalid training materials name

19.6 Trainer / Trainee View Training Materials

Test Case ID: TC6

Test Case Name: Trainer / Trainee view training materials

No	Actions	Input (valid/invalid)	Expected Output	Actual Output	Pass/Fail	Remarks
1	User view training materials	Valid	Show training materials	Show training materials	Pass	

19.8 Give User Feedback Management

Test Case ID: TC7

Test Case : User give user's feedback

No	Actions	Input (valid/invalid)	Expected Output	Actual Output	Pass/Fail	Remarks
1	User enter blank feedback.	Invalid FeedbackContent = " "	Error Message "Please fill out this field."	Error Message "Please fill out this field."	Pass	
2	User enter valid user's feedback	Valid FeedbackContent = "This course is useful"	Feedback submitted successfully.	Feedback submitted successfully.	Pass	

19.9 Show User Feedback Management

Test Case ID: TC8

Test Case : User view user's feedback

No	Actions	Input (valid/invalid)	Expected Output	Actual Output	Pass/Fail	Remarks
1	User view user's feedback	Valid	Show user feedback.	Show user feedback.	Pass	