

E-Learning System

Revision History

| Version No. | Date | Prepared by / Modified by | Significant Changes |
|-------------|------------|---------------------------|-------------------------------|
| 01 | 13/04/2022 | Team C | document submitted for review |
| | | | |

Glossary

| Abbreviation | Description |
|--------------|----------------------|
| BSC | Business Story Card |
| TSC | Technical Story Card |
| DB | Data Base |
| | |
| | |
| | |

Table of Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 3 |
| 2 | Scope of Change | 3 |
| 3 | List of impacted modules | 3 |
| 4 | Design and Detailed technical updates | 4 |
| 4.1 | Process model | 4 |
| 4.1.1 | Use case Model | 4 |
| 4.1.2 | Use Case Scenarios and Wireframes | 5 |
| 4.1.3 | Class Diagram | 9 |
| 4.1.4 | ER Diagram | 10 |
| 4.1.5 | Sequence Diagram | 12 |
| 4.2 | Proposed user Interface design | 13 |
| 4.3 | Database design changes | 21 |
| 4.4 | Refactoring related changes | 25 |
| 4.5 | Construction strategy and re-use | 27 |
| 5 | Details of Alternative Design Approach | 27 |
| 6 | Other Technical changes | 28 |
| 6.1 | Automation tasks / changes | 28 |
| 6.2 | CI / Build relates tasks / changes | 28 |
| 6.3 | Non-functional related changes | 28 |
| 7 | Additional details | 29 |
| 7.1 | Open Questions / clarifications / Assumptions | 29 |
| 7.2 | Additional notes to technical team | 29 |
| 8 | References | 29 |

1 Introduction

E-learning system is a platform which allows trainers and learners to connect. Learners once added to the platform, could view learning materials related to the courses they have been enrolled. Learners have a unique account page which they could navigate between courses. Trainers could publish notices, quizzes and learning material to each course they are assigned to. Learning materials include documents(.pdf, .ppt, .doc), images, audio, video, and external links.

Learners could attempt quizzes within the deadline in which they are active. Grading for quizzes would be generated automatically by the system. Additionally, learners could provide feedback on the quality of the courses they are enrolled in. Notices could be published through the application.

The admin could register learners, trainers and other admins to the system and generate a login for each of the people. Moreover, users of the system could also register themselves to the system by submitting the registration form.

2 Scope of Change

Change will not be entertained and whatever specifications mentioned in this document is final.

3 List of impacted modules

All the functional modules will be created from scratch.

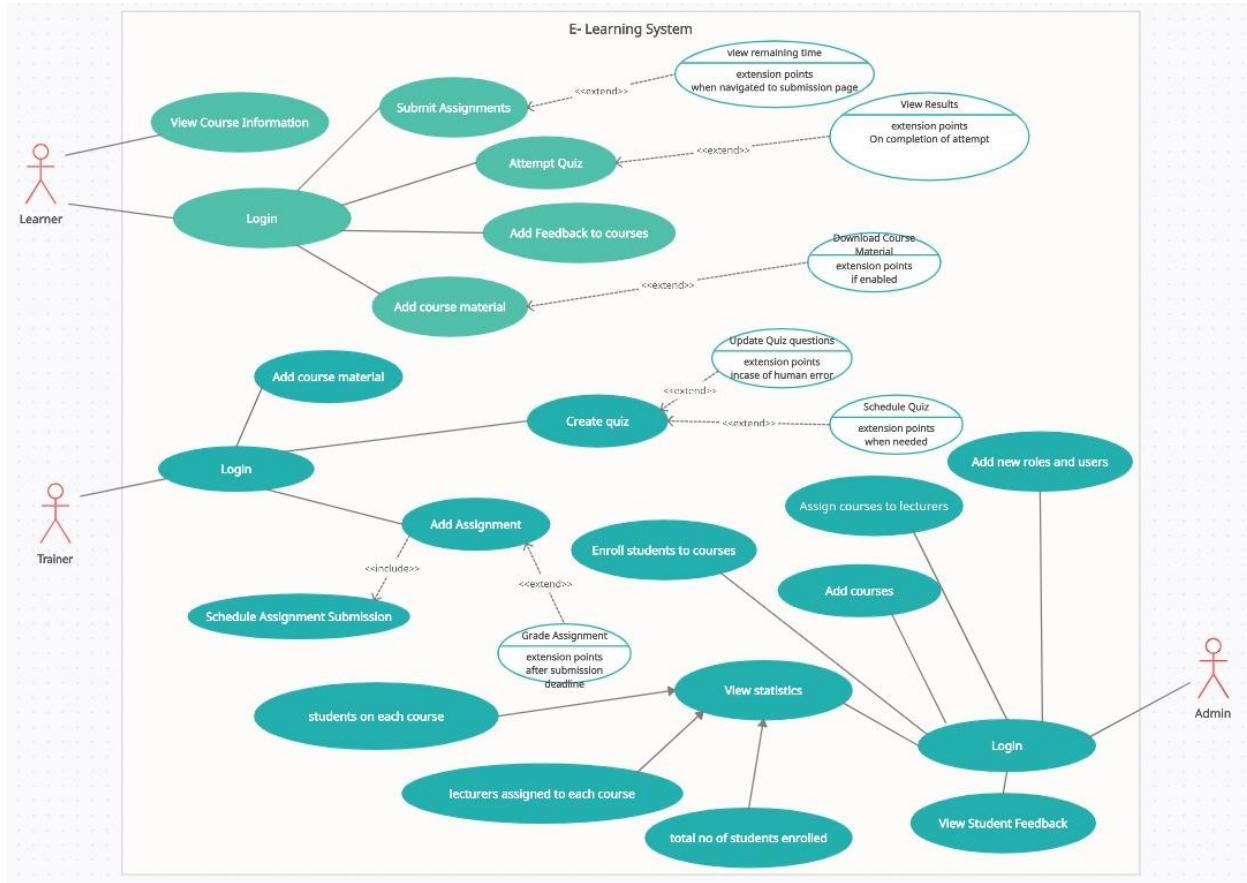
Impacted Modules are as follows;

- add new Learners to the system
- add new courses and view available course list
- add new lessons with course material to existing courses
- view course page consisting of lessons and course material
- attempt quizzes and view results of quizzes
- add feedback on the courses and view feedback on courses
- sign in and registration of users
- adding notices to system

4 Design and Detailed technical updates

4.1 Process model

4.1.1 Use case model



4.1.2 Use case scenario

A Web Page

https://Sign_Up

E-learning Platform

An online learning platform is a webspace or portal for educational content and resources that offer a student everything they need in one place: lectures, resources, opportunities to meet and chat with other students, and more. It is also an excellent way for the student and the teacher to monitor student progress. The first advantage of e-learning is flexibility in terms of time and place. Learning content is

Learn More

Welcome!

Please Sign Up To Continue

Name
Enter Name Required

UserName
Enter UserName Required

Email
Enter Email Required

Password
Enter Password Required

Confirm Password
Enter Confirm Password Required

Sign up

Already have an account ? [Sign In](#)

| Brief Description | Student Sign Up |
|-------------------|--|
| Basic Flow | <p>This use case describes how a student sign up in to the system</p> <ol style="list-style-type: none">1. The student has to sign up himself into the system.2. After the successful sign up, student can access the E-learning system.3. The following information is required during the sign up process<ul style="list-style-type: none">· Name· Email· User name· Password· Re-Enter Password |

| | |
|-----------------|--|
| Alternate Flow | The system will validate the information provided. If any invalid data is found, the input form will be redirected with an error message. |
| Validation | <ol style="list-style-type: none"> 1. Name is required. 2. The Email should be valid. 3. Password should be matched with Re-Enter Password. |
| Pre-Conditions | Users should have network access. |
| Post-Conditions | Success message should be shown. |

Create an Account'."/>

A Web Page

https://Sign In

Welcome Back!

Please Sign In To Continue

Username or email address

Enter Your Username or Email

Password

Enter Your Password

[Forgot Password ?](#)

Don't have an account? [Create an Account](#)

| Brief Description | Student Sign In |
|--------------------------|--|
| Basic Flow | <p>This use case describes how a student sign-in in to the system</p> <ol style="list-style-type: none"> 1. The student has to sign up himself into the system. 2. After the successful sign in, users will be taken to the appropriate landing page. 3. The following information is required to login. <ul style="list-style-type: none"> · Email address · Password |
| Alternate Flow | The system will validate the credentials provided. If credentials are invalid, the login form will be redirected again with an error message. |
| Validation | <ol style="list-style-type: none"> 1. Valid Email address 2. Valid password |
| Pre-Conditions | Users should have network access and Browser with latest updates. |
| Post-Conditions | Landing page has to be displayed. |

| Brief Description | Add Course Material |
|--------------------------|--|
| Basic Flow | <p>This use case describes how a lecturer upload a new course material in to the system</p> <ol style="list-style-type: none"> 1. The lecturer has to login into the system. 2. The following information is required during add course material. <ul style="list-style-type: none"> • Course name • Course Code • Start Date • End Date optional fields: • description |
| Alternate Flow | The system will validate the information provided. If any invalid data is found, the input form will be redirected with an error message. |
| Validation | course name, course code, start date and end date cannot be empty during form submission. |

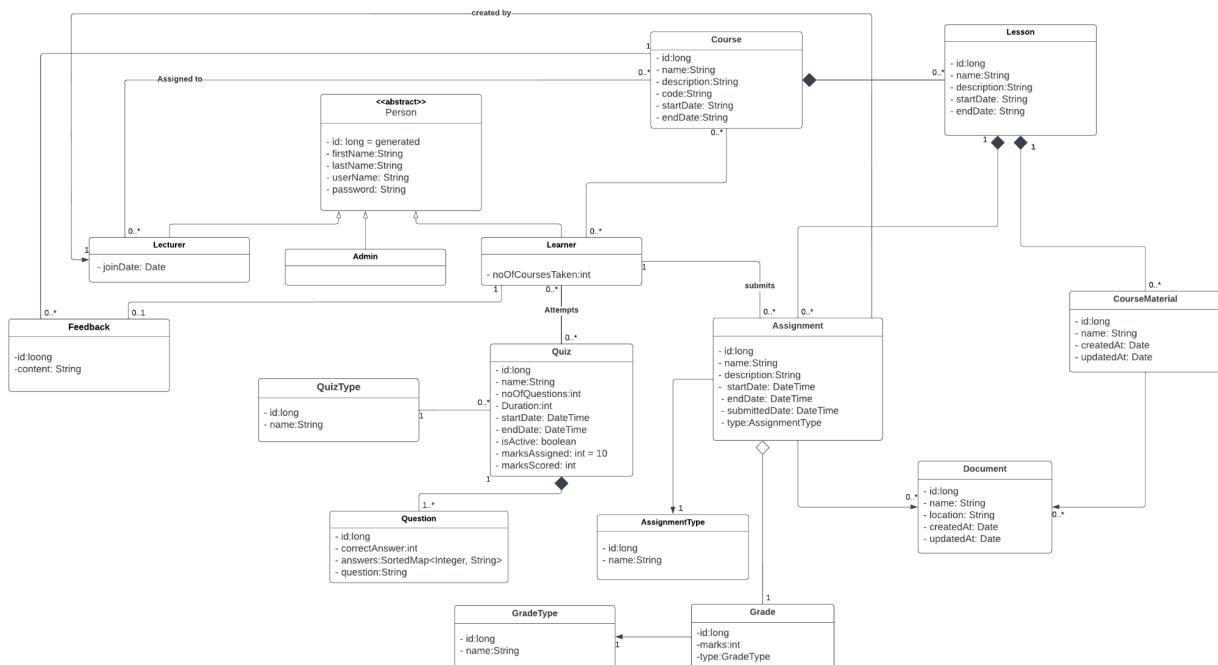
| | |
|-----------------|--|
| Pre-Conditions | lecturer should have already logged into the system. |
| Post-Conditions | Success message should be shown. |

| Brief Description | Add Assignment |
|-------------------|--|
| Basic Flow | <p>This use case describes how a lecturer upload a new assignment in to the system</p> <ol style="list-style-type: none"> 1. The lecturer has to login into the system. 2. The following information is required during the add assignment. <ul style="list-style-type: none"> Course name |
| Alternate Flow | Not Applicable |
| Validation | Not Applicable |
| Pre-Conditions | lecturer should have already logged into the system. |
| Post-Conditions | Success message should be shown. |

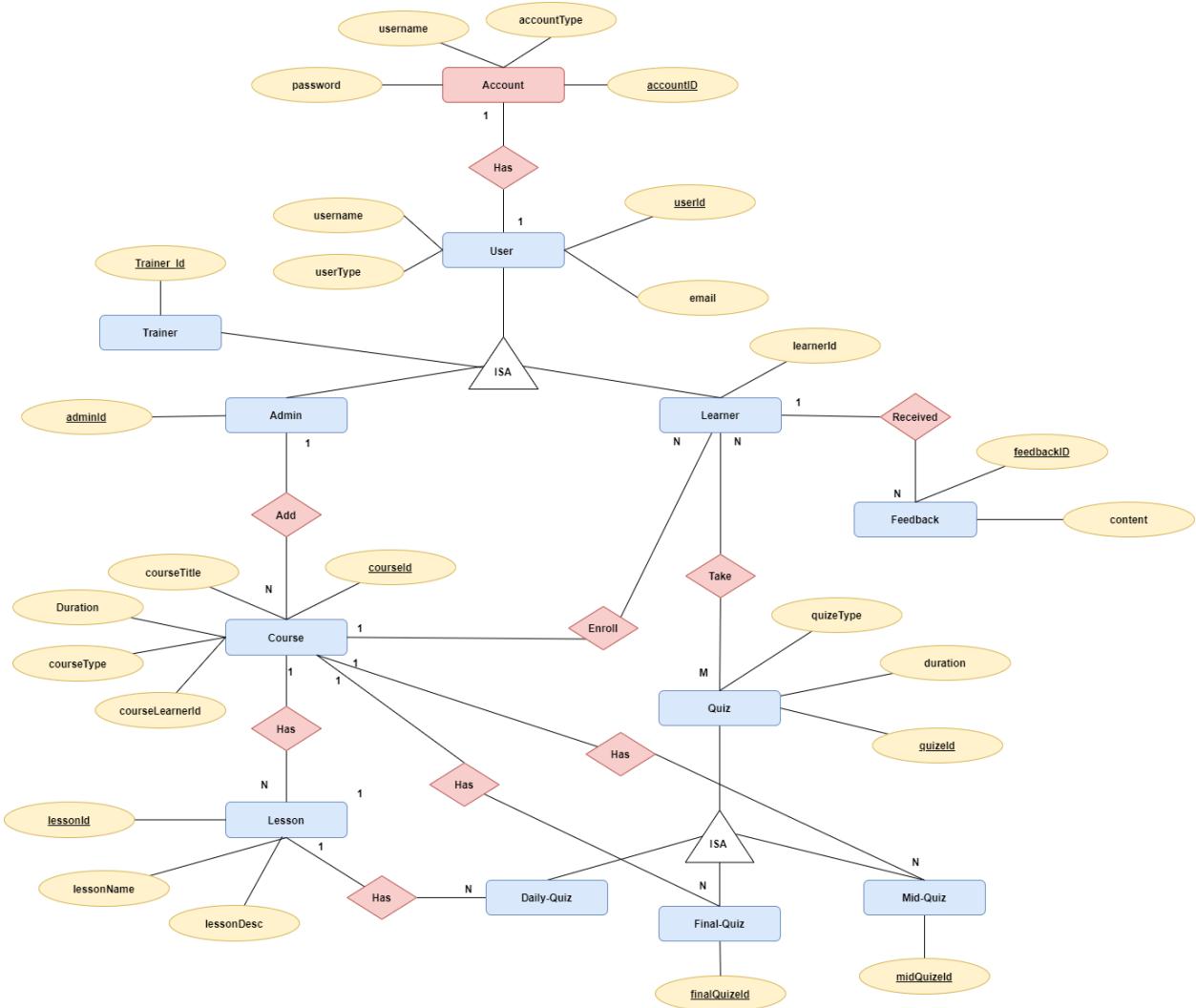
| Brief Description | Create Quiz |
|-------------------|--|
| Basic Flow | <p>This use case describes how a lecturer upload a new quiz in to the system</p> <ol style="list-style-type: none"> 1. The lecturer has to login into the system. 2. The following information is required during the upload invoice. <ul style="list-style-type: none"> Course name |
| Alternate Flow | Not Applicable |
| Validation | Not Applicable |
| Pre-Conditions | lecturer should have already logged into the system. |
| Post-Conditions | Success message should be shown. |

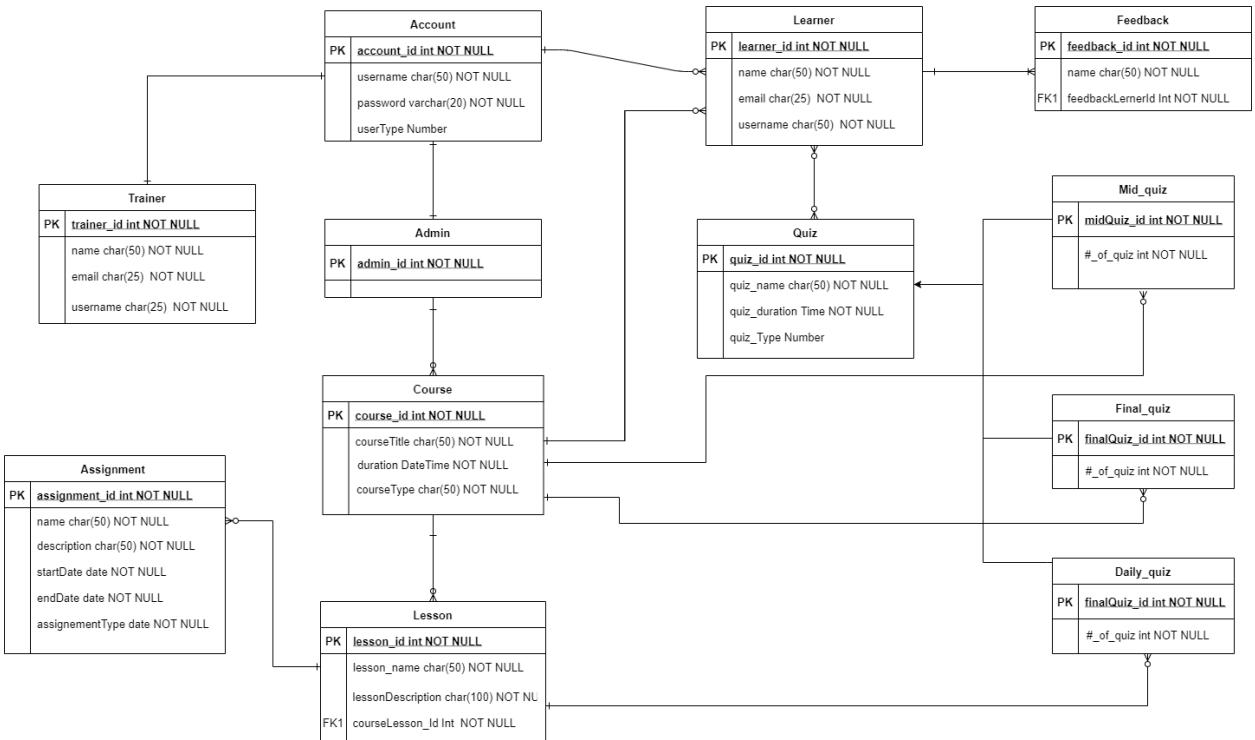
| Brief Description | View Course Material |
|-------------------|---|
| Basic Flow | <p>This use case describes how a student view course material in the system</p> <ol style="list-style-type: none"> 1. The student has to login into the system. 2. Student should navigate to the course page 3. Student could view course material under each lesson of a specific course 4. Students could download course material by clicking on the course material. |
| Alternate Flow | Not Applicable |
| Validation | Not Applicable |
| Pre-Conditions | Students should have already logged into the system. |
| Post-Conditions | Not Applicable. |

4.1.3 Class Diagram

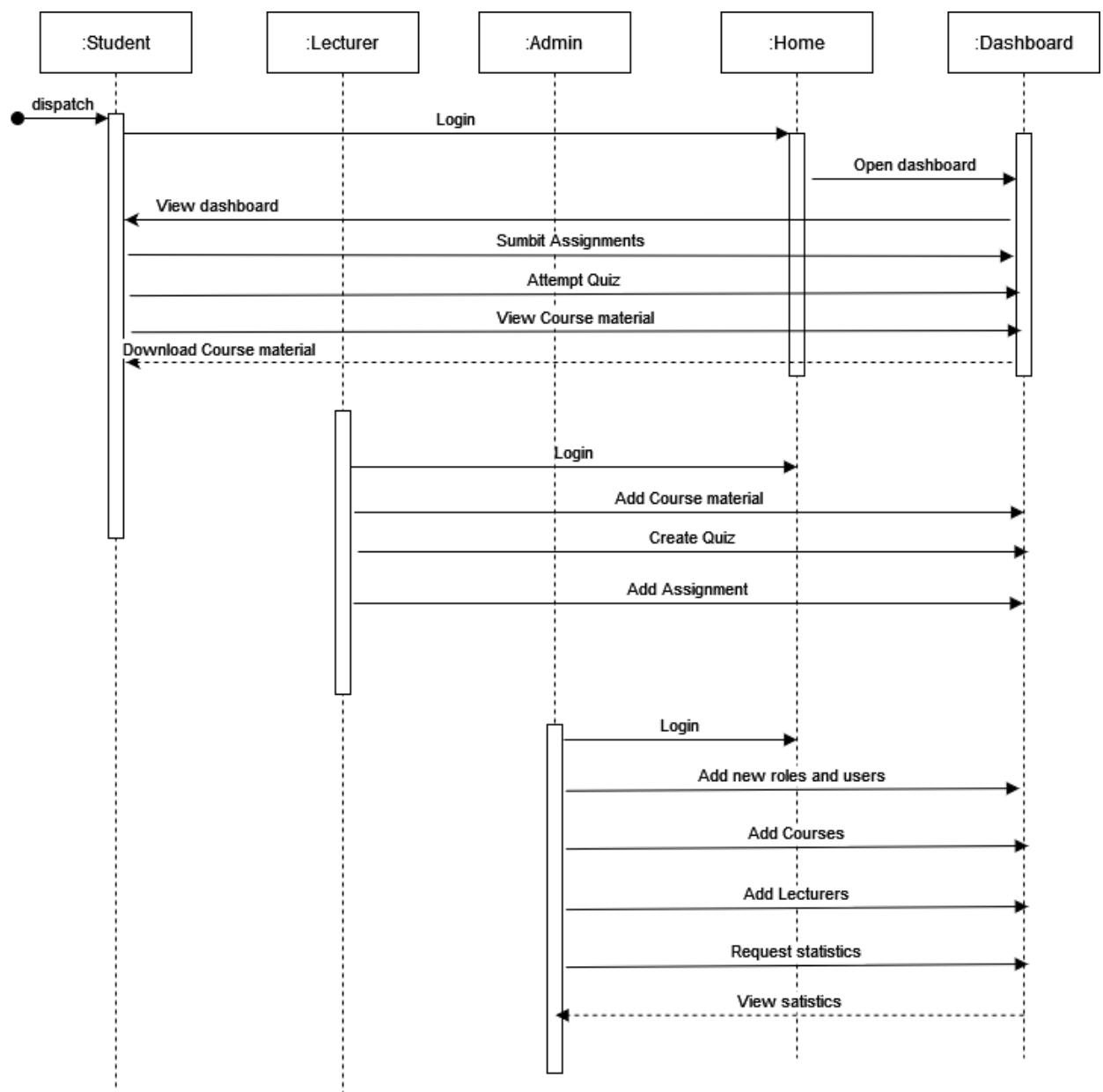


4.1.4 ER Diagram





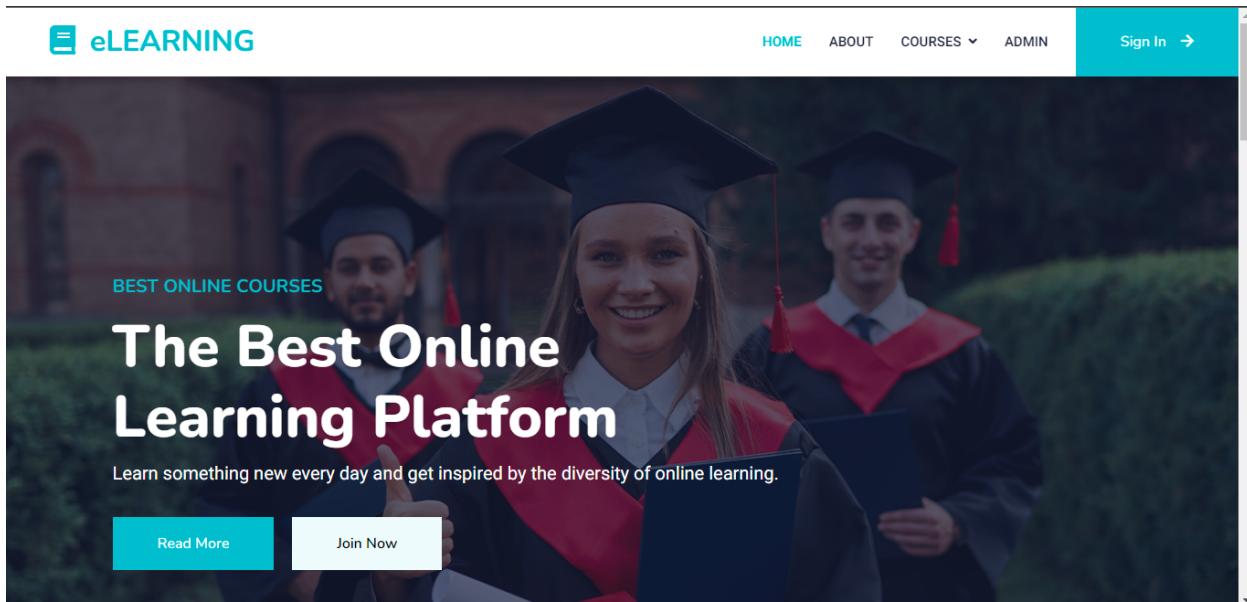
4.1.5 Sequence diagram



4.2 Proposed user Interface design

The following depicts the various interfaces available to the user through this story and module.

Home Page



Sign In Page

The following depicts the form with validation which is used to sign in to the system

A screenshot of a sign-in form. At the top, there are two buttons: "Sign in" (highlighted in blue) and "Sign up". Below the buttons is a field labeled "Email address" with a red border, indicating an error. A small red circle with a question mark icon is to the right of the input field. Below the field is a note: "We'll never share your email with anyone else." Underneath that, an error message says: "You entered wrong email formate or you haven't entered." Below the email field is a password field with a red border and a small red circle with a question mark icon to its right. To the left of the password field is the label "Password". At the bottom left is a checkbox labeled "Check me out" with a red border. At the bottom right is a green "Submit" button.

Sign Up Page

The following depicts the form with validation which is used to sign in to the system

Sign in Sign up

First Name
 Ex- Chathuranga
Invalid First Name!

Last Name
 Ex- Tennakoon
Invalid Last Name!

User Name
 Ex- Bandara

Email address
 Exqwertyui@gmail.com
Invalid Email address!

Password

Re-Enter Password

Check me out

About Page

 [eLEARNING](#) [HOME](#) [ABOUT](#) [COURSES](#) [CONTACT](#) [Sign out](#)



[ABOUT US](#) [Welcome to eLEARNING](#)

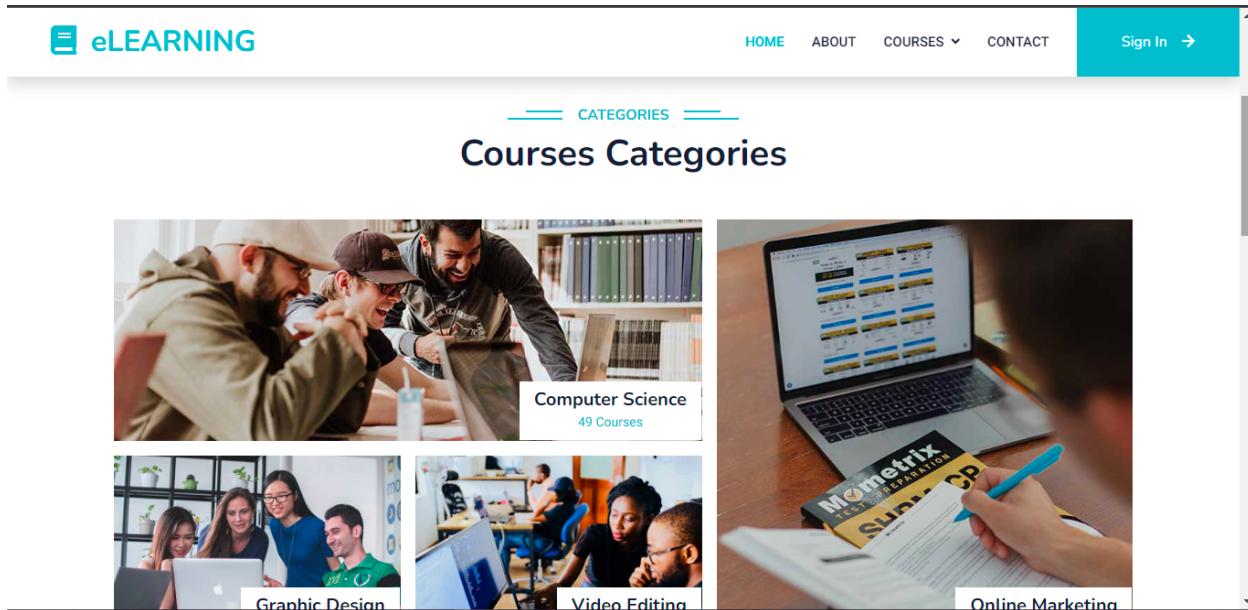
Online learning offers great educational opportunities to students who seek alternatives to attendance on campus due to scheduling conflicts, child-care, work, or other commitments. Students throughout the world are now able to pursue a high quality college education without the challenges of travel and relocation.

Take a moment to review our current online course schedule. We offer a great range of online courses that meet career and transfer needs. As you consider whether online learning is for you, keep in mind that online courses may require more from you, in terms of motivation, discipline, reading, and time.

→ Skilled Instructors → Online Classes
→ International Certificate → Skilled Instructors
→ Online Classes → International Certificate

[Read More](#) 

Courses page



The screenshot shows the 'Courses Categories' page of an eLearning website. At the top, there's a navigation bar with 'eLEARNING' on the left, 'HOME', 'ABOUT', 'COURSES', 'CONTACT', and 'Sign In' on the right. Below the navigation is a section titled 'CATEGORIES' with the heading 'Courses Categories'. There are four main categories displayed as cards:

- Computer Science**: 49 Courses. Shows two men working together on a laptop.
- Graphic Design**. Shows three people looking at a laptop screen.
- Video Editing**. Shows two people working on a video editing project.
- Online Marketing**. Shows a person writing in a notebook with a laptop displaying a marketing dashboard in the background.

Quizzes page



The screenshot shows the 'Quizzes' page of the eLearning website. The layout is similar to the courses page, with a navigation bar at the top. On the left, there's a sidebar with links to various quizzes:

- HOME
- JEE Mains
- JEE Advanced
- GATE
- C/C++
- Java
- Python
- JavaScript
- Data Structures
- Algorithm
- Interview Questions

The main content area features a large, central illustration of a smiling man pointing upwards with both hands. The background of the illustration is a hand-drawn style with various educational icons and text elements, such as 'ONLINE EDUCATION!', 'ART AND DESIGN', 'RESEARCH!', 'SCIENCE!', 'GEOGRAPHY', 'ONLINE', and 'SOLAR SYSTEM'. There is also a small text block in the bottom left corner of the illustration area.

Contact page

The screenshot shows a contact form titled "Contact For Any Query". The form includes fields for "Your Name" and "Your Email", a "Subject" field, a large "Message" area, and a "Send Message" button. To the left, there's a "Get In Touch" section with icons for Office (location), Mobile (phone), and Email (envelope), along with their respective details: 123 Street, New York, USA; +012 345 67890; and info@example.com.

Admin page

The screenshot shows the eLEARNING admin dashboard. On the left, there's a sidebar with links: "Dashboard" (highlighted with a red "Add New" button), "Course Material", "lecture", "Assignment", "Feedback", and "Document". The main content area is currently empty.

Add New Course

The following depicts the form with validation which is used to get course details and all course details available in the system.

Courses

| | |
|-----------------------------------|-----------------------------------|
| Course Name | Course Code |
| <input type="text"/> ⓘ | <input type="text"/> ⓘ |
| Course Name cannot be empty | |
| Start Date | End Date |
| <input type="text"/> dd/mm/yyyy ⓘ | <input type="text"/> dd/mm/yyyy ⓘ |
| start date cannot be empty | |
| Description | |
| <input type="text"/> | |
| optional | |
| <button>Add Course</button> | |

| # | Name | Code | Duration (days) | Start Date | End Date | Actions |
|---|------|------|-----------------|---------------------|---------------------|---|
| 1 | java | JAVA | 56 | 2022-03-04 00:00:00 | 2022-04-29 00:00:00 | <button>Update</button> <button>View</button> |
| 2 | css | CSS | 12 | 2022-04-16 00:00:00 | 2022-04-28 00:00:00 | <button>Update</button> <button>View</button> |
| 3 | html | HTML | 4 | 2022-05-05 00:00:00 | 2022-05-09 00:00:00 | <button>Update</button> <button>View</button> |

3 records.

After Addition of the new Course the success message is shown.

Courses

| | |
|--|-----------------------------------|
| Course Name | Course Code |
| <input type="text"/> | <input type="text"/> |
| Start Date | End Date |
| <input type="text"/> dd/mm/yyyy ⓘ | <input type="text"/> dd/mm/yyyy ⓘ |
| Description | |
| <input type="text"/> | |
| optional | |
| ✓ Added Successfully × <button>Add Course</button> | |

| # | Name | Code | Duration (days) | Start Date | End Date | Actions |
|----|-------------|--------|-----------------|---------------------|---------------------|---|
| 1 | java | JAVA | 56 | 2022-03-04 00:00:00 | 2022-04-29 00:00:00 | <button>Update</button> <button>View</button> |
| 2 | css | CSS | 12 | 2022-04-16 00:00:00 | 2022-04-28 00:00:00 | <button>Update</button> <button>View</button> |
| 3 | html | HTML | 4 | 2022-05-05 00:00:00 | 2022-05-09 00:00:00 | <button>Update</button> <button>View</button> |
| 19 | Java Script | JS_WEB | 14 | 2022-04-15 00:00:00 | 2022-04-29 00:00:00 | <button>Update</button> <button>View</button> |

4 records.

Course details page depicts a specific course and all the lessons available under the course. The course material available under each lesson could be downloaded by the user.

java

JAVA (2022-04-20 - 2022-05-19)

induction training

Add Lesson

Hibernate

2022-04-15 - 2022-04-19

ORM and Java compatible data types are discussed

Course Material

Lecture Slides

[discourse_markers.docx](#)

videos

[use-case-diagram.png](#)

[Use Case Diagram for E-Learning.\(1\).png](#)

Java Core

2022-04-15 - 2022-04-19

Addition of lessons could be done for a specific course. The user could upload course material documents through this form..

Add New Lesson

Lesson Name

Course Name

Lesson Name cannot be empty

Start Date

End Date

Description

optional

Add Course Material

Name

Files

Add

Name

Files

Add

Go Back

Add Lesson form with validation could be depicted as follows.

Add New Lesson

Lesson Name java Lesson Name cannot be empty

Course Name java

Start Date dd/mm/yyyy start date cannot be empty End Date dd/mm/yyyy end date cannot be empty

Description

optional

Add Course Material

Name ✓ Files Choose Files No file chosen ✓ Add

Add Go Back

A Course could be updated as follows.

Update Course Information

Course Name java Course Code JAVA

Start Date 12/03/2022 End Date 23/04/2022

Description A server side language with multiple functionalities. Introduced in 1992. Developed by Team Green.

optional

Update Go Back

✓ Updated Successfully X

Add New Feedback

The following depicts the form which is used to sign in to the system

ADD YOUR FEEDBACK

Id

Name ! Please fill out this field.

Course

Feedback

submit your feedback

Learners Feedbacks

| ID | Name | Course | Feedback |
|----|------|--------|----------|
|----|------|--------|----------|

4.3 Database design changes

When implementing the E-Learning database, mainly consider the mapping that includes the regular entities, ISA Hierarchy, binary relationships with key constraints, final types, and the tables.

Logical Database creation

```
SQL*Plus
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> create type admin_type as object(
  2   adminId number(10),
  3   firstName varchar2(25) ,
  4   lastName varchar2(25),
  5   username varchar2(25),
  6   email varchar2(50),
  7   password varchar2(10)
  8 )
 9 )
10 /
Type created.

SQL> create table admin of admin_type(
  2   CONSTRAINT admin_pk PRIMARY KEY (adminId)
  3 )
 4 /
Table created.

SQL> create type learner_type  as object(
  2   learnerId number(19),
  3   firstName varchar2(25) ,
  4   lastName varchar2(25),
  5   userName varchar2(25),
  6   email varchar2(50),
  7   password varchar2(10),
  8   noOfCoursesTaken number(10)
  9 )
10 /
Type created.

SQL> create table learner of learner_type (
  2   CONSTRAINT learner_pk PRIMARY KEY (learnerId)
  3 )
 4 /
Table created.

SQL> create type trainer_type as object(
  2   trainerId number(19),
  3   firstName varchar2(25) ,
  4   lastName varchar2(25),
  5   userName varchar2(25),
  6   email varchar2(50)
```

```
SQL*Plus
Type created.

SQL> create table trainer of trainer_type(
  2  CONSTRAINT trainer_pk PRIMARY KEY (trainerId)
  3  )
  4  /
Table created.

SQL> create table course(
  2  courseId number(19),
  3  courseName varchar2(100),
  4  description varchar2(225),
  5  code varchar2(10),
  6  duration date,
  7  startDate date,
  8  endDate date,
  9  CONSTRAINT course_pk PRIMARY KEY (courseId)
 10  )
 11  /
Table created.

SQL>
SQL> create table lesson(
  2  lessonId number(19),
  3  lessonName varchar2(100),
  4  description varchar2(225),
  5  duration date,
  6  startDate date,
  7  endDate date,
  8  lesCourseid number(19),
  9  CONSTRAINT lesson_pk PRIMARY KEY (lessonId),
 10  CONSTRAINT fk_course FOREIGN KEY (lesCourseid) REFERENCES course(courseId) ON DELETE CASCADE
 11  )
 12  /
Table created.

SQL> create table courseMaterial(
  2  materialId number(19),
  3  name varchar2(100),
  4  createdAt date,
  5  updatedAt date,
  6  lessonMaterialId number(19),
  7  CONSTRAINT courseMaterial_pk PRIMARY KEY (materialId),
  8  CONSTRAINT fk_lesson_material FOREIGN KEY (lessonMaterialId) REFERENCES lesson(lessonId) ON DELETE CASCADE
  9  )
 10  /

```

```
SQL*Plus
SQL> create table assignmentType(
  2  id number(19),
  3  name varchar2(100),
  4  CONSTRAINT assignmentType_pk PRIMARY KEY (id)
  5  )
  6  /
Table created.

SQL> create table assignment(
  2  assignId number(19),
  3  name varchar2(100),
  4  description varchar2(225),
  5  startDate date,
  6  endDate date,
  7  submittedDate date,
  8  trainer_assignId number(19),
  9  lesson_assignId number(19),
 10  assignTypeId number(19),
 11  CONSTRAINT assignment_pk PRIMARY KEY (assignId),
 12  CONSTRAINT fk_trainer_assign FOREIGN KEY (trainer_assignId) REFERENCES trainer(trainerId) ON DELETE CASCADE,
 13  CONSTRAINT fk_lesson_assign FOREIGN KEY (lesson_assignId) REFERENCES lesson(lessonId) ON DELETE CASCADE,
 14  CONSTRAINT fk_assigntype FOREIGN KEY (assignTypeId) REFERENCES assignmentType(id) ON DELETE CASCADE
 15  )
 16  /
Table created.

SQL> create table document(
  2  id number(19),
  3  name varchar2(100),
  4  createdAt date,
  5  updatedAt date,
  6  location varchar2(225),
  7  docAssignId number(19),
  8  docMaterialId number(19),
  9  CONSTRAINT document_pk PRIMARY KEY (id),
 10  CONSTRAINT fk_doc_assign FOREIGN KEY (docAssignId) REFERENCES assignment(assignId) ON DELETE CASCADE,
 11  CONSTRAINT fk_doc_material FOREIGN KEY (docMaterialId) REFERENCES courseMaterial(materialId) ON DELETE CASCADE
 12  )
 13  /
Table created.

SQL> create table gradeType(
  2  id Number(19) Not Null,
  3  name varchar2(100) Not Null,
  4  CONSTRAINT gradeType_pk PRIMARY KEY (id)
```

```
SQL*Plus

Table created.

SQL> create table feedback(
  2  id number(19) Not Null,
  3  content varchar2(225) Not Null,
  4  courseFdbid number(19),
  5  learnerFdbid number(19),
  6  CONSTRAINT feedback_pk PRIMARY KEY (id),
  7  CONSTRAINT fk_feedbackLearner FOREIGN KEY (learnerFdbid) REFERENCES learner(learnerId) ON DELETE CASCADE,
  8  CONSTRAINT fk_feedbackCourse FOREIGN KEY (courseFdbid) REFERENCES course(courseId) ON DELETE CASCADE
  9
 10 /
Table created.

SQL> create table assignTo(
  2  coid number(19) Not Null,
  3  learnid number (19) Not Null,
  4  CONSTRAINT assignTo_pk PRIMARY KEY (coid , learnid),
  5  CONSTRAINT fk_assignTolearner FOREIGN KEY (learnid) REFERENCES learner(learnerId) ON DELETE CASCADE,
  6  CONSTRAINT fk_assignToCourse FOREIGN KEY (coid) REFERENCES course(courseId) ON DELETE CASCADE
  7
 8 /
Table created.

SQL> create table submit(
  2  id number(19) Not Null,
  3  tid number(19) Not Null,
  4  CONSTRAINT submit_pk PRIMARY KEY (id, tid),
  5  CONSTRAINT fk_submit FOREIGN KEY (tid) REFERENCES trainer(trainerId) ON DELETE CASCADE,
  6  CONSTRAINT fk_submitCourse FOREIGN KEY (id) REFERENCES course(courseId) ON DELETE CASCADE
  7
 8 /
Table created.

SQL> create table createdBy(
  2  aid number(19) Not Null,
  3  lid number(19) Not Null,
  4  CONSTRAINT createdBy_pk PRIMARY KEY (aid ,lid),
  5  CONSTRAINT fk_createdByassign FOREIGN KEY (aid) REFERENCES assignment(assignid) ON DELETE CASCADE,
  6  CONSTRAINT fk_acreatedBylearner FOREIGN KEY (lid) REFERENCES learner(learnerId) ON DELETE CASCADE
  7
 8 /
Table created.

SQL>
```

```
SQL*Plus

SQL> create table notice(
  2  id number(19) Not Null,
  3  title varchar2(100) Not Null,
  4  description varchar2(225),
  5  createdBy varchar2(100),
  6  position varchar2(100),
  7  postdate DATE,
  8  createId number(19),
  9  CONSTRAINT notice_pk PRIMARY KEY (id),
 10  CONSTRAINT fk_notice FOREIGN KEY (createId) REFERENCES trainer(trainerId) ON DELETE CASCADE
 11
 12 /
Table created.

SQL>
```

Database Table

```
SQL>
SQL>
SQL>
SQL>
SQL> Desc admin
Name          Null?    Type
-----          -----
ADMINID      NOT NULL NUMBER(19)
FIRSTNAME    VARCHAR2(25)
LASTNAME     VARCHAR2(25)
USERNAME     VARCHAR2(25)
EMAIL        VARCHAR2(50)
PASSWORD     VARCHAR2(10)

SQL> desc learner
Name          Null?    Type
-----          -----
LEARNERID    NOT NULL NUMBER(19)
FIRSTNAME    VARCHAR2(25)
LASTNAME     VARCHAR2(25)
USERNAME     VARCHAR2(25)
EMAIL        VARCHAR2(50)
PASSWORD     VARCHAR2(10)
NOOFCOURSESTAKEN NUMBER(10)

SQL> Desc trainer
Name          Null?    Type
-----          -----
TRAINERID    NOT NULL NUMBER(19)
FIRSTNAME    VARCHAR2(25)
LASTNAME     VARCHAR2(25)
USERNAME     VARCHAR2(25)
EMAIL        VARCHAR2(50)
PASSWORD     VARCHAR2(10)

SQL> Desc lesson
Name          Null?    Type
-----          -----
LESSONID      NOT NULL NUMBER(19)
LESSONNAME    VARCHAR2(100)
DESCRIPTION   VARCHAR2(225)
DURATION     DATE
STARTDATE    DATE
ENDDATE      DATE
LESSOURCEID  NUMBER(19)

SQL> Desc course
Name          Null?    Type
```

```
SQL> Desc course
Name          Null?    Type
-----          -----
COURSEID      NOT NULL NUMBER(19)
COURSENAME    VARCHAR2(100)
DESCRIPTION   VARCHAR2(225)
CODE          VARCHAR2(10)
DURATION     DATE
STARTDATE    DATE
ENDDATE      DATE

SQL> Desc courseMaterial
Name          Null?    Type
-----          -----
MATERIALID   NOT NULL NUMBER(19)
NAME         VARCHAR2(100)
CREATEDAT    DATE
UPDATEDAT    DATE
LESMMATERIALID NUMBER(19)

SQL> Desc assignmentType
Name          Null?    Type
-----          -----
ID           NOT NULL NUMBER(19)
NAME         VARCHAR2(100)

SQL> Desc assignment
Name          Null?    Type
-----          -----
ASSIGNID     NOT NULL NUMBER(19)
NAME         VARCHAR2(100)
DESCRIPTION  VARCHAR2(225)
STARTDATE    DATE
ENDDATE      DATE
SUBMITTEDDATE DATE
TRAINER_ASSIGNID NUMBER(19)
LESSON_ASSIGNID NUMBER(19)
ASSIGNTYPEID  NUMBER(19)

SQL> Desc document
Name          Null?    Type
-----          -----
ID           NOT NULL NUMBER(19)
NAME         VARCHAR2(100)
CREATEDAT    DATE
UPDATEDAT    DATE
LOCATION     VARCHAR2(225)
DOCASSIGNID  NUMBER(19)
DOC MATERIALID NUMBER(19)
```

```

SQL> Desc document
Name          Null?    Type
-----        -----
ID           NOT NULL NUMBER(19)
NAME          VARCHAR2(100)
CREATEDAT    DATE
UPDATEDAT    DATE
LOCATION     VARCHAR2(225)
DOCASSIGNID  NUMBER(19)
DOCATERIALID NUMBER(19)

SQL> Desc greadetype
Name          Null?    Type
-----        -----
ID           NOT NULL NUMBER(19)
NAME          NOT NULL VARCHAR2(100)

SQL> Desc grade
Name          Null?    Type
-----        -----
ID           NOT NULL NUMBER(19)
MARK          NOT NULL NUMBER(10)
TYPEID        NUMBER(19)

SQL> Desc feedback
Name          Null?    Type
-----        -----
ID           NOT NULL NUMBER(19)
CONTENT       NOT NULL VARCHAR2(225)
COURSEFOBID   NUMBER(19)
LEARNERFOBID  NUMBER(19)

SQL> Desc notice
Name          Null?    Type
-----        -----
ID           NOT NULL NUMBER(19)
TITLE         NOT NULL VARCHAR2(100)
DESCRIPTION   VARCHAR2(225)
POSTEDBY     VARCHAR2(100)
POSITION      VARCHAR2(100)
POSTDATE     DATE
CREATEID     NUMBER(19)

SQL>

```

4.4 Refactoring related changes

The code is currently refactored using the best practices as mentioned below.

1. Usage of cases
 - Class Names: Pascal Case
 - Methods and Variable Names: Camel Case
 - Constants: Upper Case
2. String literals used commonly are applied in **Constants.java** class as constants.
3. URL patterns used are initialised in a central location named **UrlConstants.java** class as constants.
4. Each method is provided through an interface enabling abstraction.
5. The classes and interfaces are arranged in packages based on functionality as follows.

```
< M&J > ELearning [e-learning add_course]
  > Deployment Descriptor: Archetype Created Web Application
  > src/main/java
    > com.elearning.controller
    > com.elearning.dao
    > com.elearning.dao.impl
    > com.elearning.model
    > com.elearning.repository
    > com.elearning.service
    > com.elearning.servicimpl
    > com.elearning.util
```

6. Utility functions such as date formatting, exception handling, declaring url patterns, generating folder names for file upload and classes containing request status are added to **com.elearning.util** package.

```
> com.elearning.util
  > ApplicationException.java
  > Constants.java
  > ELearningDateFormatter.java
  > SuccessStatus.java
  > UploadFolderNameFormatter.java
  > UrlConstants.java
```

4.5 Construction strategy and re-use

All Components of the system are created manually.

1. The following classes involving commonly used utility functions such as date formatting, exception handling, declaring url patterns, generating folder names for file upload and classes containing request status are applied in the project.

```
✓ com.elearning.util
  > ApplicationException.java
  > Constants.java
  > ELearningDateFormatter.java
  > SuccessStatus.java
  > UploadFolderNameFormatter.java
  > UrlConstants.java
```

2. All modules of the project implementation are preceded by interface declaration of the methods enabling reuse.
 - a. **com.elearning.dao** package corresponds to interfaces containing data access related method declarations whereas the **com.elearning.dao.impl** package corresponds to classes containing corresponding data access related method definitions.
 - b. **com.elearning.service** package corresponds to interfaces containing the business logic related method declarations whereas the **com.elearning.service.impl** package corresponds to implementing classes containing the business logic related method definitions.

5 Details of Alternative Design Approach

There is no alternative design approach.

6 Other Technical changes

6.1 Automation tasks

Automation is not included as it is not in the scope of the project.

6.2 CI / Build relates tasks

The project is implemented using a version control system named GitHub enabling continuous integration. The GitHub link for the project is as follows.

<https://github.com/himashaperera/e-learning.git>

6.3 Non-functional related changes

"Any requirement that specifies how the system performs a certain function," says the definition of a non-functional requirement. In other words, a non-functional requirement will specify how a system should behave as well as its functional limitations. Why are non-functional requirements important if the product's functionality is not dependent on them? The answer lies in the usability of the product. Non-functional requirements have an impact on the user experience because they define the behaviour, features, and general characteristics of a system. When non-functional requirements are well defined and implemented, the system will be easier to use and perform better.

- Performance - the application has a response time of less than 1 second
- Reliability - the system handles possible exceptions and able to recover from possible issues
- Manageability - the code is arranged according to the best practices enabling readability and maintainability
- Data Integrity - all transactions are performed according to the ACID rules preserving data integrity
- Usability - an attractive UIs are included into improve user experience

7 Additional details

7.1 Open Questions / clarifications / Assumptions

Assumptions:

- Admin adds all users namely: new Admins, Learners and Trainers to the system
- Users can attempt quizzes only once
- The quiz score would be generated after submission of the quiz.
- Learners could view abrupt details of the courses available before registration.
- Trainers and Admins could add new courses to the elearning system.
- A new course could be added with lessons containing course material such as images, videos and documents.

7.2 Additional notes to technical team

Please set up the Project using the below instructions

1. Please clone the project from <https://github.com/himashaperera/e-learning.git>.
2. Import the project into the eclipse workspace as a maven project.
3. Change the **UPLOADED_FOLDER** from `Constants.java` in `com.elearning.util` package to a location in your PC.
4. Install ORACLE SQL Server XE edition.
5. Open spring-servlet.xml file in src/main/webapp/WEB-INF folder. In a bean named “dataSource” change property field values of username and password to your oracle DB installation.
6. Update the maven project so install the relevant dependencies mentioned on pom.xml
7. Install Apache Tomcat Server 9.0 and run the project using the following steps.
 - Right click on the project and select **Run As** and then choose **Run on Server**
 - Select the server Apache Tomcat 9.0
 - Add the project to Server and click finish

8 References

1. Bootstrap 5.0 Documentation: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
2. Spring Framework Documentation:
<https://docs.spring.io/spring-framework/docs/current/reference/html/>
3. Java Documentation: <https://docs.oracle.com/javase/8/docs/>
4. Downloading and Uploading Files using Spring MVC (visited on 7th April 2022):
<https://www.baeldung.com/java-download-file>
<https://www.baeldung.com/spring-file-upload>