**UDEMY REPLICA**



*Developed by:*

**Ameer Shahroz khan**

3785-FBAS/BSSE/F18

**Umair Khalid**

3895-FBAS/BSSE/F18

*Supervised By:*

**Mr Zulqarnain**

**Department of Computer Science and Software Engineering**

**International Islamic University, Islamabad**

**2022**

# FINAL APPROVAL

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

It is certified that I have read the project report titled “UDEMY REPLICA” submitted by **Ameer Shahroz Khan (3785-FBAS/BSSE/F18)** and **Umair Khalid (3895-FBAS/BSSE/F18)**. This project meets the necessary criteria for acceptance by the International Islamic University, Islamabad, for a bachelor’s degree in Software Engineering.

**Supervisor**

**Mr. Zulqarnain**

Assistant Professor

Department of Computer Science & Software engineering

International Islamic University, Islamabad \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Examiner**

**Dr. Muhammad Nasir**

Assistant Professor

Department of Computer Science & Software engineering

International Islamic University, Islamabad \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**External Examiner**

**Mr. Islam Zada**

Assistant Professor

Department of Computer Science & Software engineering

International Islamic University, Islamabad \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# DECLARATION

I hereby declare that this application, neither as a whole nor as a part there of has been copied out from any source, it is further declared that I developed this application entirely on the basis of our personal efforts made under the sincere guidance of my supervisor and teachers.

No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other institute of learning.

**Umair Khalid**

**3895-FBAS/BSSE/F18**

And

**Ameer Shahroz Khan**

**3785-FBAS/BSSE/F18**

# ACKNOWLEDGMENT

I would like to thank my project supervisor **Mr. Zulqarnain** for his continuous guidance and support.

Finally, I would like to thank all the teachers of faculty who taught me during the course of 4 years in university. Without their guidance and knowledge, I would not be on this step today.

# PROJECT IN BRIEF

|  |  |
| --- | --- |
| **Tile:** | Udemy Replica |
| **Undertaken By:** | Umair Khalid, Ameer Shahroz Khan |
| **Supervised By:** | Mr. Zulqarnain |
| **Started on:** | December 03,2021 |
| **Completed on:** | August 10, 2022 |
| **Tools Used:** | Visual Code 2021, Android Studio 2021, Paradigm, Microsoft Visio |
| **Language:** | Flutter( Dart), MongoDB, NodeJS, |
| **System Used:** | Core i5, 8GB RAM, Windows10 |

**TABLE OF CONTENT**

Contents

[FINAL APPROVAL i](#_Toc111402889)

[DECLARATION ii](#_Toc111402890)

[ACKNOWLEDGMENT iii](#_Toc111402891)

[PROJECT IN BRIEF iv](#_Toc111402892)

[Chapter 1: Introduction 8](#_Toc111402893)

[1.1 Abstract View: 9](#_Toc111402895)

[1.2 Problem Statement: 9](#_Toc111402896)

[1.3 Proposed Solution: 9](#_Toc111402897)

[1.4 Objectives: 10](#_Toc111402898)

[1.5 Project Scope: 10](#_Toc111402899)

[1.6 Modules 10](#_Toc111402900)

[1.7 Tools And Technologies 10](#_Toc111402901)

[Chapter 2: System analysis 11](#_Toc111402902)

[2.1 Introduction to System Analysis: 12](#_Toc111402903)

[2.2 Requirement Analysis: 12](#_Toc111402904)

[2.3 Use-Case Modeling: 12](#_Toc111402905)

[2.3.1 Methodology: 13](#_Toc111402906)

[2.4 Use Case Diagrams: 13](#_Toc111402907)

[2.5 Use Cases in Brief Format: 15](#_Toc111402908)

[2.5.1 Registration 15](#_Toc111402909)

[2.5.2 Login 15](#_Toc111402910)

[2.5.3 Edit Data 15](#_Toc111402911)

[2.5.4 System Maintenances 16](#_Toc111402912)

[2.5.5 Upload Lecture 16](#_Toc111402913)

[2.5.6 Edit Lecture 16](#_Toc111402914)

[2.5.7 Search Courses 17](#_Toc111402915)

[2.5.8 Enroll 17](#_Toc111402916)

[2.5.9 Payment 17](#_Toc111402917)

[2.5.10 Save Course 18](#_Toc111402918)

[2.5.11 Watch Videos 18](#_Toc111402919)

[2.5.12 Rate Courses 18](#_Toc111402920)

[2.6 Use Cases in Extended Format: 19](#_Toc111402921)

[2.6.1 Registration 19](#_Toc111402922)

[2.6.2 Login 20](#_Toc111402923)

[2.6.3 Edit Data 20](#_Toc111402924)

[2.6.4 System Maintenance 21](#_Toc111402925)

[2.6.5 Upload Lectures 21](#_Toc111402926)

[2.6.6 Edit Lectures 22](#_Toc111402927)

[2.6.7 Search Course 22](#_Toc111402928)

[2.6.8 Enroll 23](#_Toc111402929)

[2.6.9 Payment 23](#_Toc111402930)

[2.6.10 Watch video lectures 24](#_Toc111402931)

[2.6.11 Save Lectures 24](#_Toc111402932)

[2.6.12 Rate Course 25](#_Toc111402933)

[2.7 System Sequence Diagrams: 26](#_Toc111402934)

[2.7.1 Student Login 26](#_Toc111402935)

[2.7.2 Student Registration: 27](#_Toc111402936)

[2.7.3 Student Course Enrollment: 28](#_Toc111402937)

[2.7.4 Teacher Upload Permission: 29](#_Toc111402938)

[2.7.6 System Sequence Diagram: 30](#_Toc111402939)

[Chapter 3: System Design 31](#_Toc111402940)

[3.1 Class Diagram: 31](#_Toc111402941)

[3.2 Domain Model: 32](#_Toc111402942)

[3.3 Entity Relationship Diagram (EERD): 33](#_Toc111402943)

[Chapter 4: Implementation 34](#_Toc111402944)

[4.1 Detail Implementation Tools and technologies: 34](#_Toc111402945)

[4.2 Component Diagram: 36](#_Toc111402946)

[4.3 Deployment Diagram: 37](#_Toc111402947)

[Chapter 5: System Testing 38](#_Toc111402948)

[5.1 Methodology: 38](#_Toc111402949)

[5.2 White Box Testing: 38](#_Toc111402950)

[5.3 Black Box Testing: 38](#_Toc111402951)

[5.4 Verification: 39](#_Toc111402952)

[5.5 Validation: 39](#_Toc111402953)

[5.6 Test Cases: 39](#_Toc111402954)

[Table 5.6.1: TC 01 to test the Signup/Registration page. 39](#_Toc111402955)

[Table 5.6.2: TC 02 to test the Sign in/login page. 39](#_Toc111402956)

[Table 5.6.3: TC 03 to test the Home/Main page. 40](#_Toc111402957)

[Table 5.6.4: TC 04 to test the Teacher uploading lectures. 40](#_Toc111402958)

[Table 5.6.5: TC 05 to test the Student download and watch lectures. 41](#_Toc111402959)

[Table 5.6.6: TC 06 to test the logout/sign out session. 41](#_Toc111402960)

[Chapter 6: Conclusion 42](#_Toc111402961)

[6.1 Goals Achieved 42](#_Toc111402962)

[6.2 Future Goals 42](#_Toc111402963)

[USER MANUAL 43](#_Toc111402964)

[1 Sign up Page 43](#_Toc111402965)

[2 Sign in Page 44](#_Toc111402966)

[3 Home Page 45](#_Toc111402967)

[4 User Account 46](#_Toc111402968)

[5 Trending Courses 47](#_Toc111402969)

[6 Course Upload 48](#_Toc111402970)

[7 Admin page 49](#_Toc111402971)

[REFERENCES 50](#_Toc111402972)

# Chapter 1: Introduction

In current era of pandemic and other viruses work from home or remote base work is highly recommended and widely used these days. So, beside where employees are working from home, students also need a platform to learn online by taking seat in their home. So, our project is going to help those students to learn latest technologies or other informative stuff without stepping out of their houses and where teachers would create courses for students and students will get access by enrolling to the specific course.

## .

## 1.1 Abstract View:

The Purpose of this system is to create new possibilities for people and organization everywhere by connecting them to knowledge and skills they need to succeed in a changing world.

By using this system, the students and people of any age will be able to register themselves in the system and then the student would be able to watch video lectures of the enrolled courses. The database and main controls are in the hands of admin.

The students will be able to learn anytime anyplace according to their schedule. Teacher can sign up as a teacher and upload courses. Courses can be paid or free as per teachers will.

## 1.2 Problem Statement:

The Udemy website is very helpful during Covid-19. People learnt new skills online but the main issue is Udemy demanding courses are highly paid and Udemy app is not useful as it needs to be.

The second big issue is courses on Udemy set dates and you have to follow the schedule no matter what problems you have. You are strict to your course deadline. There are no assignments and quizzes in free courses.

## 1.3 Proposed Solution:

To overcome the above described problem, I developed an Android based system through which the students can sign in and select any course and complete it by following their own schedule. The courses will not be highly paid. Our mission is to provide as much low-price courses as student demands.

The admin can manage course data, delete, edit, send notifications through emails. The teachers will be able to upload files in the system corresponding to their course, enrolled student and admin can download and view the files.

The teacher can reward certificates to the students who completed their course. Once student in enrolled the course will be on his/her dashboard and they can watch lectures anytime anywhere.

Students can pay course enrollment fee using easy paisa or jazz cash.

## 1.4 Objectives:

* Access course from your mobile by enroll.
* Discover and search for relevant content
* Student can take video, text lectures online and also download the lectures.
* Student can study in their own comfort zone.
* Student gives quizzes and practical exams from their mobile phones.

## 1.5 Project Scope:

This project consists of several modules for example admin, teacher, students, etc.

The admin is able to manage everything. He can perform almost all of the operations in the system.

The Student can only enroll a course, watch lectures and view the status of his submitted assignments/projects, can ask the teacher for certificate.

## 1.6 Modules

* Course creation
* Couse enrollment
* Course management
* Payment
* Course rating
* Search course

**Modules to be done in future**

* 24/7 service
* Latest courses according to the time change
* Low price courses/Discount
* Scholarships
* Quizzes and assignments
* Certification
* Student and teacher feedback

## 1.7 Tools And Technologies

* Flutter(DART Language)
* Node js
* Express
* MongoDB

## 

# 

# Chapter 2: System analysis

It is a method to process data, analyze problems, and divide the system into components. It makes the process of system analysis much easier than before. It helps in developing a risk free system or product which work more efficient because we work with proper study of system using use case diagrams, sequence diagrams etc. We follow software development models to reduce time, cost, and risk and increase efficiency of our system.

## 2.1 Introduction to System Analysis:

Software Development Life Cycle (SDLC) is a process in Software Engineering that defines various stages and steps involved in development of a high-quality deliverable software.

The different phases of SDLC are given below.

* Requirement elicitation
* Analysis of requirements
* Designing of System
* Implementation
* Verification and Validation
* Testing
* Deployment
* Maintenance

We will see each step of SDLC for our system (Udemy Replica) separately, starting from Requirements elicitation and analysis.

## 2.2 Requirement Analysis:

In the first phase of SDLC, we need clear cut requirements about the system, that what we need to implement in the system and what we don’t. After completing this stage, we have a clear picture of the system in mind and we start on designing the system.

In case of our system, we needed to elicit the requirements from the teachers and students, because these are the end users of the system eventually and the stakeholder too. Online learning platforms also give us necessary information to build this system.

I have held a few meetings with teachers and fellows of the university and I got a bunch of requirements. That I analyzed and designed this system in the following way.

## 2.3 Use-Case Modeling:

The Use-cases of the system describes the proposed functionality of the whole system and the actors that can use certain functionality. In-short, use case modeling is the graphical representation of requirements that we have elicited. This model shows that which actor/user can interact with system in certain ways.

### **2.3.1 Methodology:**

During use case modeling, we have to identify a few elements. That elements help us to get a better understanding of the system.

These elements are

* + - System Boundary, that is the separation between our system and the external world
    - Actors, which are the stakeholder or the individuals that are interacting with the system

While we analyzed, we get that our system will have four actors, i.e. admin, student, teacher.

## 2.4 Use Case Diagrams:

Following are the use case diagrams for each actor separately and then a combined diagram for all the actors of the system. There are three main actors of the system, i.e. Admin, Student and Teacher.

**Use Case Diagrams**

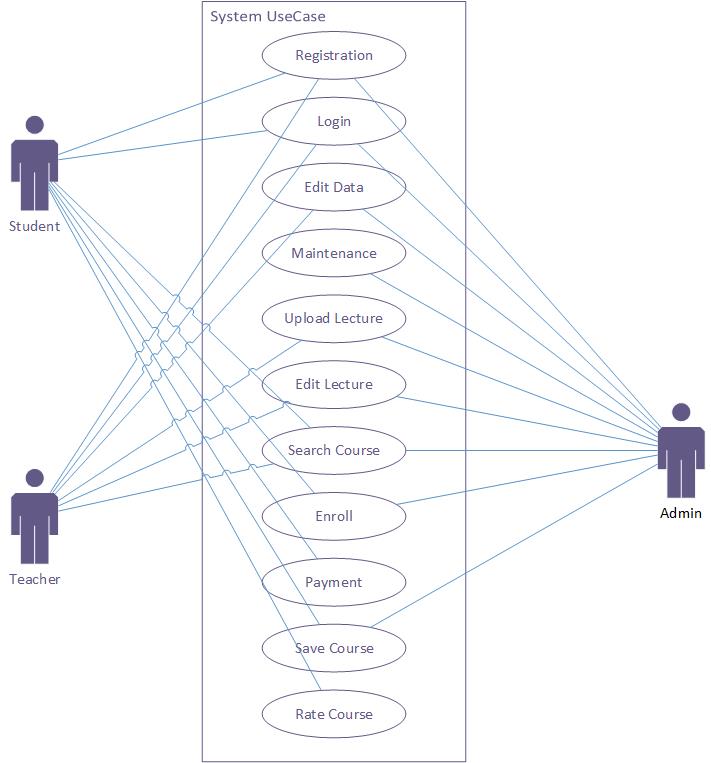


Figure 1

## 2.5 Use Cases in Brief Format:

### **2.5.1 Registration**

Table 1

|  |  |
| --- | --- |
| **Use case Name:** | Registration |
| **Actor:** | Student, Teacher, Admin |
| **Type:** | Primary |
| **Description:** | Student or Teacher need to create an account and need to select the role of teacher or student to access application. |

### **2.5.2 Login**

Table 2

|  |  |
| --- | --- |
| **Use case Name:** | Login |
| **Actor:** | Admin, Student, Teacher |
| **Type:** | Primary |
| **Description:** | Any user (admin, teacher, and student) can login and interact with their corresponding sections based on their roles. |

### **2.5.3 Edit Data**

Table 3

|  |  |
| --- | --- |
| **Use case Name:** | Edit Data |
| **Actor:** | Admin |
| **Type:** | Primary |
| **Description:** | Admin can add, delete, updates data or other kinds of data. |

### **2.5.4 System Maintenances**

Table 4

|  |  |
| --- | --- |
| **Use case Name:** | System Maintenance |
| **Actor:** | Admin |
| **Type:** | Primary |
| **Description:** | Admin have to maintain the whole system |

### **2.5.5 Upload Lecture**

Table 5

|  |  |
| --- | --- |
| **Use case Name:** | Upload Lecture |
| **Actor:** | Teacher |
| **Type:** | Primary |
| **Description:** | Teacher can upload video lectures and other resources for their students. |

### **2.5.6 Edit Lecture**

Table 6

|  |  |
| --- | --- |
| **Use case Name:** | Edit Lecture |
| **Actor:** | Teacher |
| **Type:** | Primary |
| **Description:** | Teacher can added or delete any lecture video, quizzes, assignments notes etc. |

### **2.5.7 Search Courses**

Table 7

|  |  |
| --- | --- |
| **Use case Name:** | Search Courses |
| **Actor:** | Student |
| **Type:** | Primary |
| **Description:** | Student can search courses here what he wants to study. |

### **2.5.8 Enroll**

Table 8

|  |  |
| --- | --- |
| **Use case Name:** | Enroll |
| **Actor:** | Student |
| **Type:** | Primary |
| **Description:** | Student can enroll any selected courses what he wants to study. |

### **2.5.9 Payment**

Table 9

|  |  |
| --- | --- |
| **Use case Name:** | Payment |
| **Actor:** | Student |
| **Type:** | Primary |
| **Description:** | Student can pay money with the given payment method. |

### **2.5.10 Save Course**

Table 10

|  |  |
| --- | --- |
| **Use case Name:** | Save Course |
| **Actor:** | Student |
| **Type:** | Primary |
| **Description:** | Student can save their courses. |

### **2.5.11 Watch Videos**

Table 11

|  |  |
| --- | --- |
| **Use case Name:** | Watch Videos |
| **Actor:** | Student |
| **Type:** | Primary |
| **Description:** | Student can watch their course videos online. |

### **2.5.12 Rate Courses**

Table 12

|  |  |
| --- | --- |
| **Use case Name:** | Rate Courses |
| **Actor:** | Student |
| **Type:** | Primary |
| **Description:** | Student can rate (1-5) courses and can add comment. |

## 2.6 Use Cases in Extended Format:

The expanded format of use cases describes the sequence of messages from actors to the system and the system’s response for each message during the execution of a use case. This format also shows the structure and content of the message passed between actors and the system.

The following are the extended formats of use cases:

### **2.6.1 Registration**

Table 13

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Register | |
| **Actor** | Student, Teacher, Admin | |
| **Purpose** | To make a profile to enroll, upload and mange courses. | |
| **Overview** | Student and Teacher must have register to enroll and upload lectures and admin has access to manage the data. | |
| **Type** | Primary | |
| **Pre-condition** | User must have email. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. The user enters registration data.   3) The admin receives new user notification. | | 2) The System display the request form  4) The System creates new profile if credentials are valid otherwise error message. |
| **Post-Condition** | User is registered successfully. | |

### **2.6.2 Login**

Table 14

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Login | |
| **Actor** | Student, Teacher, Admin | |
| **Purpose** | To visit a profile to enroll, upload and mange courses. | |
| **Overview** | Student and Teacher must have login to enroll and upload lectures and admin has to login to manage the data. | |
| **Type** | Primary | |
| **Pre-condition** | User must have registered. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. The user enters login data.   3) The admin receives user notification. | | 2) The System proceed the request  4) The System opens user profile if credentials are valid otherwise error message. |
| **Post-Condition** | User login successfully. | |

### **2.6.3 Edit Data**

Table 15

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Edit Data | |
| **Actor** | Admin | |
| **Purpose** | To add, delete, update and manage data of users. | |
| **Overview** | Admin can control and manage data. | |
| **Type** | Primary | |
| **Pre-condition** | Must have admin access. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. The user enters data.   3) The admin manages the data. | | 2) The System proceed the data.  4) The System save new data. |
| **Post-Condition** | Data edited successfully. | |

### **2.6.4 System Maintenance**

Table 16

|  |  |  |
| --- | --- | --- |
| **User Case Name** | System Maintenance | |
| **Actor** | Admin | |
| **Purpose** | To manage all resources in the system. | |
| **Overview** | Admin can control the system. | |
| **Type** | Primary | |
| **Pre-condition** | User must have Admin access. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. Admin manage all flow of data.   3) Admin handle all error in the system. | | 2) The System proceeds the data.  4) System save all the changes. |
| **Post-Condition** | System maintain successfully. | |

### **2.6.5 Upload Lectures**

Table 17

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Upload Lectures | |
| **Actor** | Teacher | |
| **Purpose** | To educate user/student and earn money. | |
| **Overview** | Teacher can upload course in which they have expertise. | |
| **Type** | Primary | |
| **Pre-condition** | User must login with teacher email. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. Teacher upload new videos and other study material. | | 2) The System save the new data.  3) The System notifies the admin. |
| **Post-Condition** | Teacher upload lectures successfully. | |

### **2.6.6 Edit Lectures**

Table 18

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Edit Lectures | |
| **Actor** | Teacher | |
| **Purpose** | To edit lectures. | |
| **Overview** | Teacher can edit lectures if he want to change something. | |
| **Type** | Primary | |
| **Pre-condition** | User must login with teacher email. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. Teacher wants to edit lectures. | | 2) The System shows option to edit lectures. |
| **Post-Condition** | Lecture edit successfully. | |

### **2.6.7 Search Course**

Table 19

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Search Course | |
| **Actor** | Student | |
| **Purpose** | To easily search courses. | |
| **Overview** | Student can search the course they want to study. | |
| **Type** | Primary | |
| **Pre-condition** | User must login with student email. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. Student type course in search engine. | | 2) The System finds the course from the database.  3) Then system shows the related courses to the student. |
| **Post-Condition** | Student search course successfully. | |

### **2.6.8 Enroll**

Table 20

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Enroll | |
| **Actor** | Student | |
| **Purpose** | To enroll student for course. | |
| **Overview** | Student can enroll whatever course they want to study. | |
| **Type** | Primary | |
| **Pre-condition** | User must login with student email. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. Student selects a course and wants to enroll the course. | | 2) The System shows options to enroll the course. |
| **Post-Condition** | Student enroll course successfully. | |

### **2.6.9 Payment**

Table 21

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Payment | |
| **Actor** | Student | |
| **Purpose** | To pay money. | |
| **Overview** | Student can use these the given payment method to buy courses. | |
| **Type** | Primary | |
| **Pre-condition** | User must login with student email. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. Student select the payment methods. | | 2) The System shows options for payment. |
| **Post-Condition** | Payment done successfully. | |

### **2.6.10 Watch video lectures**

Table 22

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Watch video lectures | |
| **Actor** | Student | |
| **Purpose** | To show video lectures to the students. | |
| **Overview** | Student can watch the video lectures of their courses they enroll. | |
| **Type** | Primary | |
| **Pre-condition** | User must login with student email and enroll the course. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. Student wants to watch their lectures. | | 2) The System shows video lectures of the course they enroll. |
| **Post-Condition** | Student watch lectures successfully. | |

### **2.6.11 Save Lectures**

Table 23

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Save Lectures | |
| **Actor** | Student | |
| **Purpose** | To save lectures to watch later. | |
| **Overview** | Student can save their lectures to watch whenever they are free. | |
| **Type** | Primary | |
| **Pre-condition** | User must login with student email and they enroll the course. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. Student wants to save lectures. | | 2) The System saves the lectures for the user. |
| **Post-Condition** | Lectures save successfully. | |

### **2.6.12 Rate Course**

Table 24

|  |  |  |
| --- | --- | --- |
| **User Case Name** | Rate Course | |
| **Actor** | Student | |
| **Purpose** | For course ratings. | |
| **Overview** | Student can Rate course from 1-5 stars and also add comment. | |
| **Type** | Primary | |
| **Pre-condition** | User must login with student email and enroll that course. | |
| **Typical Course Of Events** | | |
| **Actor Actions** | | **System Response** |
| 1. Student rate stars to the course and comment. | | 2) The System saves the ratings and comments. |
| **Post-Condition** | Student rate course successfully. | |

## 2.7 System Sequence Diagrams:

The system sequence diagram graphically shows the sequence of messages from actor to the system. It also depicts the contents of messages within the system and to the external systems also.

Following are the system sequence diagrams of this system.

### **2.7.1 Student Login**

### 

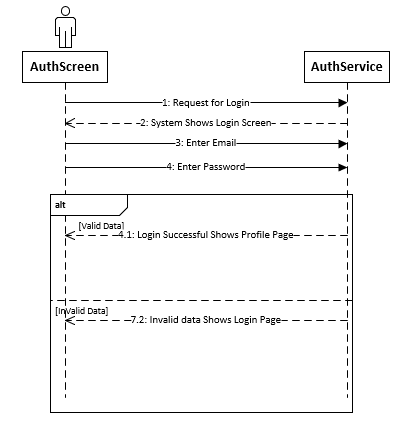


Figure 2

### **2.7.2 Student Registration:**

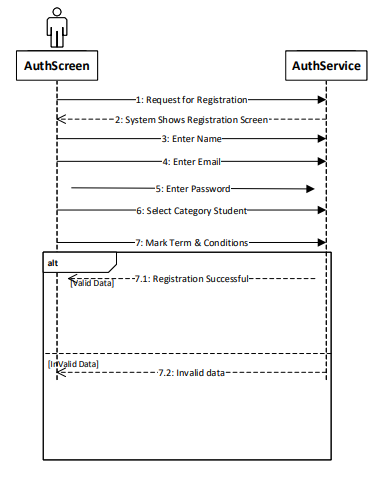


Figure 3

### **2.7.3 Student Course Enrollment:**

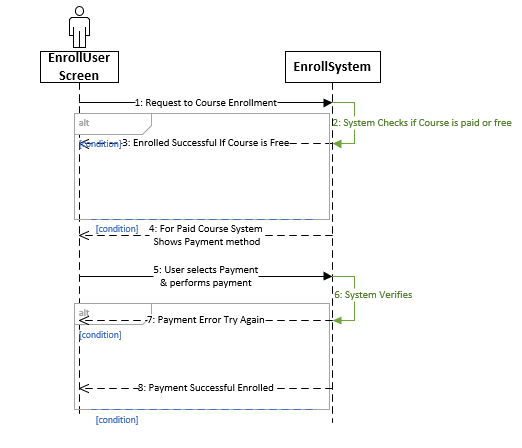


Figure 4

### **2.7.4 Teacher Upload Permission:**

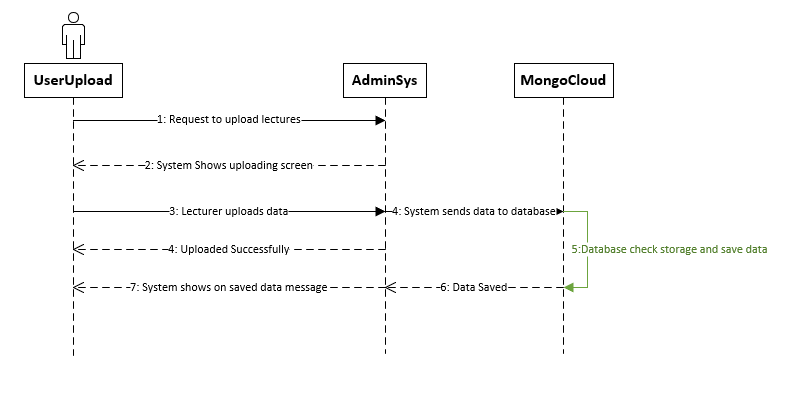


Figure 5

### **2.7.6 System Sequence Diagram:**

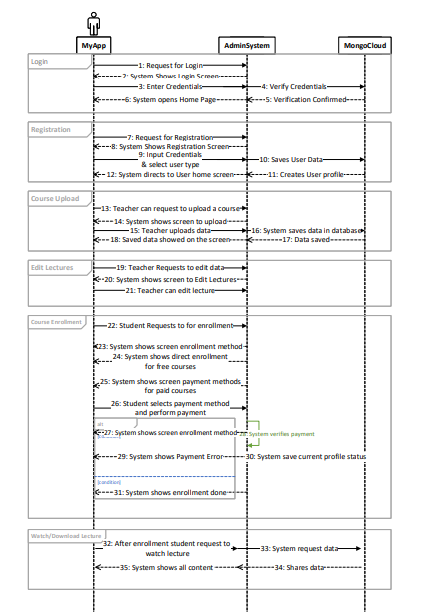


Figure 6

# Chapter 3: System Design

System design is the process of defining the components, modules, and data for a system to satisfy specified requirements. System development is the process of creating a systems, along with the processes and methodologies used to develop the product/system. System design includes following diagrams.

1. Class Diagram
2. ERD
3. Domain Model

## 3.1 Class Diagram:

A class diagram of a system shows the domain classes of a system, the attributes of classes, their operations, relationships between the classes and the interaction between these classes. A class diagram is helpful to start development of the system.

Below is the class diagram of our system that we created after detailed analysis and design in the previous sections:

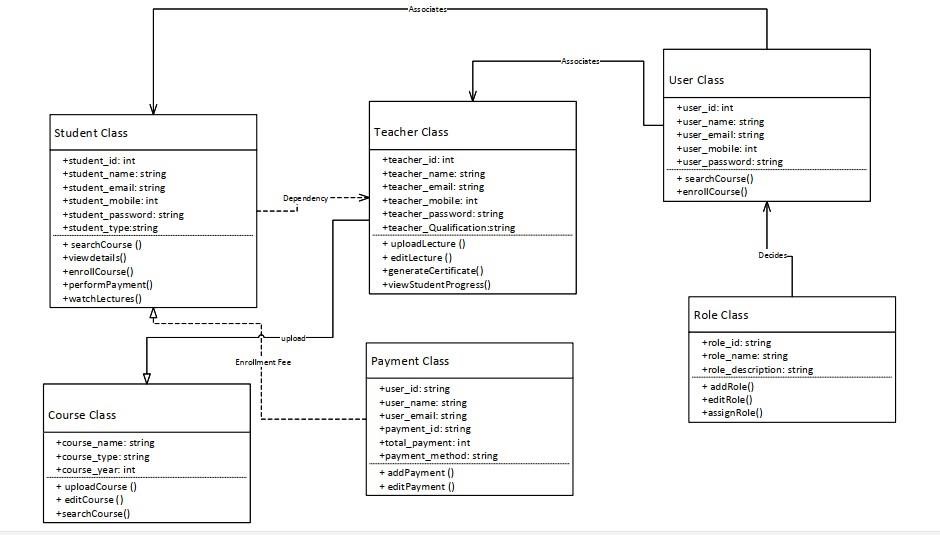


Figure 7

## 3.2 Domain Model:

Domain model is the graphical representation of conceptual classes that we create after analyzing the requirements and use cases of the system. It contains the domain classes, relationship and association between these domain classes.

.

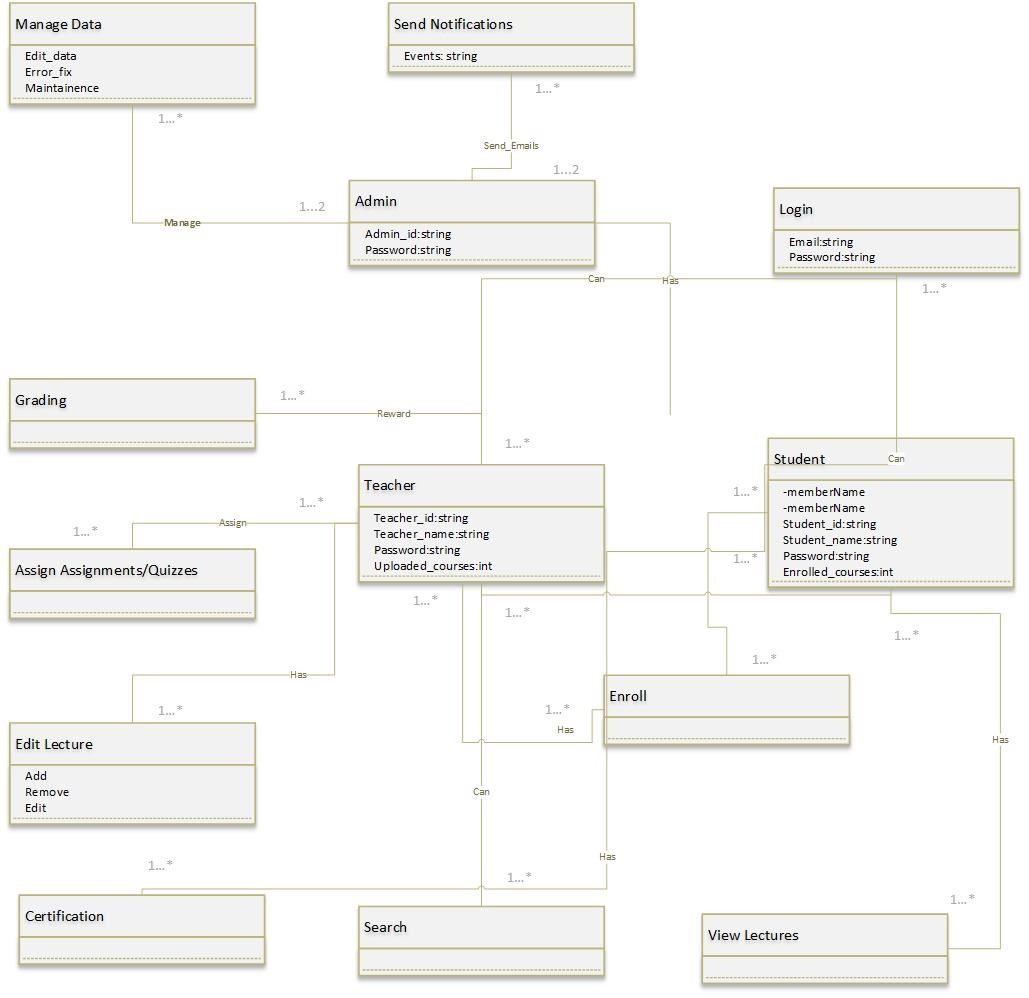


Figure 8

## 3.3 Entity Relationship Diagram (EERD):

The entity relationship diagram shows the entities and their relationship between each other in the normalized form.

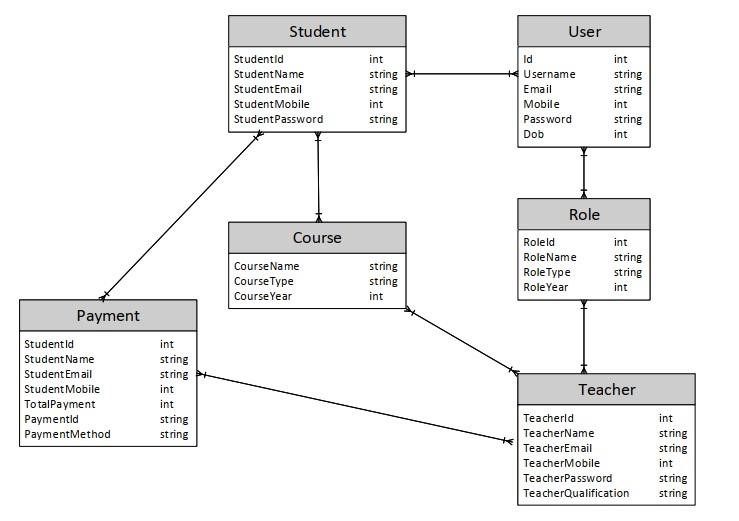


Figure 9

# Chapter 4: Implementation

Implementation is where we implement all the stuff we did in previous system analysis and system design. In this phase we have all the requirements and modules to work on in a standardized way. We work on latest try to work on latest tools and technologies to keep our systems updated and more secure than ever. In software engineering implementation is a creation of a product or a system through programming techniques, latest tools and technology approaches etc.

## 4.1 Detail Implementation Tools and technologies:

During implementation, we create the actual product that the user needed. It is an exciting phase because our idea of the product becomes reality and tangible. In this phase we begin developing and coding our project. As a team of two people we have to collect the specific requirements for our project. Like, what technology is best for the project or which is better for coding and programming.

**4.1.1 Technologies**

* **Flutter (Dart)**

We are using Flutter framework for the Front-end by using Dart language. We are using flutter because its new technology for cross platform development like apps, webs, desktop application etc. all in one code. Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase. First described in 2015, Flutter was released in May 2017. Dart is a programming language designed for client-side development such as web and mobile apps. Developed by Google, it can also be used to build server and desktop applications. It is an object-oriented, class-based, garbage-collecting language with a C-style syntax

* **Node JS, Express**

For the Back-end we are using the Node JS that serve the connection between the UI and the database the MongoDB. Node.js is an open source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside of a web browser designed to build scalable network applications.

* **MongoDB**

MongoDB is an open-source, cross-platform, document-centric database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and is licensed under the Server-Side Public License, which is considered non-free by many distributions. And also we another Back-end technology called MongoDB. MongoDB is a platform developed by Google for building mobile and web applications.

**4.1.2 Tools**

* **Android Studio**

We are working on the Android Studio and Visual Code for the coding. Android Studio is Google's official integrated development environment for the Android operating system, based on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

* **Visual Studio Code**

Visual Studio Code, also commonly known as VSCode, is Microsoft's source code editor for Windows, Linux, and macOS. Features include debugging support, syntax highlighting, smart code completion, code snippets, code refactoring, and built-in Git.

So, these are all details of Implementation and Tools and technologies we are using for the creating the product.

## 4.2 Component Diagram:

A component diagram, also called a UML component diagram, describes the organization and wiring of the physical components of a system. Component diagrams are often developed to model implementation details and to verify that every aspect of the system's intended functionality is covered by the intended development.

**Component Diagram**

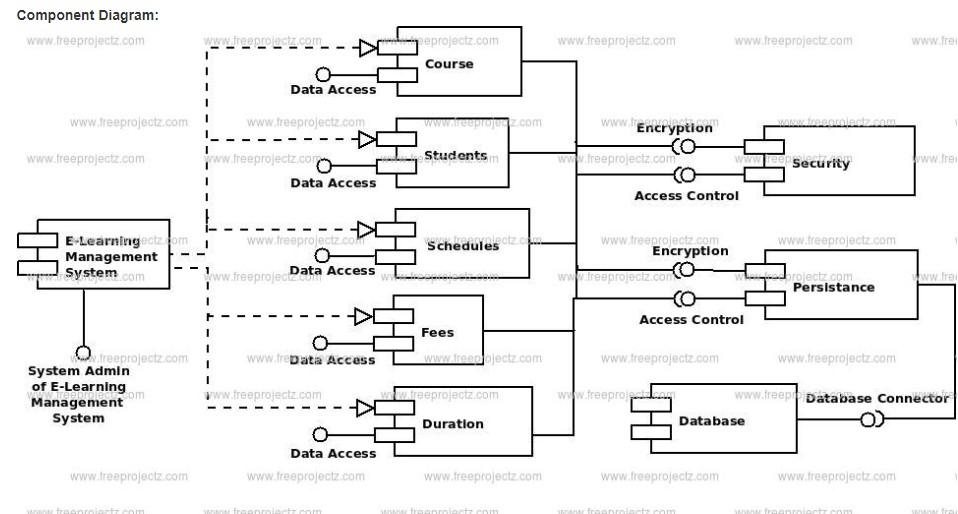
****

Figure 10

## 4.3 Deployment Diagram:

Implementation diagram is a type of diagram that specifies the physical hardware on which the software system runs. It also determines how the software is implemented on the underlying hardware. It assigns software from a system to the device it is running on.The implementation diagram maps the software architecture created in the design to the architecture of the physical system on which it runs. In distributed systems, it models the distribution of software on physical nodes.

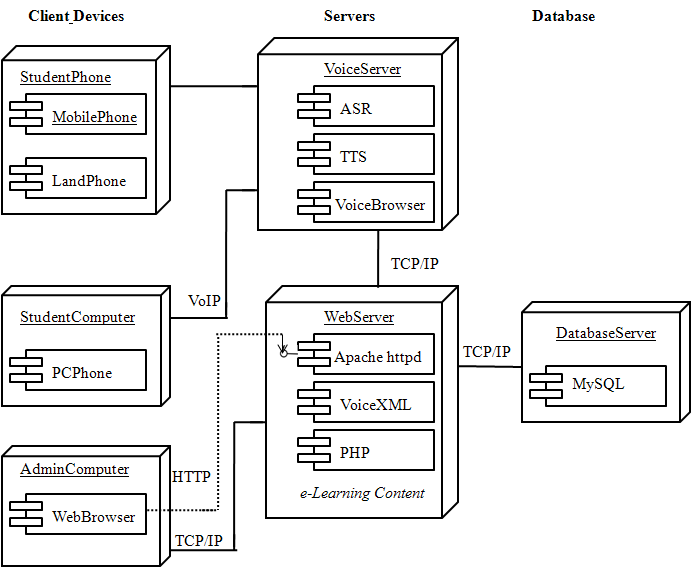
****

Figure 11

# Chapter 5: System Testing

Testing is a way to check either the system output is the required one or not or is it contain bugs and errors in it. For testing there are various techniques used to check standardized output those techniques are mentioned and briefly explained below. Software testing is process of examining the behavior of the software under test by validation and verification. Software testing can also provide a way towards quality assurance and quality control.

Quality assurance is the maintenance of quality in a product, especially by looking after and doing testing on each phase/component.

Quality control is a system of maintaining standards in developed products by testing a standard output against the specific input.

## 5.1 Methodology:

With this type of test framework, the designer can test his code using a model and a sketch after each individual part of the specified problem model. This test required each model to be tested in multiple environments, and the test itself consisted of multiple factors. This was referring to the tests that were performed throughout the lifecycle of the framework. In the meantime, all the elements of the structure were functioning. Starts with a base breach points that is needed to be finished.

Make a list of elements with the basic definition, a diagram showing the basic tables of flow information, a sectional plan and a diagram of each part, with a list of prerequisites and other tables with unused data by section. Pass the data to the parser and work with it as you write the code to verify that the moves in the report can be executed and that the expected results of each attack scenario can be achieved.

If the analyzer discovers a difficult step to perform, or the subject's experts feel that there is a lack of additional knowledge to gauge its strength, orchestrate 2 and, in most cases, proceed to orchestration 3.

## 5.2 White Box Testing:

White box testing is a testing technique where all aspects or internal causes of the tidal framework of the building system are tested. White box testing is a software testing methodology that tests the internal structures or operation of an application as opposed to its functionality. White box testing uses an inside perspective of the system along with programming skills to design test cases.

## 5.3 Black Box Testing:

Black box testing is a testing technique where all the internal complexities of the code and framework design remain undetected and the focus is on the output based on the input. Black box testing is a software testing method that examines the functionality of an application without examining its internal structures or operation. This testing method can be applied to virtually all levels of software testing: unit, integration, system, and acceptance testing.

## 5.4 Verification:

This process is done to ensure that the framework works as we need it to or as we expect it to work when it is released.

## 5.5 Validation:

The process to ensure that we have implemented the system according to user requirements and that the results are what the end user wants?

## 5.6 Test Cases:

### **Table 5.6.1: TC 01 to test the Signup/Registration page.**

|  |  |
| --- | --- |
| **TC ID** | 01 |
| **Strategy** | White Box |
| **Engineer** | Ameer Shahroz Khan |
| **Test Name** | Do Register |
| **Main Objective** | The objective is to test registration system |
| **Pre-Condition** | Internet connection must be established |
| **Steps to perform** | Enter the email number and click verify |
| **Expected Result** | Registration Successful |
| **Test Result** | Pass |

### **Table 5.6.2: TC 02 to test the Sign in/login page.**

|  |  |
| --- | --- |
| **TC ID** | 02 |
| **Strategy** | White box |
| **Engineer** | Ameer Shahroz Khan |
| **Test Name** | Do Login/sign in |
| **Main Objective** | Enter the email or username and password |
| **Pre-Condition** | Account must be registered |
| **Steps to perform** | Enter Email id and password |
| **Expected Result** | Successfully Entered the home screen |
| **Test Result** | Pass |

### **Table 5.6.3: TC 03 to test the Home/Main page.**

|  |  |
| --- | --- |
| **TC ID** | 03 |
| **Strategy** | Black Box |
| **Engineer** | Umair Khalid |
| **Test Name** | Check working of main screen |
| **Main Objective** | Search and every component of navigation bar are working properly |
| **Pre-Condition** | Must be login with Email id |
| **Steps to perform** | Need to write something in Search engine and need to check functionalities of nav bar |
| **Expected Result** | Search engine and other nav bar functionalities working successfully. |
| **Test Result** | Pass |

### **Table 5.6.4: TC 04 to test the Teacher uploading lectures.**

|  |  |
| --- | --- |
| **TC ID** | 04 |
| **Strategy** | White Box |
| **Engineer** | Umair Khalid |
| **Test Name** | Checking lecture uploads |
| **Main Objective** | Teacher upload lectures easily |
| **Pre-Condition** | Must Login with teacher Email id. |
| **Steps to perform** | Need to upload videos of to check. |
| **Expected Result** | Videos upload successfully |
| **Test Result** | Pass |

### **Table 5.6.5: TC 05 to test the Student download and watch lectures.**

|  |  |
| --- | --- |
| **TC ID** | 05 |
| **Strategy** | Black Box |
| **Engineer** | Ameer Shahroz Khan |
| **Test Name** | Watching and Downloading lectures |
| **Main Objective** | Student easily watch and download their lectures |
| **Pre-Condition** | Must login with Student Email id and have enrolled the course |
| **Steps to perform** | Play the video lectures and try to download. |
| **Expected Result** | It’s successfully working. |
| **Test Result** | Pass |

### **Table 5.6.6: TC 06 to test the logout/sign out session.**

|  |  |
| --- | --- |
| **TC ID** | 06 |
| **Strategy** | White Box |
| **Engineer** | Umair Khalid |
| **Test Name** | Do logout/sign out |
| **Main Objective** | Destroy the session. |
| **Pre-Condition** | Must be login/sign out. |
| **Steps to perform** | Click logout/sign out button |
| **Expected Result** | Successfully logout |
| **Test Result** | Pass |

# Chapter 6: Conclusion

Our Final year project is about the replica of udemy the e-learning app has been completed. It most widely use nowadays.

E-learning and e-learning are viable forms of education for populations with easy access to ICT, regardless of the remoteness of their physical location. Most potential countries are already ready to make full use of these new tools. However, in low- to medium-capacity countries, a number of political and socio-economic factors stand in the way of realizing the full potential of e-learning. To increase the effectiveness of online education, public and private institutions in each of these countries have met to improve infrastructure, revise education policies, and increase funding, and train users and operators. Work should be done to ensure quality and access. Indeed, more research is needed to clarify the relationship between ICT access and education and pave the way for such improvements. Distance learning is nothing new in Pakistan, the region has long valued synchronous and asynchronous learning technologies, but access to ICT is now increasing, even in countries least able to develop the online learning.

Our project has allowed us to learn and experience many new things, such as working with Flutter and Dart for front end and nodeJS as a backend to create android app development. These technologies are widely used in software engineering and computing fields.

## 6.1 Goals Achieved

**T**he main goals of our project are followings;

* Access course from your mobile by enroll.
* Discover and search for relevant content
* Student can take video, text lectures online and also download the lectures.
* Student can study in their own comfort zone.
* Student give quizzes and practical exams from their mobile phones.

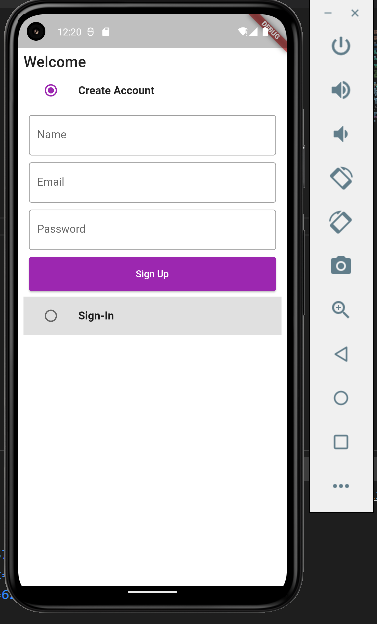
## 6.2 Future Goals

The following are the future goals;

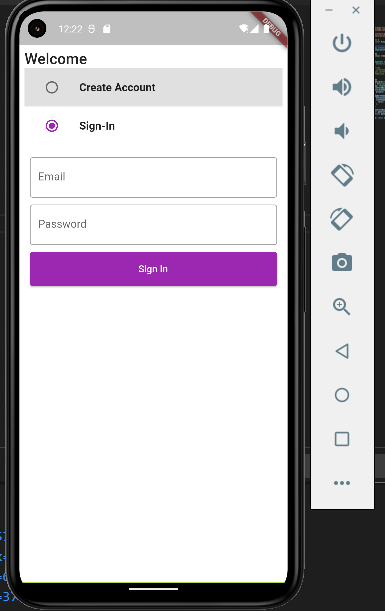
* 24/7 service
* Latest courses according to the time change
* Low price courses/Discount
* Scholarships
* Quizzes and assignments
* Certification
* Student and teacher feedback

# USER MANUAL

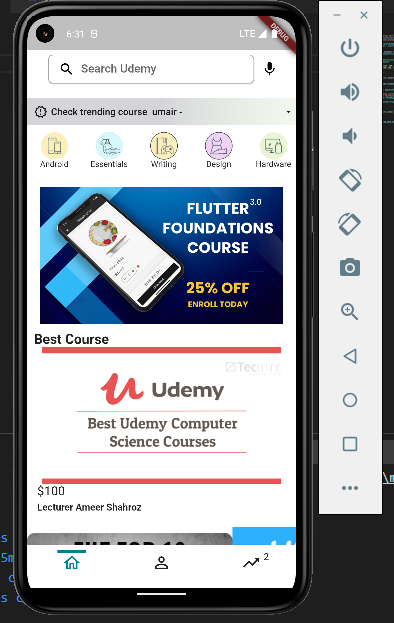
### **1 Sign up Page**

****

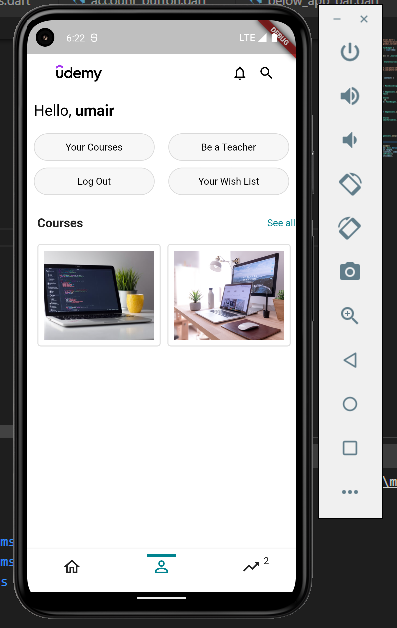
### **2 Sign in Page**

****

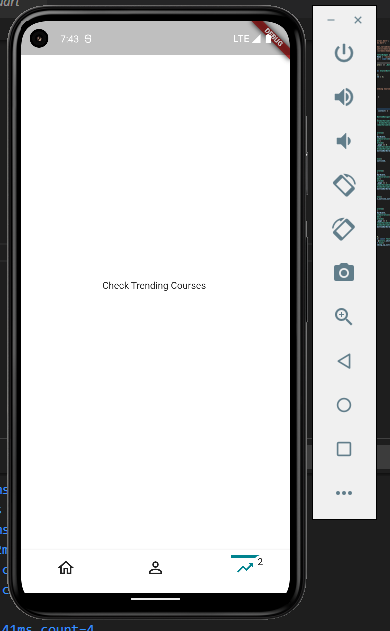
### **3 Home Page**

****

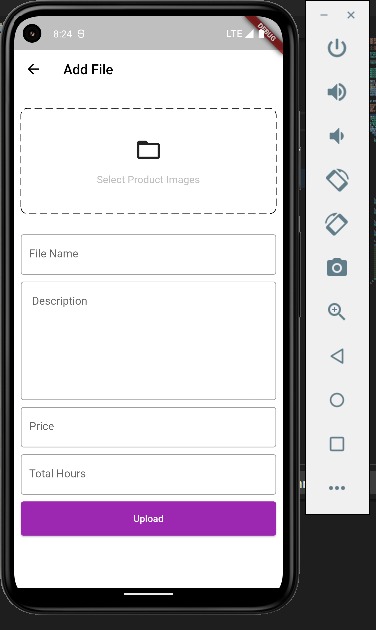
### **4 User Account**

****

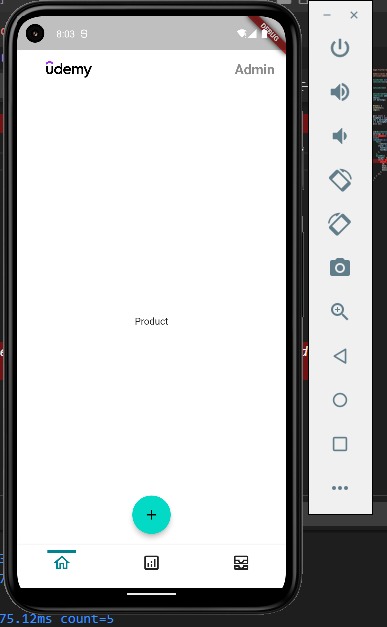
### **5 Trending Courses**

****

### **6 Course Upload**



### **7 Admin page**



# REFERENCES

1. <https://www.udemy.com>
2. <https://code.visualstudio.com/>
3. https://stackoverflow.com/
4. <https://en.wikipedia.org/wiki/Express.js>
5. https://expressjs.com/
6. <https://www.mongodb.com/products/compass>
7. <https://www.postman.com/>
8. <https://support.microsoft.com/office-2016-or-office-2010-7c695b06-6d1a-4917-809c-98ce43f86479>
9. <https://www.mongodb.com/basics/mongodb-atlas-tutorial#:~:text=MongoDB%20Atlas%20is%20a%20fully,scale%20MongoDB%20in%20the%20cloud>
10. https://cloudinary.com/console/c-1d9ccf2fff8d568cdeb9eb48c9ae7d/getting-started