Final Year Project

Smart Course Recommender System

Submitted By:

Syed Asad Ali Bukhari (2018-Isc-60)

Shuja Ur Rehman (2018-Isc-69)

Supervised By:

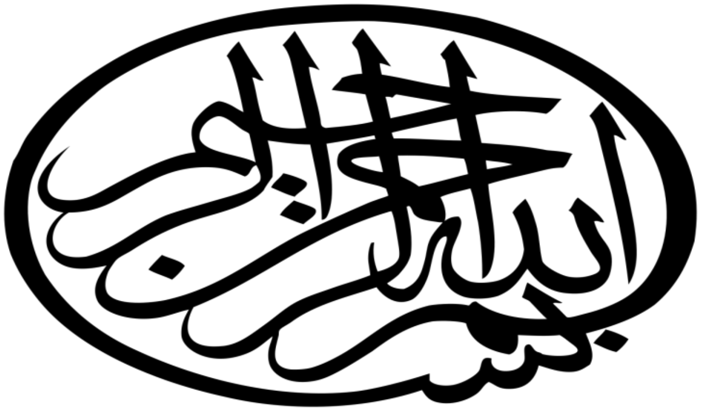
Sir Hafiz Asif

Session:

2018-2022

Bachelor of Science in Computer Science (BSCS)

Affiliated College
School of Advance Business and Commerce, Lahore
Punjab University College of Information Technology



IN THE NAME OF

ALLAH

THE MOST MERCIFUL

THE MOST GRACIOUS

THE MOST BENEFICIENT

"That man has nothing but what he strives for, and that his striving will be seen, and that afterwards he will be repaid for it with the fullest repayment, and that to thy Lord is the goal."

(Al - QURAN)

Dedication

This dissertation is dedicated to our parents for their relentless support throughout this time of final year project. Their unconditional care and prayers for us helped complete this documentation. Secondly, we dedicate this project to students who have a hard time taking a decision about their future education.

Statement of Submission

This is certify that **Syed Asad Ali Bukhari** and **Shuja Ur Rehman**, Registration No. **2018-Isc-60** and **2018-Isc-69** respectively has successfully completed the final year project named: **Smart Course Recommender System** at the School of Advance Business and Commerce affiliated college of PUCIT Lahore, to fulfill the partial requirement of the degree of Bachelor's in Computer Science (BSCS).

Acknowledgment

We are grateful to **Allah Almighty**, who gave us the motivation and drive to make this project a reality. Without the time, effort and support from several people, this dissertation would not reach its completion.

All our love and respect to the **Holy Prophet Hazrat Mohammed (P.B.U.H)** who guided us in the light of Holy Quran so that we can distinguish between the righteous and the wrong.

We would like to thank our advisor **Sir Hafiz Asif** for the guidance and supervision throughout our project. I would also like to say thanks to my dear friends **Mohsin jutt**, **Usama Meer** and **Furqan Asad** who guided me a lot in making the documentation and supported me in many aspects of this project. This project would not be accomplished without your support.

We are thankful to all our friends and teachers and specially parents for supporting us during the course of this documentation. We could not have completed this dissertation without the people we mentioned earlier.

Abstract

Students in the starting of their degree are often confused as to what they should study as the university offers a wide array of courses to study from. Likewise, the difficulty of these courses ranges from beginner to expert level and some require lab sessions too. Thus, to make the decision process easy for the students, we created this smart course recommendation system, so that students can use its services and make an informed decision about what to study in the coming semester. Furthermore, the system shows online and on-campus courses that are being taught at PUNJAB and other universities. Thus, when a university is teaching a certain course, you can simultaneously learn its specialty in an online course; this will benefit the student in future employment. Computer science is the main field that is being targeted in this project.

Index terms – Course recommendation, recommendation, online courses, on-campus courses, computer science field, IT field.

Undertaking

We certify that project titled" Smart Course Recommender System" in our own work. The work has not been presented elsewhere for assessment.

Signature of Students

Syed Asad Ali Bukhari (2018-Isc-60)

Shuja Ur Rehman (2018-Isc-69)

Table of Contents

Definitions and Acronyms

Acronym	**Definition**
PU	University of the Punjab
SCRS	Smart Course Recommender System
FR	Functional Requirements
NFR	Non-functional Requirements
SD	Sequence Diagram
CD	Collaboration Diagram
UC	Use Case
TC	Test Case
RID	Requirement ID
TID	Test case ID
UCID	Use Case ID

List of Figures

List of Tables

1. INTRODUCTION

1.1 Motivations

Motivation behind building smart course recommender is to help university students and professional individuals to find subjects of their interest and also provide online sources from where they can easily study a particular course. As students rush to schedule classes for the next semester, they feel overwhelmed by the number of options the institute has to offer. Moreover, if a student wants to take up a certain course and there is no room for them, there is little time to change their opinion. If the courses are recommended to students, they would have a potential list of top courses that will appeal to their interests and majors so they are not left in a doubt.

1.2 Project Overview

Our main goal is to provide undergraduate students a platform where they can easily get help in finding different subjects in which they are interested. Moreover, this project also recommends limited number of courses to students to study per semester based on level of difficulty of the courses or student's aptitude level by analyzing his previous CGPA. According to research done by group members, website provides their students this type of recommender system to help them figure out what they should study and how many courses to study in a semester and the difficulty level of the subject.

1.2.1 Overview Statement

In order to come to a solution for the issue of course selection, it is important to consider what data is available that could potentially be useful, and how to interpret and interact with this data. Clearly, there would have been a large number of students that have registered and succeeded in the same fields in the past. By looking at the combinations of courses some of these students took, and their performance in these courses, it is possible to determine certain course combinations that worked particularly well, and which course combinations were likely to result in failure.

1.2.2 Customer

As the system is free and publically available, we have no customers; rather we shall call the audience as students.

1.2.3 Goals

With the recent rise in employers embracing skills-based hiring practices, it has become integral for students to take courses that improve their marketability and support their long-term career success.

1.2.4 System attributes

In our work, we have adopted this assessment model of GAs and designed within it a recommender system that would use the data supplied by the students and would recommend courses to students based on their self-assessment.

1.3 Problem Statement

The problem can be summarized in two ways; First, the happiness of students with their courses, and Secondly, the likelihood of being successful in their studies. Clearly, these two factors have implications not only for students, but for the university they attend as well. A student who is unhappy with the material being taught in a course may choose to de-register it. This may result in the student having to wait until the next semester in order to register for another course, which results in time being wasted. From a university's perspective, any self-funding students who choose to leave the university would result in a loss in fees that would have been paid not only in that particular year but in future years as well. Furthermore, negative feedback of discontented students may cause potential students to decide not to enroll in a course or at a certain university.

1.4 Objectives

To recommend courses, we decided to apply user-based collaborative filtering which is a technique to predict the rating of an item based on the ratings of most similar other users who rated other items. In terms of courses and students, we decided to use a weighted average of common courses and grades among those courses as the rating of a student. We then use the Pearson correlation coefficient to calculate the correlation among each current student and all the previous students using the rating of common courses taken. Since there are a great amount of previous students and current students, this becomes difficult to process with traditional data processing applications. This system allows students to plan their courses for an individual term, multiple terms, or for their entire duration at the university. This would allow students to track their progress towards meeting the requirements associated with their career objectives, manage personalized academic plans where requirements are directly linked to course Sign up, and eliminate the need to access separate systems to track a student's history, grades, academic plans and other information.

2. Domain Analysis

2.1 Customer

As this is a free recommendation service system, we don't have customers, we shall call the targeted audience users i.e. students.

2.2 Stakeholders

The following table lists the stakeholders along with their roles in the working of our system.

Stakeholder	**Role in System**
Student (Actor)	The student plays a significant role in using our system.
Admin (Actor)	The admin is responsible for overlooking the whole management of the System.
Guest User (Actor)	This individual is simply browsing the website.

Table 1: Stakeholders

2.3 Affected Groups with social or economic impact

- **Students**
  This project's main focus is to suggest courses according to their interests. So they are able to achieve excellent grades and learn different skills.
- **Professional Individuals**
  People who feel they need to sharpen their skills can use this system to find online sources and whether they are paid or unpaid.
- **Universities**
  Universities that offer CS short courses and sessions can benefit from this system as they can advertise and attract people who are interested in these types of courses.

2.4 Dependencies/ External Systems

This system would not be complete unless we implemented category specific filters to make the suggestion for the students using our website. We also used Rest API to make our project function accurately.

2.5 Reference Documents

[1] Resnick P. and Varian H., Recommender Systems, Communication of the ACM, 40(3), 56-58, March 1997.
[2] Montaner M., Lopez B., and Josep R., A Taxonomy of Recommender Agents on the Internet, Artificial Intelligence Review, 19, 285-330, 2003.
[3] Bridge D., Product recommendation systems: A new direction. In R.Weber, Wangenheim, C., eds.: Procs. of the Workshop Programme at the Fourth International Conference on Case-Based Reasoning. Vancouver, Canada, 79-86, 2001.
[4] Burke R., Hybrid Recommender Systems: Survey and Experiments, User Modelling and User-Adapted. Interaction 12, 4, 331-370, 2002.

2.5.1 Related Projects

In order to develop Smart Course Recommender, we looked up similar systems. Their details are listed below.

2.5.2 Feature Comparison

The table below builds a comparison between **Pluralsight** and our system.

Overall Features	Pluralsight	Our System
Whole Procedure	Online	Online
Authenticity	50%	90%
Availability	24/7	24/7
Selection of courses on what basis	Field, Interest and difficulty level	Degree , semester, student interest, difficulty and credit hours.
Highlights whether you could study a certain course	No	No
Suggestion on where to study the course physically or online source	No	Yes

Table 2: Feature Comparison

3. Requirements analysis

3.1 Functional Requirements

The following table lists down the functional requirements of the project.

RID	Description	Category	Attribute	Details and Boundary Constraints
FR1.1	The user shall Sign Up into the system.	Functional Requirements	Sign up page	○ Enter Name ○ Enter Email ○ Enter Password ○ Confirm Password
FR1.2	The user shall login into the system.	Functional Requirements	Login page	○ Enter Login ID. ○ Enter Login Password. ○ Reset forgotten Password.
FR1.3	The user shall be able to select different options.	Functional Requirements	Main Menu (Front End)	○ Home ○ Courses recommendation ○ Online courses ○ On-campus courses
FR1.4	The user shall select the fields required by the system.	Functional Requirements	Course recommendation	○ Select Degree ○ Select Semester level
FR1.5	The user can enter more information to get custom recommendation.	Functional Requirements	Course recommendation	○ Select Category
FR1.6	The user can select Difficulty Level.	Functional Requirements	Course recommendation	○ Select Difficulty
FR1.7	The user can provide his CGPA for specific suggestions.	Functional Requirements	Course recommendation	○ CGPA
FR1.8	The user shall be able to add a course to the cart and update and delete from it as well.	Functional Requirements	Online courses/ On-campus courses	○ Add to Cart ○ Update Cart ○ Delete from Cart

FR1.9	The system will recommend online and on-campus courses.	Functional Requirements	Online courses/ On-campus courses	○ College/University ○ Online Sources
FR1.10	The user shall be able to view information about selected institute.	Functional Requirements	On-campus courses	○ Course Feedback ○ Location ○ Difficulty Level
FR1.11	The user shall be able to see online sources for selected course.	Functional Requirements	Online courses/ On-campus courses	○ Paid/Unpaid
FR1.12	The user shall be able to see rating of a certain course.	Functional Requirements	Rating	○ View Rating
FR1.13	There will be a separate admin board.	Functional Requirements	Admin Board	○ He/She is responsible for site management.
FR1.14	The admin will be responsible for adding and updating information about different institutes.	Functional Requirements	Admin Board	○ Updating Institutes' Fees ○ Updating Location

Table 3: Functional Requirements

3.2 Non-Functional Requirements

The following table declares the non-functional requirements of the project.

RID	**Description**	**Category**	**Attribute**	**Details and Boundary Constraints**
NFR2.1	Performance of the system.	Non-Functional Requirements	Interface	○ This system should respond promptly.
NFR2.2	Usability of the system.	Non-Functional Requirements	Interface	○ This system should be easy to use and understand.
NFR2.3	Correctness of the system.	Non-Functional Requirements	Interface	○ This system will fulfill all the functional and non-functional requirements.
PR3.1	Web Development Tools.	Programming Requirements	Developing Site	○ Django framework ○ ReactJS ○ Xampp
DR4.1	Data store in MySQL.	Data Requirements	Storage Resource	○ The rational database includes the entity definition, attribute definition and physical database mapping to the entities used in the application. ○ MySQL Server will be used for this purpose.

Table 4: Non-functional Requirements

3.3 List of Actors

i. Administrator
ii. Students

3.4 List of Use Cases

1. Student Sign up
2. Student Login
3. View Trending Courses
4. View On-campus Courses
5. View Online Courses
6. Recommendation
7. View Difficulty and Rating
8. View Cart
9. Manage Students
10. Post News Feed
11. Add Courses
12. Update Courses

3.5 System Use Case Diagram

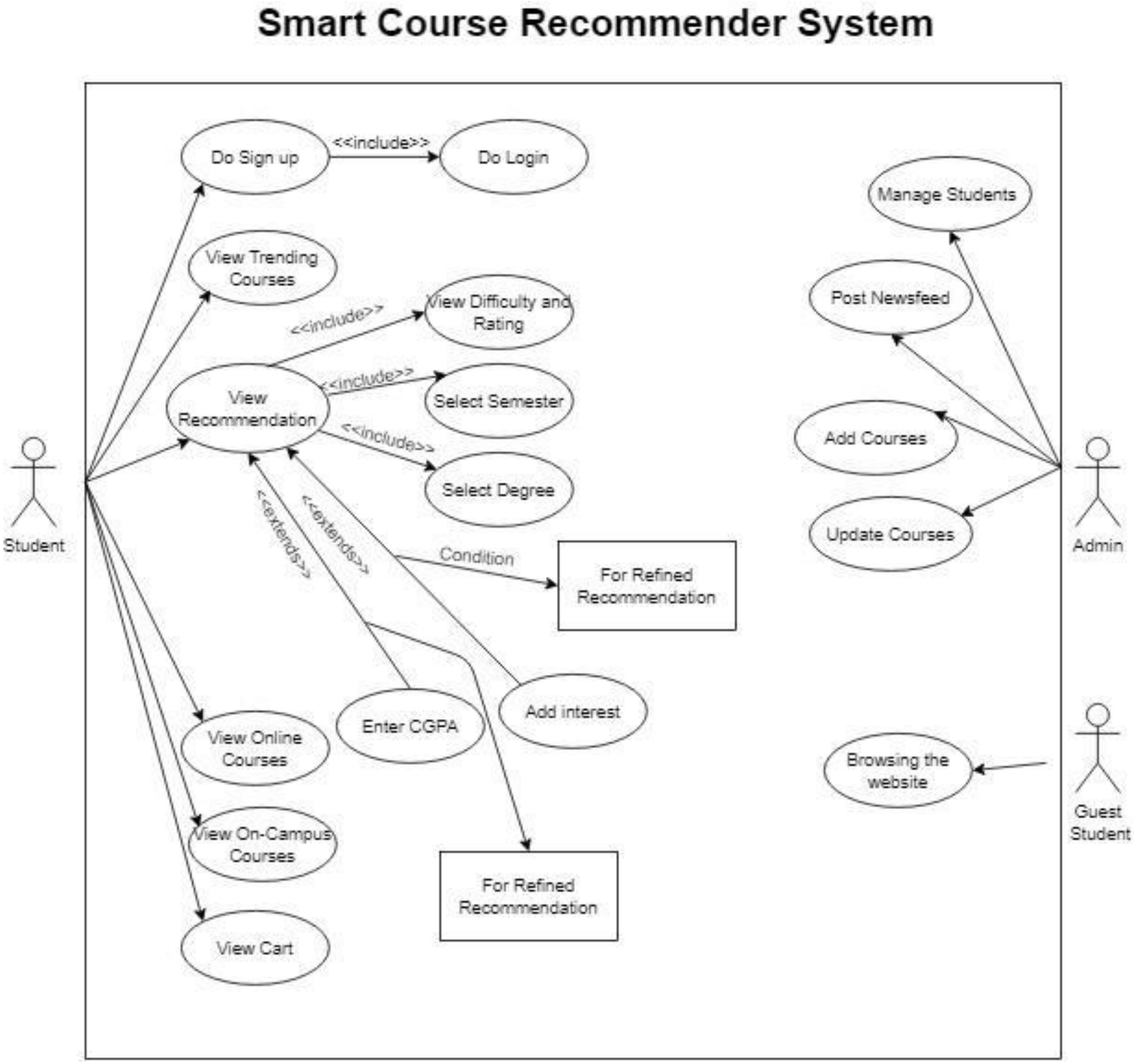


Figure 1: System Use Case Diagram

The figure above shows the working of the system and its users via the system use case diagram.

3.6 Extended Use Cases

Use case number 1 explains the student sign up function of the system.

UC-1	
Use Case Name: Student Sign up	
Created By: Shuja Ur Rehman	**Last Updated:** September 5, 2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Student	
Description: This is the first use case as it adds a new student into an existing system. It requires the student to enter basic information like Name, Email and Password. An account will be created from which the student will take further actions.	
Trigger: Click on “Sign up” on home page.	
Pre-condition: Student must have a valid email account.	
Post Condition: The student shall have an account through which they will take future action.	
Normal Flow: 1) Student enters Name. 2) Student enters Email. 3) Student enters Password. 4) Student clicks on signs up button. 5) Student has created an account.	
Alternate Flow: **4a)** In step 1 of the Normal Flow, student entered a name that is already in the system. ➢ System will give an error. ➢ Student acknowledges the error. ➢ Student enters a name that is not repeated in the system. ➢ System moves forward. **4b)** In step 4 of the Normal Flow, if the student has not entered complete fields, ➢ System gives an error. ➢ Student fills the all the required fields. ➢ System moves forward.	
Exceptions: **3a)** In step 3 of Normal Flow, the password is not according to guidelines. ➢ System will give error. ➢ Student acknowledges the error. ➢ Student enters a valid password. ➢ System moves forward.	
Frequency: On Demand	
Assumption: We assume that Student is fluent in English language.	

Table 5: UC-1: Student Sign up

Use case number 2 explains the student login function of the system.

UC-2	
Use Case Name: Student Login	
Created By: Shuja Ur Rehman	**Last Updated:** September 5, 2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Student	
Description: This use case is applicable only when a student is already registered in this system. The student enters an Email and Password and logs in the system.	
Trigger: Click on “Login” on Sign up page.	
Pre-condition: Student must have a valid account in the system.	
Post Condition: The student should be logged in this website’s account to avail its services.	
Normal Flow: 1) Student enters Email and Password. 2) System recognizes the student account and allows access. 3) Student logs in his account.	
Alternate Flow: **1a)** In step 1 of the Normal Flow, student can login by using their Google, Facebook, LinkedIn and Twitter. 1) Student clicks on “Facebook” button. 2) System confirms it is the same Student’s account logged in Facebook. 3) Student clicks confirm. 4) System logs in the website using Facebook account.	
Exceptions: **1a)** In step 1 of Normal Flow, student has entered an incorrect password. ➢ System detects a fault. ➢ Student enters correct password. ➢ System successfully logs in the system. **1b)** In step 1 of the Normal Flow, student has entered invalid email address. ➢ System shows the fault. ➢ Student enters valid and registered email address. ➢ System login is successful.	
Frequency: On demand.	
Assumption: We assume that student is familiar with English Language.	

Table 6: UC-2: Student Login

Use case number 3 explains the view trending course function of the system.

UC-3	
Use Case Name: View Trending Courses	
Created By: Shuja Ur Rehman	**Last Updated:** September 5,2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Student	
Description: This use case is for viewing the newsfeed as the student logs in his account. This page shows trending courses at real time.	
Trigger: Click on “Home”.	
Pre-condition: Student must have a valid account.	
Post Condition: The student should be able to see the courses on the website.	
Normal Flow: 1) Student enters Email and Password. 2) Student is logged in the account. 3) Student sees the trending courses.	
Exceptions: **1a)** In the step 1 of Normal Flow, student has entered an invalid email. ➢ System detects a fault. ➢ Student enters a valid email address. ➢ System continues. **1b)** In step 1 of the Normal Flow, student entered a name that is already in the system. ➢ System will give an error. ➢ Student acknowledges the error. ➢ Student enters a name that is not repeated in the system.	
Includes: **UC-1:** Student Sign up **UC-2:** Student Login	
Frequency: On demand.	
Assumption: We assume that student is familiar with English language.	

Table 7: UC-3: View Trending Courses

Use case number 4 explains the view on-campus course function of the system.

UC-4	
Use Case Name: View On-Campus Courses	
Created By: Shuja Ur Rehman	**Last Updated:** September 5, 2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Student	
Description: This use case defines how the student can view the On-campus courses.	
Trigger: Click on “On-Campus Courses”.	
Pre-condition: i. Student must have a valid email account. ii. Student must be logged in his account.	
Post Condition: The student can view the on-campus courses.	
Normal Flow: 1) Student enters Email and Password. 2) Student is logged in the account. 3) Student clicks on “On-Campus Courses”. 4) Student can see the list of courses.	
Alternative Flow: **1a)** In step 1 of the Normal Flow, student has entered improper password. ➢ System gives an error “Incorrect Password”. ➢ Student enters correct password. ➢ System continues forward.	
Exceptions: **3a)** In step 4 of the Normal Flow, student abruptly quits the website. ➢ System halts.	
Includes: **UC-1:** Student Sign up **UC-2:** Student Login	
Frequency: On demand.	
Assumption: We assume that the student belongs to Computer Science field.	

Table 8: UC-4: View On-Campus Courses

Use case number 5 explains the view online course function of the system.

UC-5	
Use Case Name: View Online Courses	
Created By: Shuja Ur Rehman	**Last Updated:** September 5, 2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Student	
Description: This use case comes in practice when the student wants to view online courses bring offered by the system.	
Trigger: Clicked on “Online Courses” on the home page.	
Pre-condition: Student must have logged into the system.	
Post Condition: The student should be able to see the online courses.	
Normal Flow: 1) Student enters Email and Password. 2) Student is logged in his account. 3) Student clicks “Online Courses”. 4) Student sees the online courses list.	
Alternative Flow: **2a)** In step 2 of the Normal Flow, student is not logged into the account. ➢ System gives an error. ➢ Student fixes the mistake. ➢ System proceeds forward.	
Exceptions: **1a)** In step 1 of the Normal Flow, Student has entered a faulty password. ➢ System gives an error “Incorrect Password”. ➢ Student enters correct password. ➢ System continues forward.	
Includes: **UC-1:** Student Sign up **UC-2:** Student Login	
Frequency: On demand.	
Assumption: We assume that the student belongs to Computer Science field.	

Table 9: UC-5: View Online Courses

Use case number 6 explains the view recommendation function of the system.

UC-6	
Use Case Name: View Recommendation	
Created By: Shuja Ur Rehman	**Last Updated:** September 5,2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Student	
Description: This is the main use case of the system. Student selects the category and difficulty level and the system shows the recommended courses.	
Trigger: Clicked on “Recommendation” on the homepage.	
Pre-condition: Student should have a valid account in the system.	
Post Condition: The student should be able to see the recommended courses.	
Normal Flow: 1) Student clicks “Recommendation”. 2) Student fills the required fields. 3) System processes the requirements. 4) Student gets the suggestions.	
Alternate Flow: **2a)** In step 2 of the Normal Flow, student has not filled a mandatory field. ➢ System gives an error “Select Degree”. ➢ Student fixes this mistake. ➢ System continues forward.	
Exceptions: **2a)** In step 2 of the Normal Flow, student left abruptly. ➢ System halts and giver error.	
Includes: **UC-1:** Student Sign up **UC-2:** Student Login	
Frequency: On demand.	
Assumption: We assume that the student belongs to Computer Science field.	

Table 10: UC-6: View Recommendation

Use case number 7 explains the view difficulty and rating function of the system.

UC-7	
Use Case Name: View Difficulty and Rating	
Created By: Shuja Ur Rehman	**Last Updated:** September 5, 2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Student	
Description: This is the additional use case in recommendation phase. Student can view the difficulty level of a course and can view the rating given by other students.	
Trigger: Clicked on “Recommendation” on home page. **Pre-condition:** Student	
should have gone through the recommendation process. **Post Condition:** The	
student should be able to see the difficulty level and rating. **Normal Flow:**	
1) Student enters email and password. 2) Student is logged in his account. 3) Student clicks “Recommendation”. 4) Student fills the required fields. 5) System processes the requirements. 6) Student gets the suggestions. 7) Student views the difficulty level. 8) Student should be able to view the rating of the courses. **Alternate Flow:**	
2a) In step 2 of the Normal Flow, Student has not filled out all fields. ➢ System will give an error. ➢ Student enters all the fields required by the system. ➢ System continues on effectively. **Exceptions:**	
1a) In step 1 of Normal Flow, student enters invalid password. ➢ System gives an error. ➢ User enters correct password. ➢ System continues forward. **Includes:**	
UC-1: Student Sign up **UC-2:** Student Login **UC-6:** View Recommendation. **Frequency:** On demand.	
Assumption: We assume that the student belongs to Computer Science field.	

Table 11: UC-7: View Difficulty and Rating

Use case number 8 explains the view cart function of the system.

UC-8	
Use Case Name: View Cart	
Created By: Shuja Ur Rehman	**Last Updated:** September 5, 2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Student	
Description: This use case enables when a student adds a course to the website’s cart.	
Trigger: Clicked on “Cart” on recommendation page.	
Pre-condition: Student should have added a course in the cart.	
Post Condition: The student will have given criticism regarding the system.	
Normal Flow: 1) Student enters Email and Password. 2) Student is logged in his account. 3) Student clicks “Recommendation”. 4) Student gives the required information for suggestion. 5) Student gets the recommendation. 6) Student adds the course to cart. 7) Student can update and delete from the cart.	
Alternate Flow: **4a)** In step 4 of the Normal Flow, Student has not filled all fields. ➢ System will give an error. ➢ Student fills required information. ➢ System moves forward.	
Exceptions: **1a)** In step 1 of the Normal Flow, Student has entered a faulty password. ➢ System gives an error “Incorrect Password”. ➢ Student enters correct password. ➢ System continues forward.	
Includes: **UC-1:** Student Sign up **UC-2:** Student Login **UC-6:** View Recommendation. **UC-7:** View Difficulty and Rating	
Frequency: On demand.	
Assumption: We assume that the student belongs to Computer Science field.	

Table 12: UC-8: View Cart

Use case number 9 explains the manage students function of the system.

UC-9	
Use Case Name: Manage Students	
Created By: Shuja Ur Rehman	**Last Updated:** September 5, 2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Administrator	
Description: This task is the accountability of the admin to see that the students are abiding by the rules and regulations of the website and there are no fake accounts in use.	
Trigger: This case is initiated when the admin is managing the audience using the site.	
Pre-condition: i. Administrator should have access to manage the Students. ii. Administrator should be able to differentiate between actual and fake situations.	
Post Condition: The website has authentic audience and feedback is real.	
Normal Flow: 1) Administrator opens the backend of the website. 2) Administrator manages the Students coming to the website.	
Frequency: 5-10 times per week	

Table 13: UC-9: Manage Students

Use case number 10 explains the posting newsfeed function of the system.

UC-10	
Use Case Name: Posting Newsfeed	
Created By: Shuja Ur Rehman	**Last Updated:** September 5, 2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Administrator	
Description: This task says that the administrator is responsible for posting news that is relevant to this website. Some example can be trending courses of the time.	
Trigger: This case is initiated when the administrator is posting newsfeed.	
Pre-condition: Administrator should have access to making changes.	
Post Condition: The administrator should be mentioning news related to courses.	
Normal Flow: 1) Administrator opens the backend of the website. 2) Administrator goes to the Home section. 3) Administrator adds news dedicated to real time trending courses.	
Exceptions: **1a)** In step 1 of the Normal Flow, administrator is not allowed to access the home section.	
Frequency: 2-5 times per week.	

Table 13: UC-10: Posting Newsfeed

Use case number 11 explains the add courses function of the system.

UC-11	
Use Case Name: Add Courses	
Created By: Shuja Ur Rehman	**Last Updated:** September 5, 2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Administrator	
Description: This use case says that the administrator is responsible for adding new courses as CS field is constantly changing. Moreover, They also have to update the course material (online sources).	
Trigger: This case is initiated when the administrator is adding new material.	
Pre-condition: i. Administrator should have access to making changes.	
Post Condition: The administrator has added the courses' material.	
Normal Flow: 1) Administrator opens the backend of the website. 2) Administrator adds the courses.	
Exceptions: **1a)** In step 1 of the Normal Flow, administrator is not allowed to access the backend.	
Frequency: 1-2 times per week.	

Table 15: UC-11: Add Courses

Use case number 12 explains the update courses function of the system

UC-12	
Use Case Name: Update Courses	
Created By: Shuja Ur Rehman	**Last Updated:** September 5, 2022
Date Created: September 4, 2022	**Last Revision Date:** December 8, 2022
Actor: Administrator	
Description: This use case says that the administrator is responsible for updating new courses as CS field is constantly changing.	
Trigger: This case is initiated when the administrator is updating new material.	
Pre-condition: ii. Administrator should have access to making changes.	
Post Condition: The administrator has updated the courses' material.	
Normal Flow: 3) Administrator opens the backend of the website. 4) Administrator updates the courses.	
Exceptions: **1a)** In step 1 of the Normal Flow, administrator is not allowed to access the backend.	
Frequency: 1-2 times per week.	

Table 16: UC-12: Update Courses

4. DATA FLOW DIAGRAM (OPTIONAL)

4.1 Data Flow Diagram Level 0

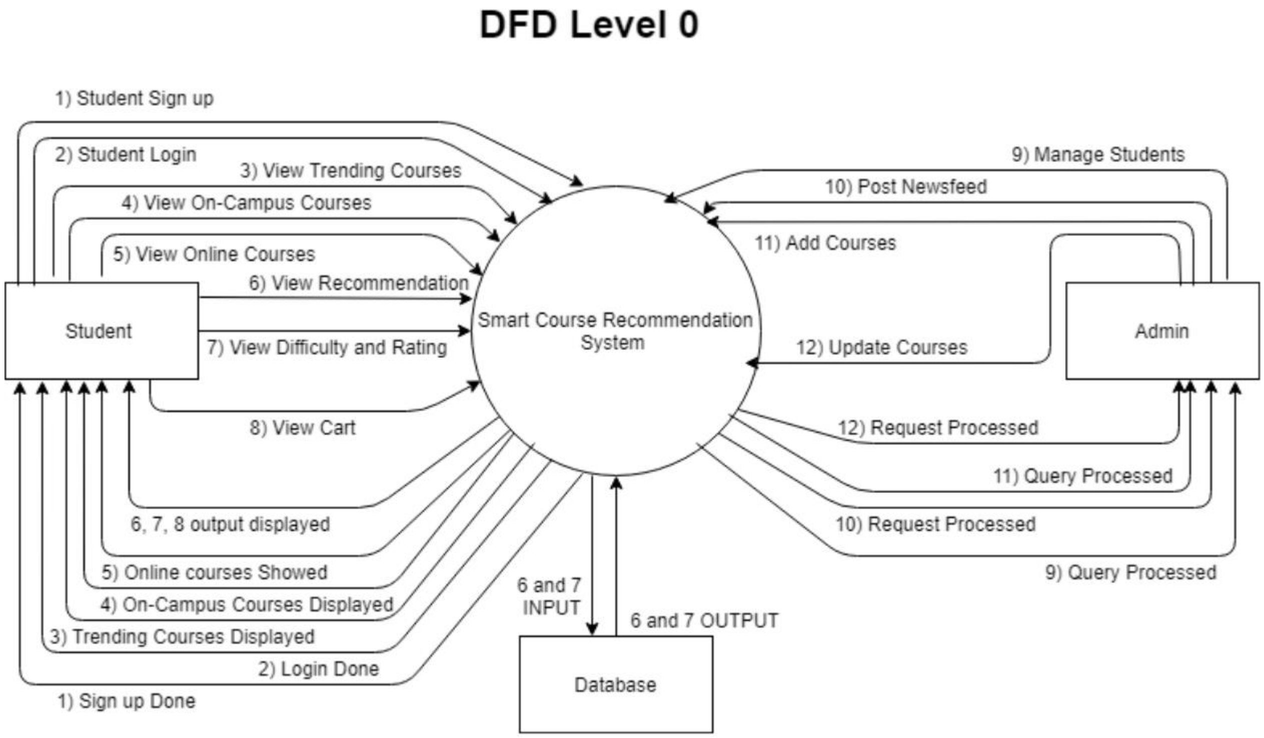


Figure 2: Data Flow Diagram Level 0

The above diagram shows the flow of data through the system referring it as DFD level 0.

4.2 Data Flow Diagram Level 1

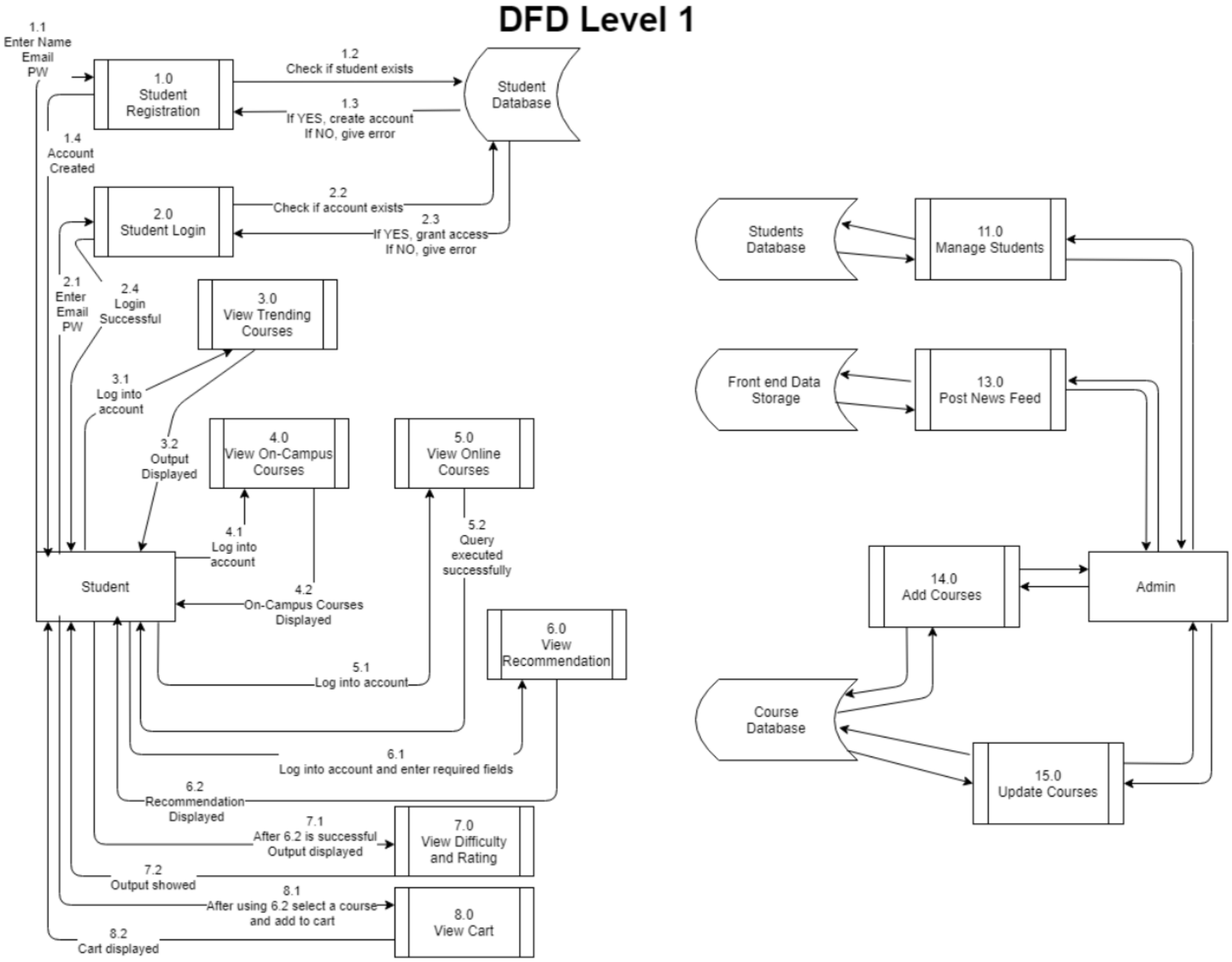


Figure 3: Data Flow Diagram Level 1

The above diagram shows the flow of data through the system referring it as DFD level 1.

5. System Design

5.1 Class Diagram

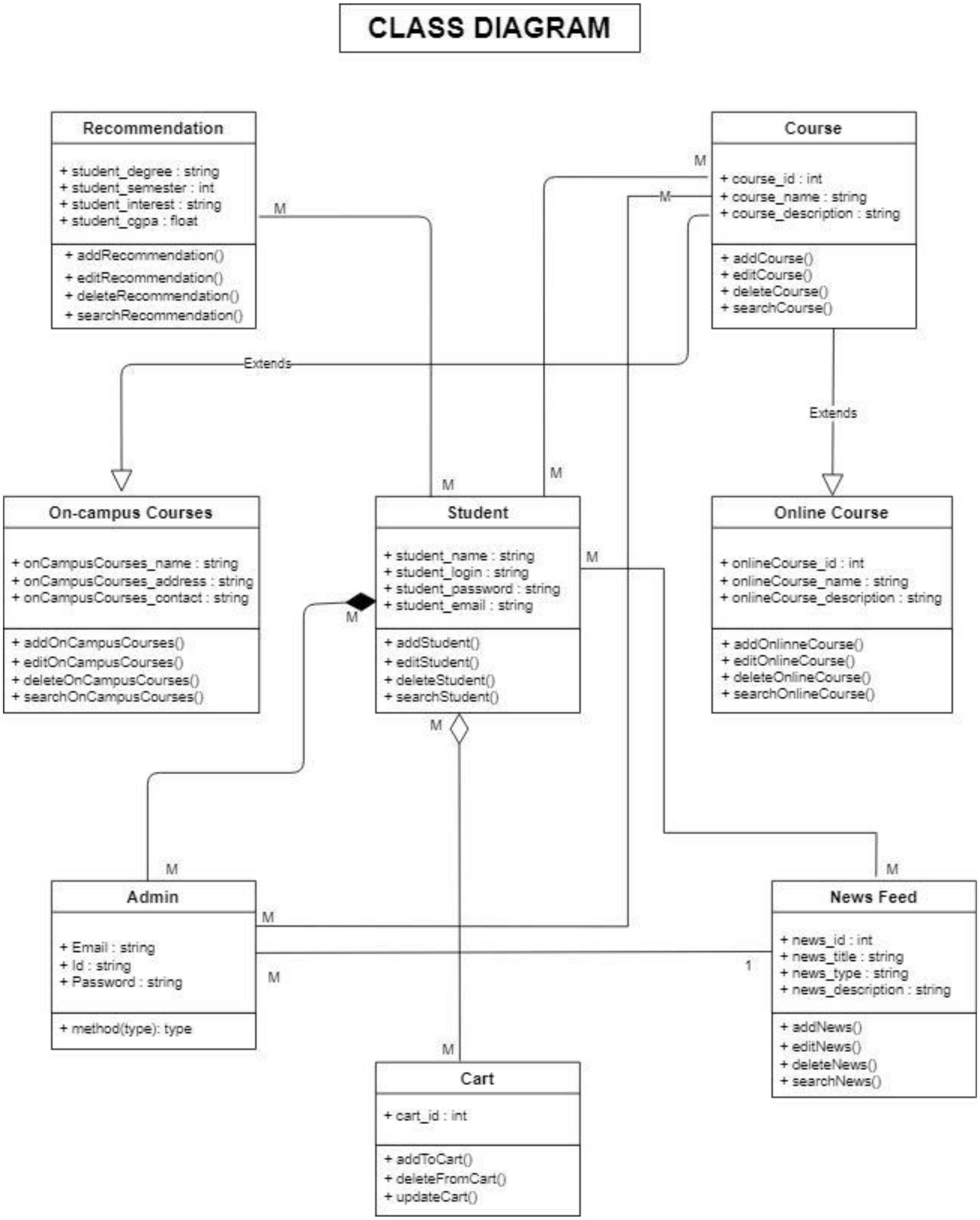


Figure 4: Class Diagram

The above diagram shows the classes, attributes and functions used in the system.

5.2 Sequence Diagrams

1. Student Sign up

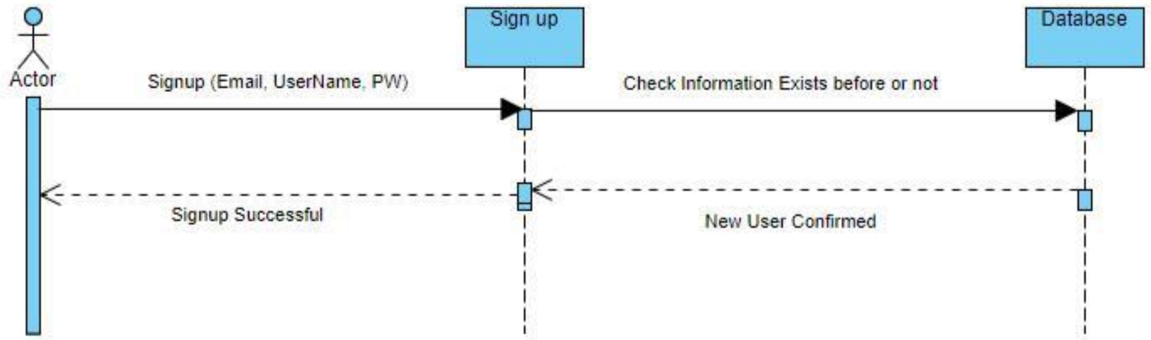

Actor
Sign up
Database
Signup (Email, UserName, PW)
Check Information Exists before or not
Signup Successful
New User Confirmed

Figure 5: SD: Student Sign up

The above figure is of student sign up in sequence diagram form.

2. Student Login

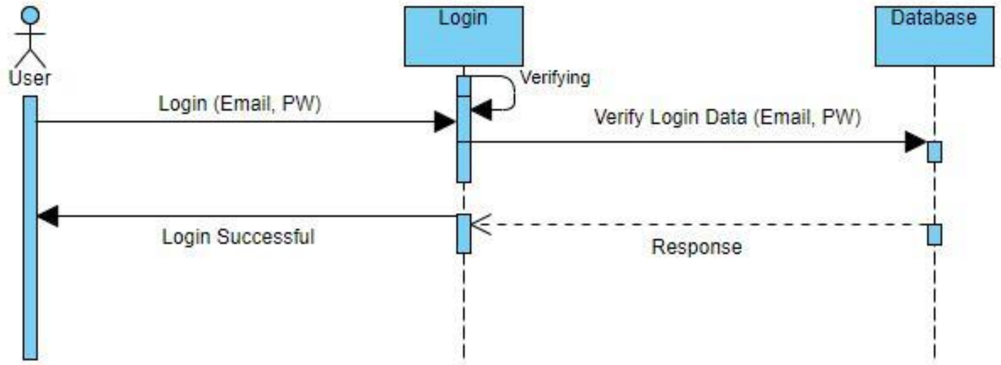


Figure 6: SD: Student Login

The above figure is of student login in sequence diagram form.

3. View Trending Courses

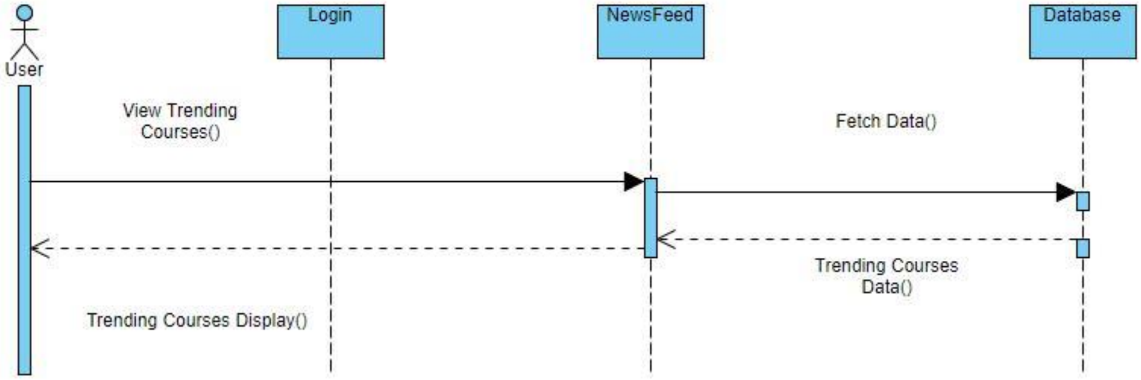


Figure 7: SD: View Trending Courses

The above figure is of view trending courses in sequence diagram form.

4. View On-Campus Courses

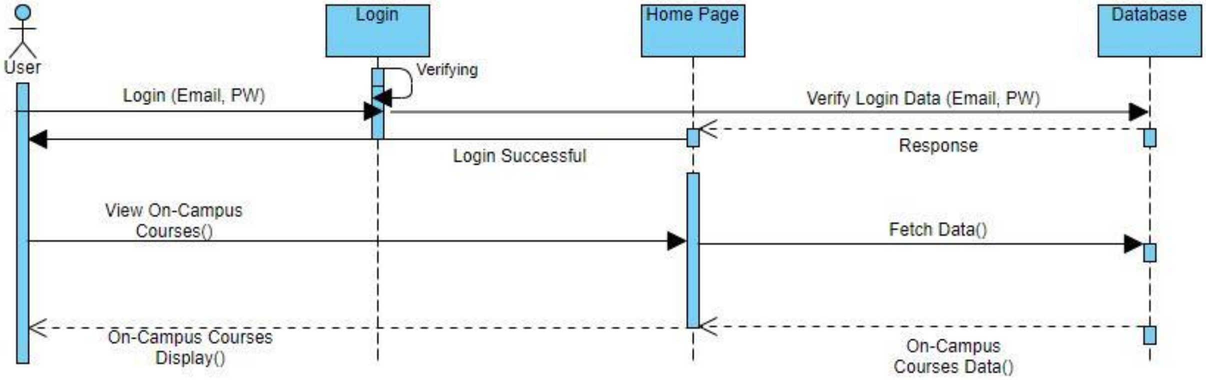


Figure 8: SD: View On-Campus Courses

The above figure is of view on-campus courses in sequence diagram form.

5. View Online Courses

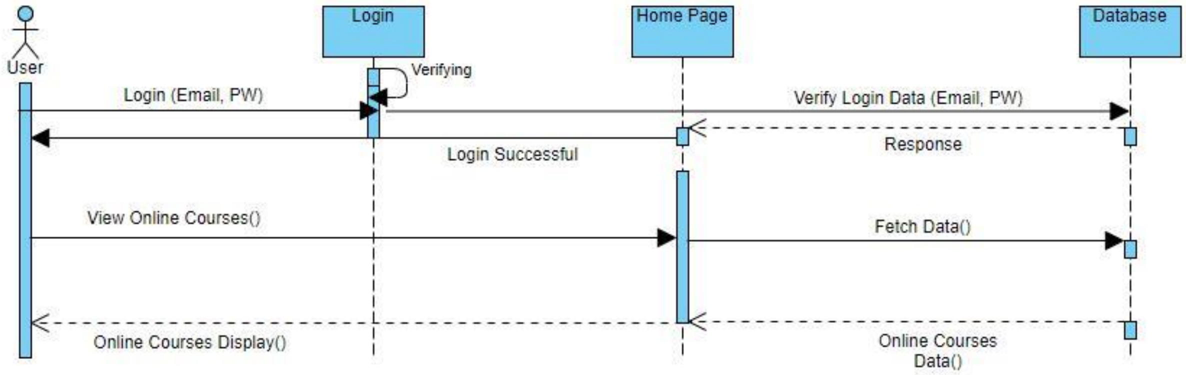


Figure 9: SD: View Online Courses

The above figure is of view online courses in sequence diagram form.

6. View Recommendation

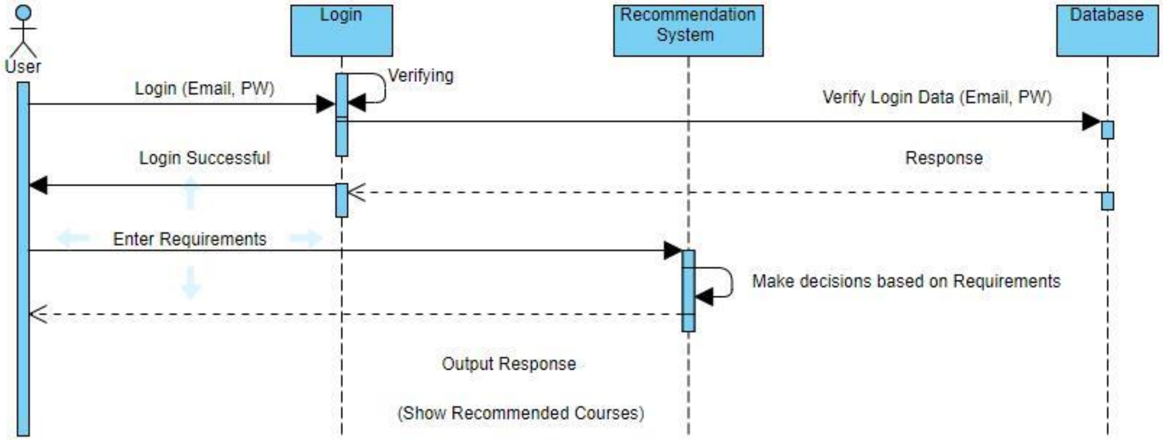


Figure 10: SD: View Recommendation

The above figure is of view recommendation in sequence diagram form.

7. View Difficulty and Rating

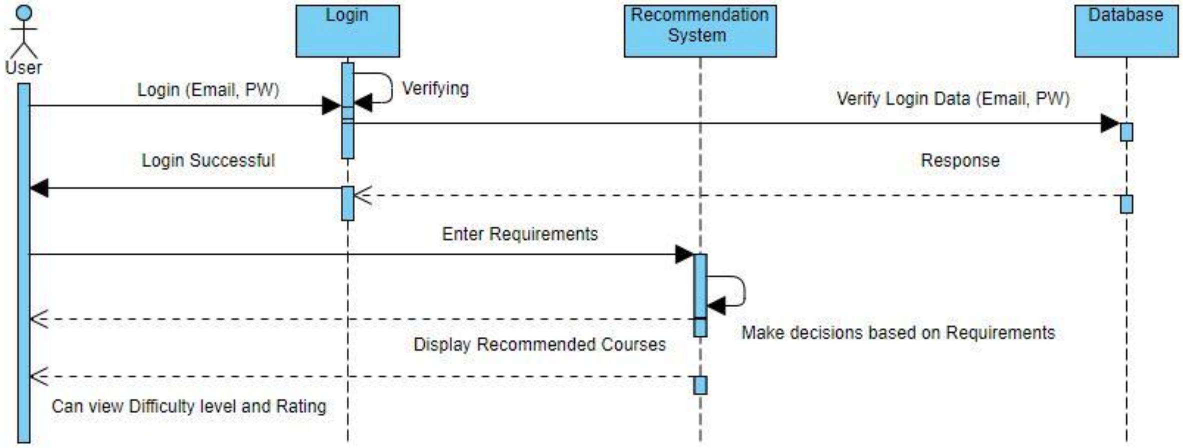

User
Login
Recommendation System
Database
Login (Email, PW)
Verifying
Verify Login Data (Email, PW)
Login Successful
Response
Enter Requirements
Make decisions based on Requirements
Display Recommended Courses
Can view Difficulty level and Rating

Figure 11: SD: View Difficulty and Rating

The above figure is of view difficulty and rating in sequence diagram form.

8. View Cart

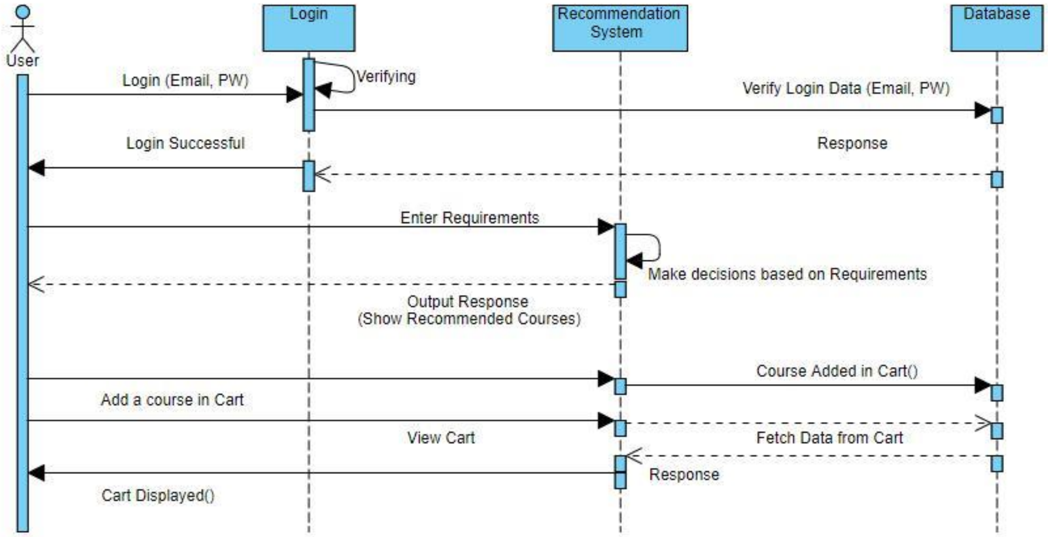


Figure 12: SD: View Cart

The above figure is of view cart in sequence diagram form.

9. Manage Students

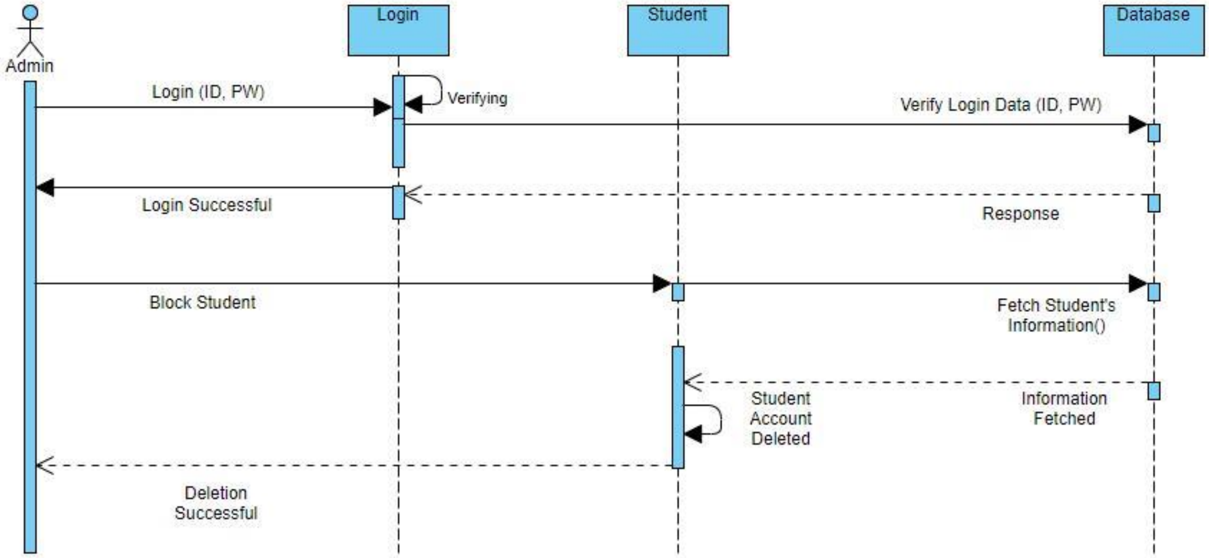


Figure 13: SD: Manage Students.

The above figure is of manage students in sequence diagram form.

10. Post Newsfeed

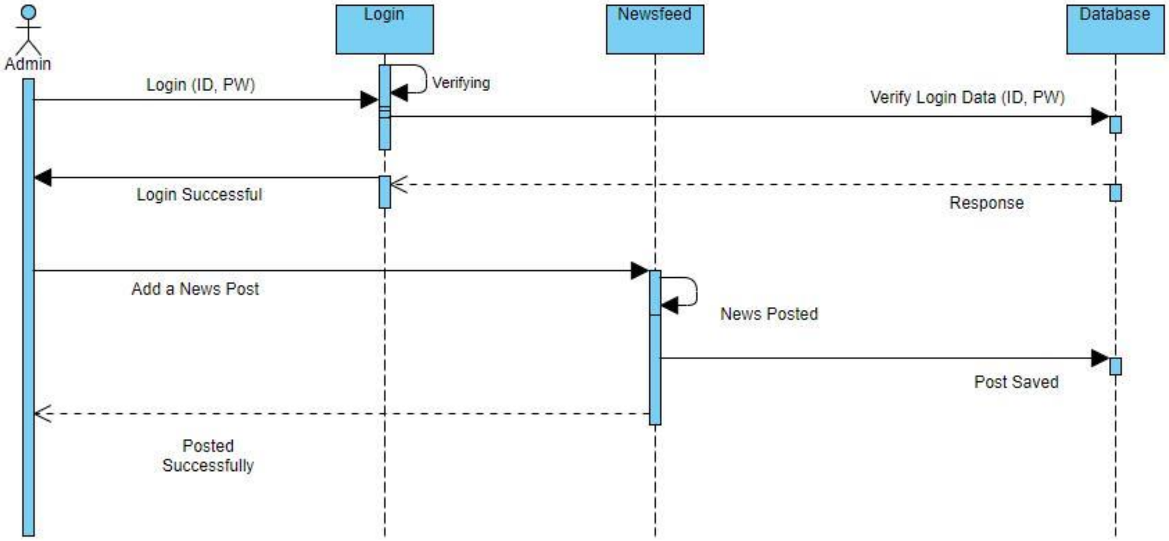


Figure 14: SD: Post Newsfeed

The above figure is of posting newsfeed in sequence diagram form.

11. Add Courses

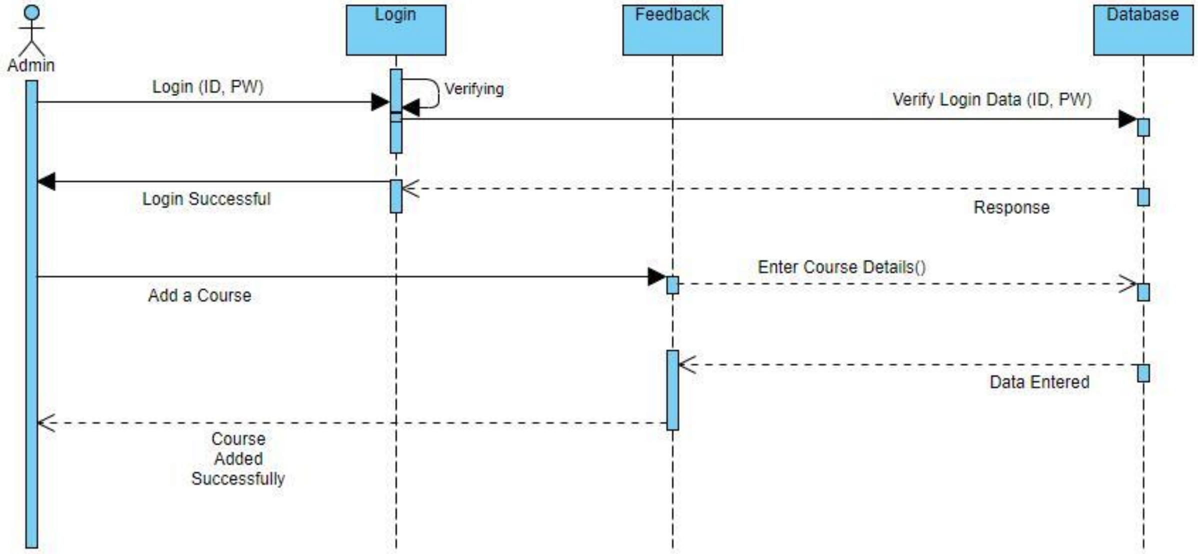


Figure 15: SD: Add Courses

The above figure is of add courses in sequence diagram form.

12. Update Courses

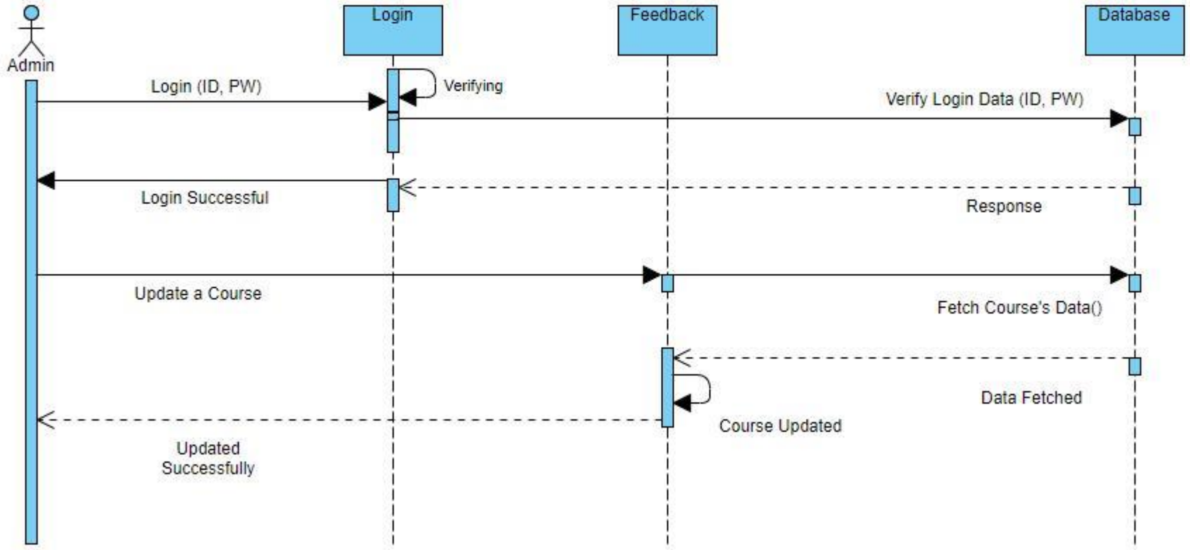


Figure 16: SD: Update Courses

The above figure is of update courses in sequence diagram form.

5.3 Collaboration Diagrams

1. Student Signup

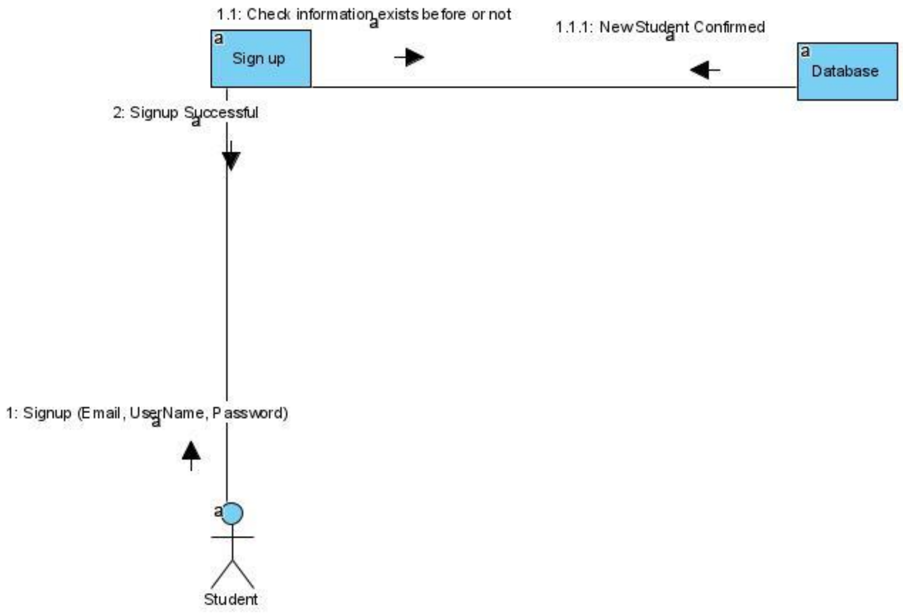

1.1: Check information exists before or not
1.1.1: New Student Confirmed
a
Sign up
a
Database
2: Signup Successful
1: Signup (Email, UserName, Password)
a
Student

Figure 17: CD: Student Signup

The above figure is of student sign up in collaboration diagram form.

2. Student Login

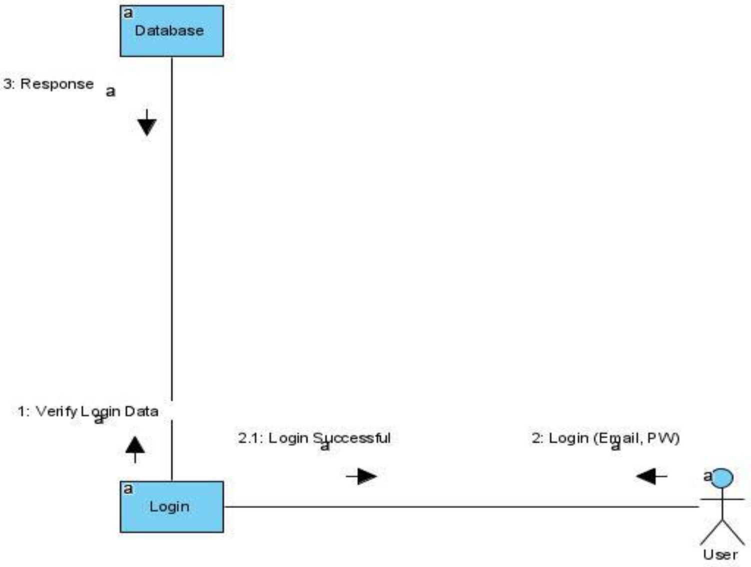


Figure 18: CD: Student Login

The above figure is of student login in collaboration diagram form.

3. View Trending Courses

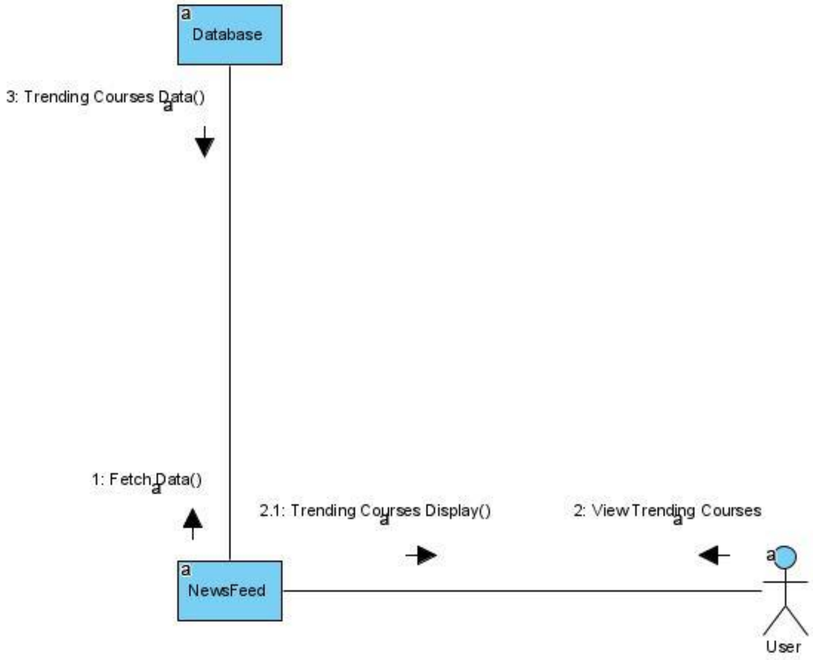


Figure 19: CD: View Trending Courses

The above figure is of view trending courses in collaboration diagram form.

4. View On-Campus Courses

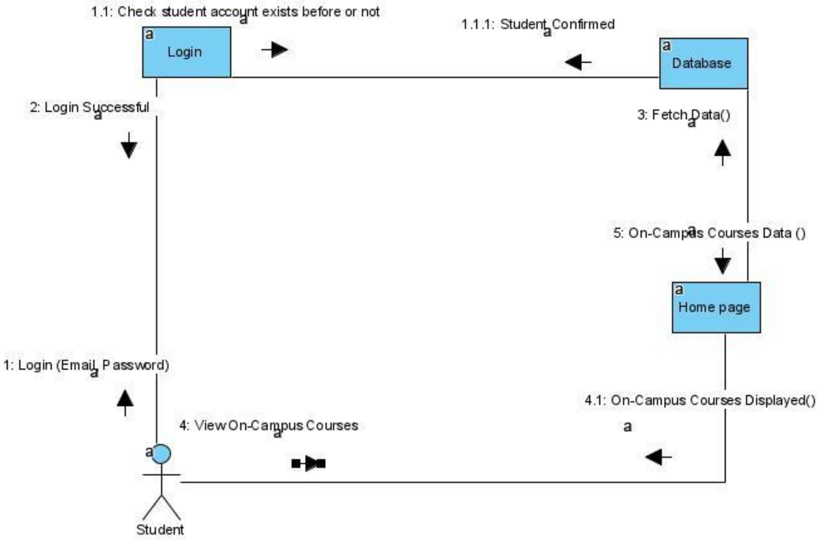

1.1: Check student account exists before or not
a
Login
1.1.1: Student Confirmed
a
Database
2: Login Successful
3: Fetch Data()
5: On-Campus Courses Data ()
a
Home page
1: Login (Email, Password)
4.1: On-Campus Courses Displayed()
4: View On-Campus Courses
a
Student

Figure 20: CD: View On-Campus Courses

The above figure is of view on-campus courses in collaboration diagram form.

5. View Online Courses

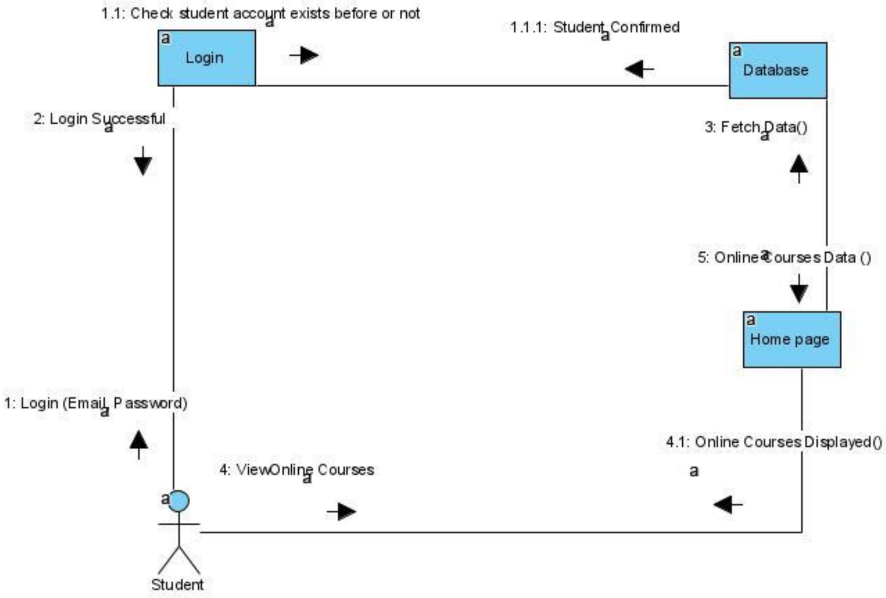

1.1: Check student account exists before or not
Login
1.1.1: Student Confirmed
Database
2: Login Successful
3: FetchData()
5: Online Courses Data ()
Home page
1: Login (Email, Password)
4.1: Online Courses Displayed()
4: ViewOnline Courses
Student

Figure 21: CD: View Online Courses

The above figure is of view online courses in collaboration diagram form.

6. View Recommendation

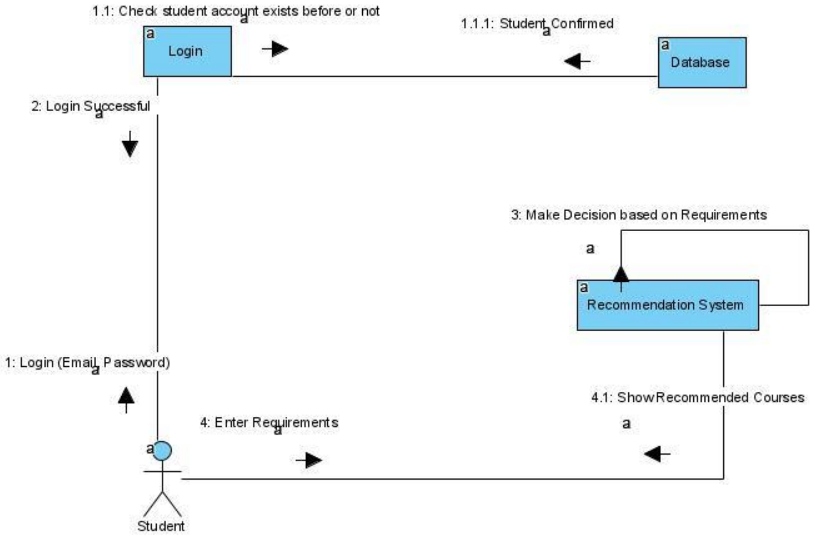

1.1: Check student account exists before or not
Login
1.1.1: Student Confirmed
Database
2: Login Successful
3: Make Decision based on Requirements
Recommendation System
1: Login (Email, Password)
4.1: ShowRecommended Courses
4: Enter Requirements
Student

Figure 22: CD: View Recommendation

The above figure is of view recommendation in collaboration diagram form.

7. View Difficulty and Rating

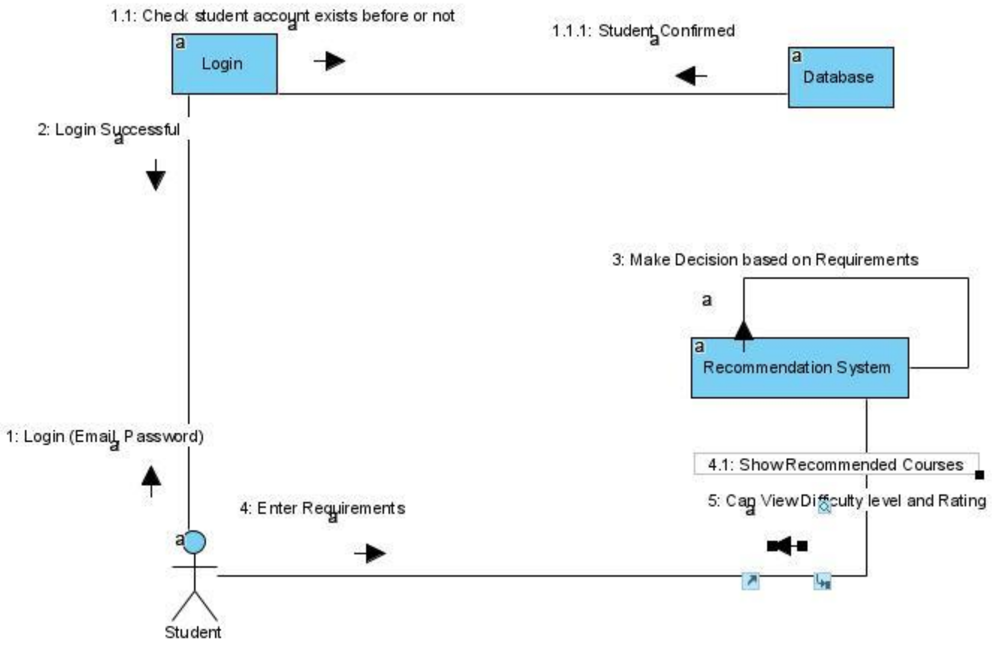

1.1: Check student account exists before or not
Login
1.1.1: Student Confirmed
Database
2: Login Successful
3: Make Decision based on Requirements
Recommendation System
1: Login (Email, Password)
4.1: ShowRecommended Courses
5: Can ViewDifficulty level and Rating
4: Enter Requirements
Student

Figure 23: CD: View Difficulty and Rating

The above figure is of view difficulty and rating in collaboration diagram form. 8. View Cart

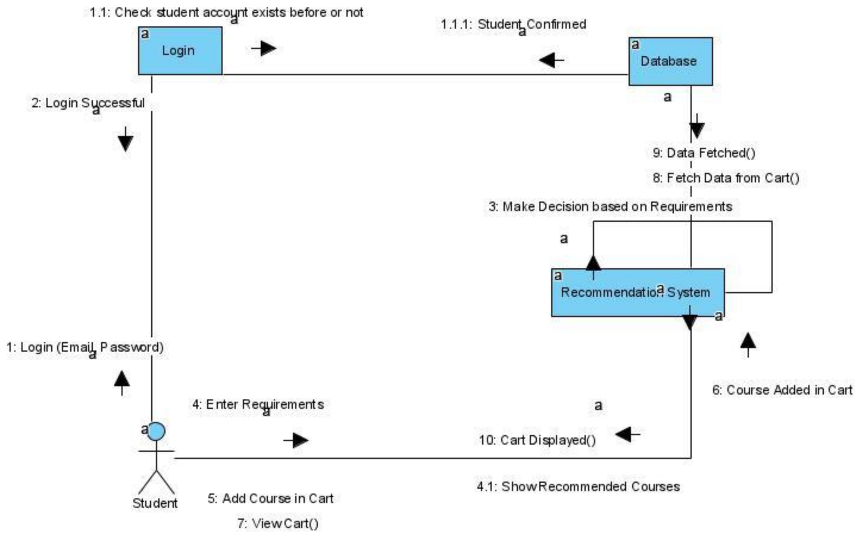


Figure 24: CD: View Cart

The above figure is of view cart in collaboration diagram form.

9. Manage Students

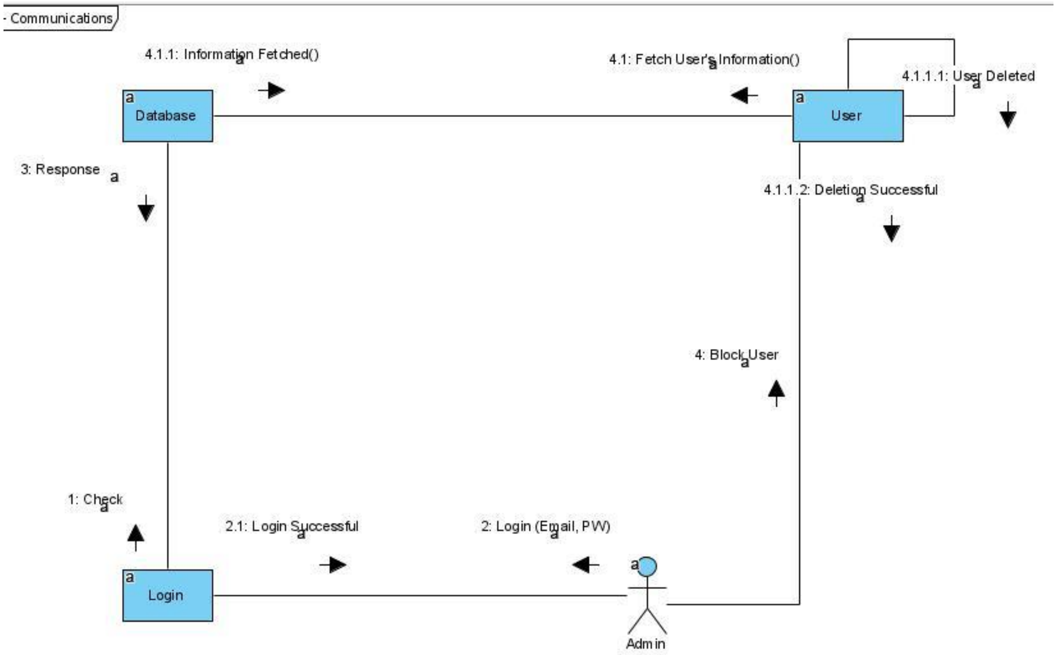


Figure 25: CD: Manage Students

The above figure is of manage students in collaboration diagram form.

10. Post Newsfeed

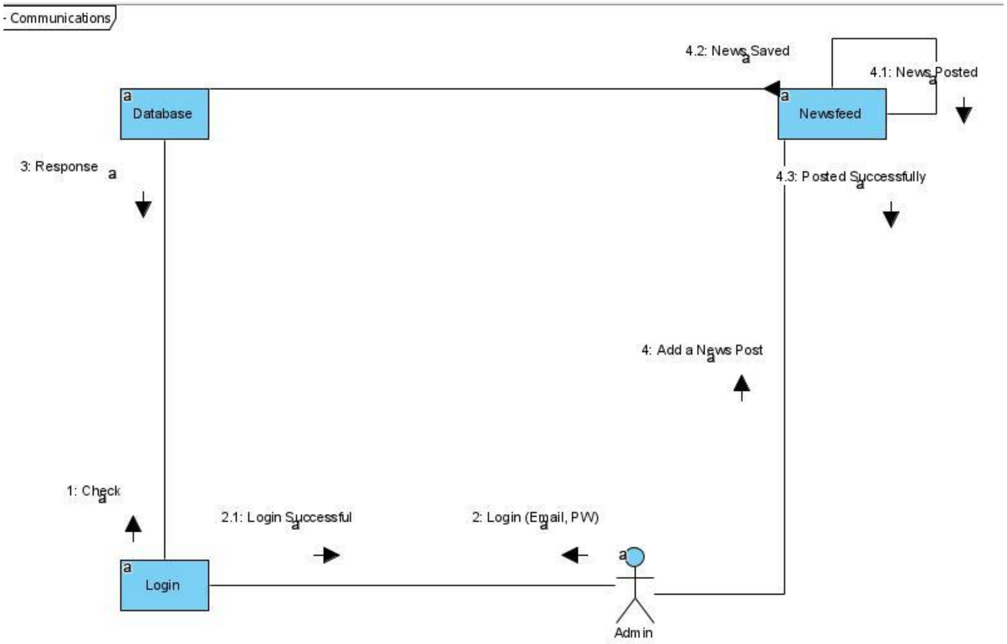


Figure 26: CD: Post Newsfeed

The above figure is of post newsfeed in collaboration diagram form.

11. Add Courses

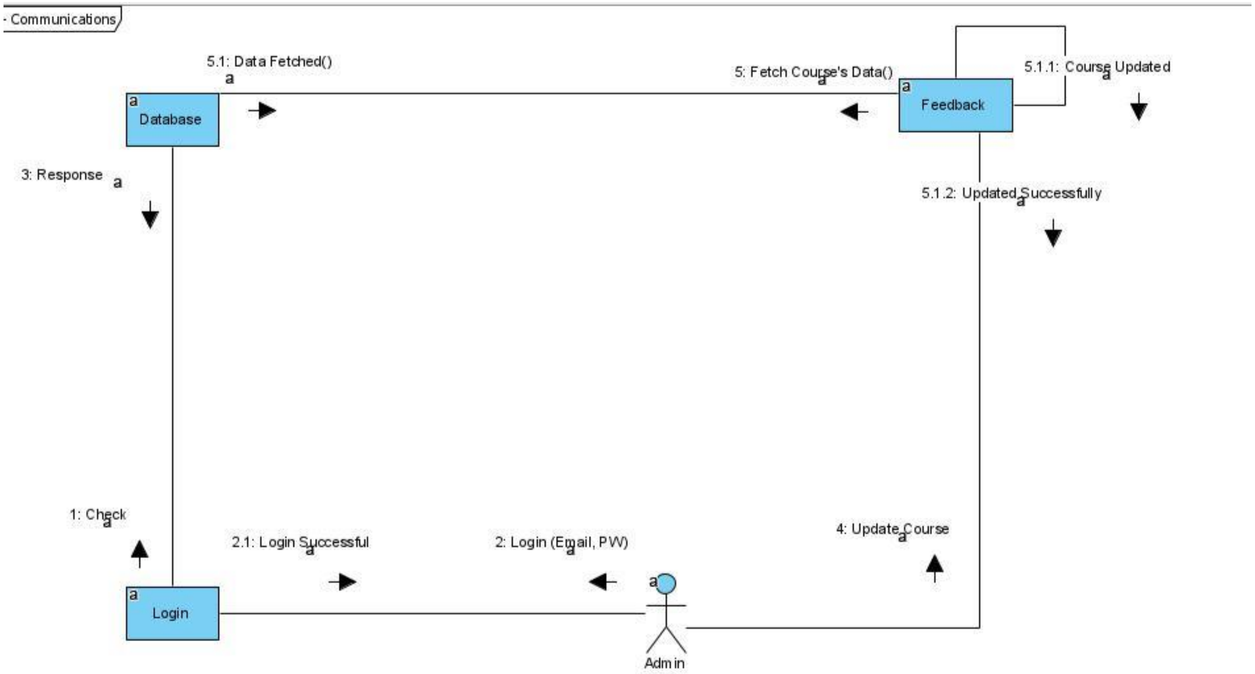


Figure 27: CD: Add Courses

The above figure is of add courses in collaboration diagram form.

12. Update Courses

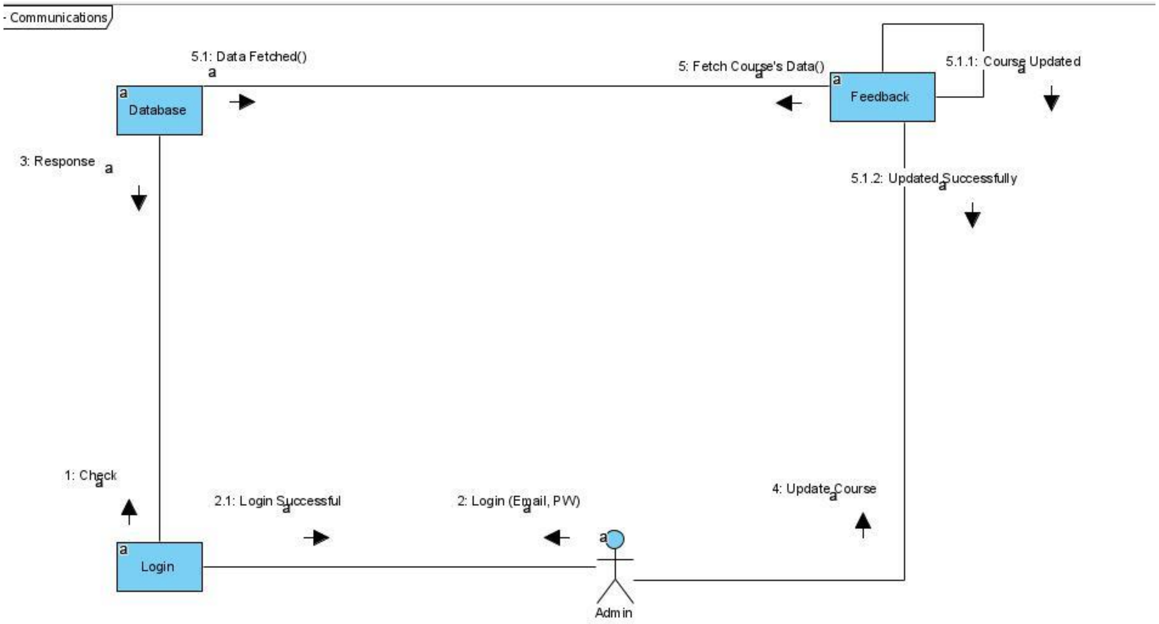


Figure 28: CD: Update Courses

The above figure is of update courses in collaboration diagram form.

5.4 Entity Relationship Diagram

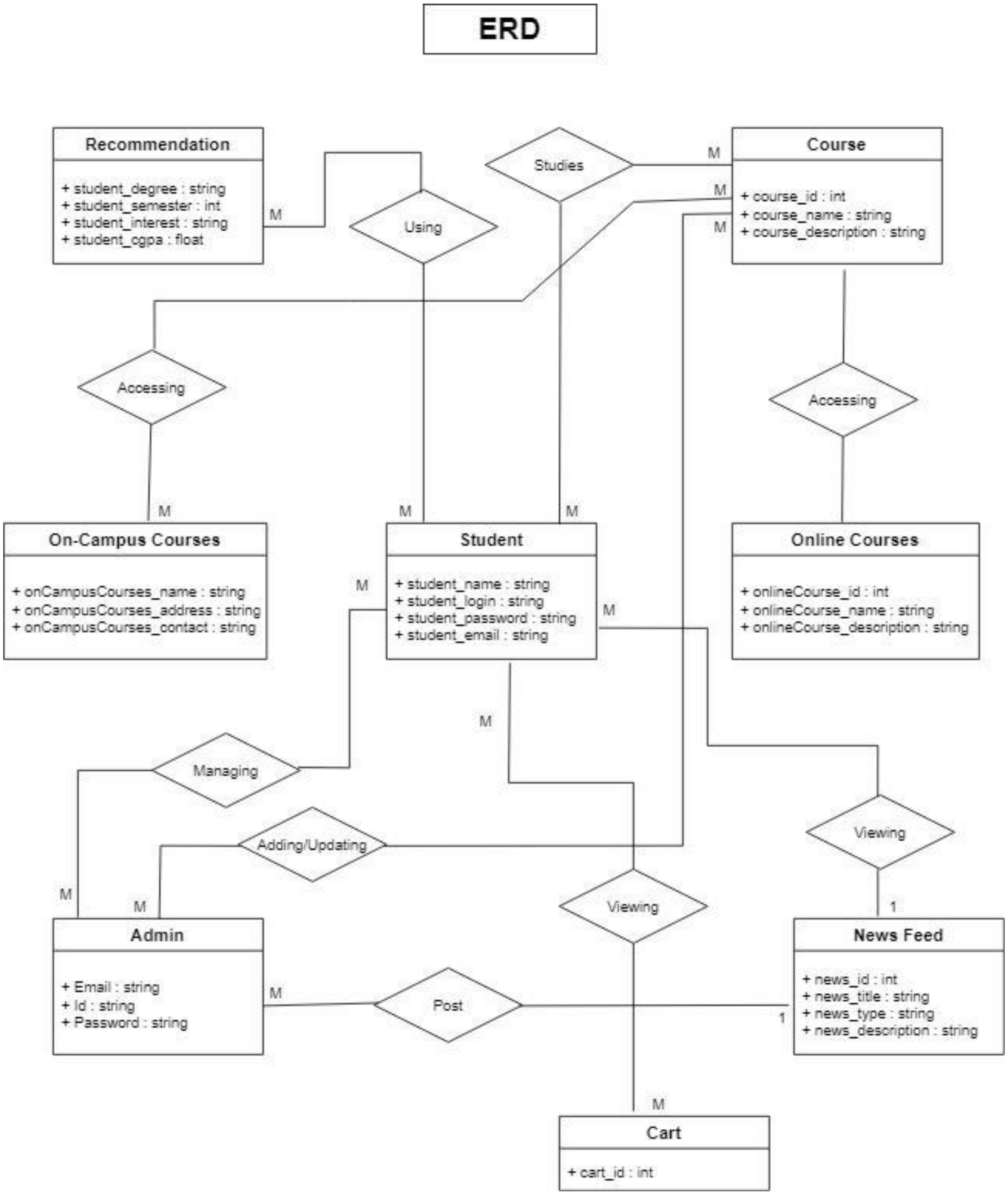


Figure 29: Entity Relationship Diagram

The above diagram shows the relationships between different entities.

6.1 Development Setup

Django (Python framework for developing website's backend)

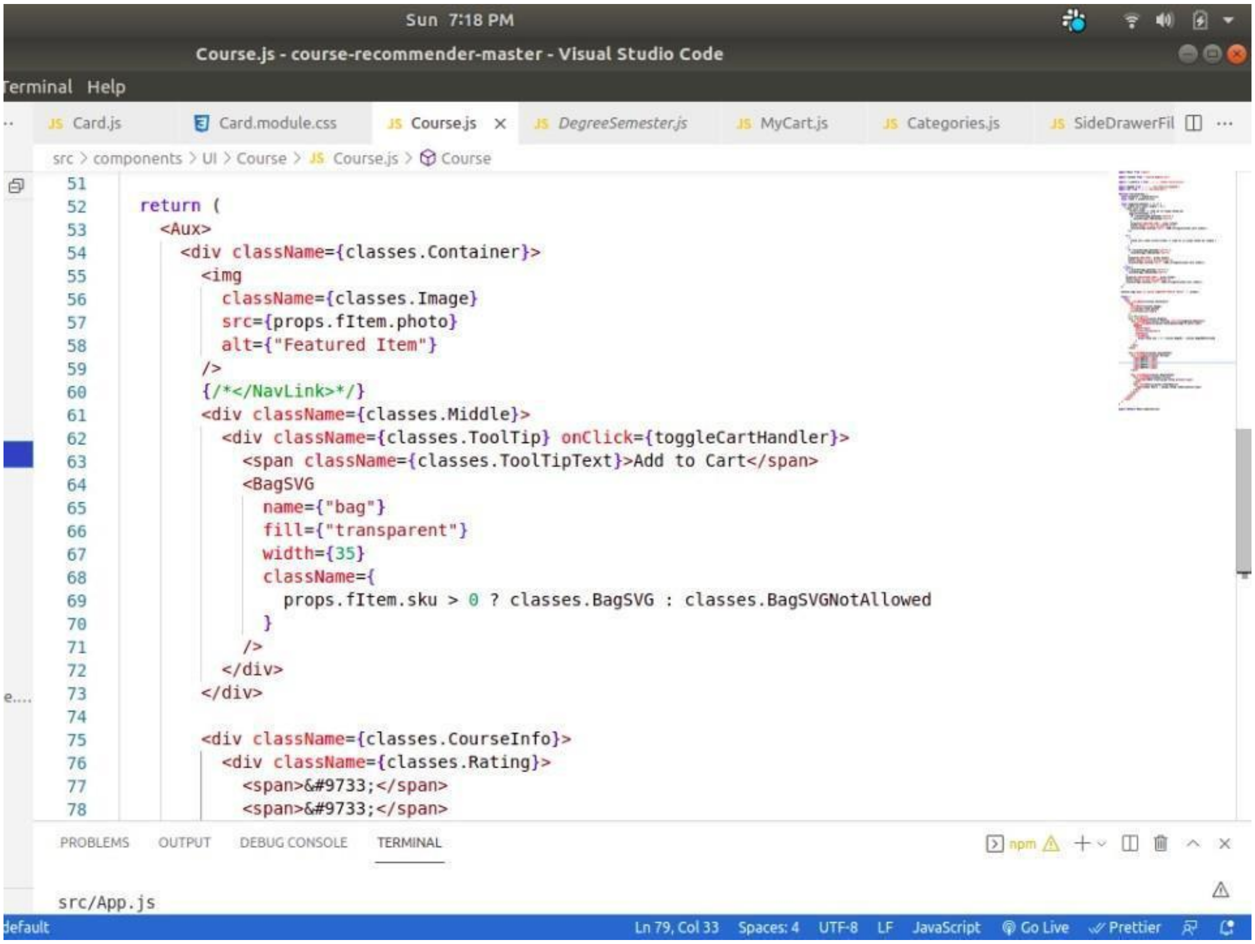
ReactJS (JavaScript framework for developing website's frontend)

MySQL (Database)

6.2 Deployment setup

We faced minor issues in deploying the system as it was a significant task to implement this system. But our advisor made sure that we had all the support and means to create this recommendation system.

6.3 Algorithms



```

    return (
      <Aux>
        <div className={classes.Container}>
          <img
            className={classes.Image}
            src={props.fItem.photo}
            alt={"Featured Item"}
          />
          {/*</NavLink>*/}
          <div className={classes.Middle}>
            <div className={classes.ToolTip} onClick={toggleCartHandler}>
              <span className={classes.ToolTipText}>Add to Cart</span>
              <BagSVG
                name={"bag"}
                fill={"transparent"}
                width={35}
                className={
                  props.fItem.sku > 0 ? classes.BagSVG : classes.BagSVGNotAllowed
                }
              />
            </div>
          </div>

          <div className={classes.CourseInfo}>
            <div className={classes.Rating}>
              <span>&#9733;</span>
              <span>&#9733;</span>
```

```
src/App.js
```

The above picture shows the algorithm used in the project.

6.4 Constraints

6.4.1 Assumptions

We have assumed that the following points are true.

- The student will get verification mail while signing up for the website.
- The system has a good interface that users will like.
- The recommendation section will generate real-time result for the student.
- Database maintenance will be done by the administrator.
- The system will show on-campus and online courses.

6.4.2 System constraints

- **Scope:** The system is mostly useful to students and to the people belonging to the CS industry. Thus, it is restricted to only this type of users.
- **Benefits:** Our system will be advantageous to students who are in search for advice regarding what to study in the next semester. Furthermore, this website will provide online sources to study from, and suggest you universities that are offering a certain course.
- **Cost:** Our project will demand not a dime from its users as it is a public website. Students just have to sign up to use the recommendation service.
- **Time:** This system will cut short the time taken by the student to make a decision.

6.4.3 Restrictions

- The student is required to sign up for using the recommendation service of the system.

6.4.4 Limitations

The motivation behind this project is to provide the students with a platform that will help them make decisions regarding their courses while taking in account their interest. The software is unable to provide:

- Our project will not make the decision for the student; rather help them in the process.
- Student cannot study from the website itself; it will provide the online links for the courses.

7. TESTING

7.1 Extended Test Cases

The below table displays the student sign up test case.

<table>
<tr><td colspan="3"><b>Test Case ID:</b> TC-01</td><td colspan="4"><b>Test Design By</b>: Shuja Ur Rehman</td></tr>
<tr><td colspan="3"><b>Test Module Name</b>: Sign upon Smart Course Recommender System.</td><td colspan="4"><b>Test Design Date</b>: September 21, 2022</td></tr>
<tr><td colspan="3"><b>Test Priority</b>: High</td><td colspan="4"><b>Test Executed By</b>: Syed Asad Ali Bukhari</td></tr>
<tr><td colspan="3"><b>Test Title</b>: To test Sign up function</td><td colspan="4"><b>Test Executed Date</b>: October 25, 2022</td></tr>
<tr><td colspan="3"><b>Description</b>: Check Student Sign up with Invalid Information.</td><td colspan="4"></td></tr>
<tr><td colspan="7"><b>Pre-condition:</b> Student should have access to the website and must enter an invalid email address.</td></tr>
<tr><td colspan="7"><b>Dependencies</b>: UC-1</td></tr>
<tr><td><b>Step</b></td><td><b>Test step</b></td><td><b>Test data</b></td><td><b>Expected result</b></td><td><b>Actual result</b></td><td><b>Status: Pass/Fail</b></td><td><b>Notes</b></td></tr>
<tr><td><b>1</b></td><td>Navigate the Sign up page.</td><td></td><td></td><td rowspan="4">As expected.</td><td rowspan="4">Pass</td><td></td></tr>
<tr><td><b>2</b></td><td>Enter invalid Email</td><td>Email:
asad2018Isc60 @pu.edu.pk</td><td></td><td></td></tr>
<tr><td><b>3</b></td><td>Enter Password.</td><td>Password:
pu@12345</td><td></td><td></td></tr>
<tr><td><b>4</b></td><td>Click on Sign up button.</td><td></td><td>System should give error “Invalid Email”.</td><td></td></tr>
<tr><td colspan="7"><b>Post condition:</b> Student Sign up will be rejected and will give error.</td></tr>
</table>

Table 17: TC-01: Sign upon SCRS

The below table displays the student login test case.

Test Case ID: TC-02	**Test Design By**: Shuja Ur Rehman
Test Module Name: Login on Smart Course Recommender System.	**Test Design Date**: September 21, 2022
Test Priority: High	**Test Executed By**: Syed Asad Ali Bukhari
Test Title: To test Login function	**Test Executed Date**: October 25, 2022
Description: Check Student Login with Valid Data.	
Pre-condition: Student should have access to the website and must have signed up beforehand.	
Dependencies: **UC-2**	

Step	**Test step**	**Test data**	**Expected result**	**Actual result**	**Status: Pass/Fail**	**Notes**
1	Navigate the Login page.			As expected.	Pass	
2	Enter Username	Username: asad2018Isc60 @pu.edu.pk				
3	Enter Password.	Password: pu@1234				
4	Click on Login button.		Student login will be successful.			
Post condition: Student Login time and date will be stored in the database.						

Table 18: TC-02: Login on SCRS

The below table displays the view trending courses test case.

Test Case ID: TC-03	**Test Design By**: Shuja Ur Rehman
Test Module Name: View Trending Courses on Smart Course Recommender System.	**Test Design Date**: September 21, 2022
Test Priority: High	**Test Executed By**: Syed Asad Ali Bukhari
Test Title: To test Trending Courses Button	**Test Executed Date**: October 25, 2022
Description: Check if Trending Courses are visible or not.	
Pre-condition: Student should have access to the website.	
Dependencies: UC-3	

Step	**Test step**	**Test data**	**Expected result**	**Actual result**	**Status: Pass/Fail**	**Notes**
1	Openthe website.			As expected.	Pass	
2	Click on “Trending Courses” Button		Student should be able to see the trending courses.			
Post condition: Student should be able to see the Trending Courses data.						

Table 19: TC-03: View Trending Courses on SCRS

The below table displays the view recommendation test case.

Test Case ID: TC-04	**Test Design By**: Shuja Ur Rehman
Test Module Name: View Recommendation on Smart Course Recommender System.	**Test Design Date**: September 21, 2022
Test Priority: High	**Test Executed By**: Syed Asad Ali Bukhari
Test Title: To test Course Recommendation function.	**Test Executed Date**: October 25, 2022
Description: Check whether the Recommendation function works or not.	
Pre-condition: Student should have access to the website and must have signed up beforehand.	
Dependencies: UC-6	

Step	Test step	Test data	Expected result	Actual result	Status: Pass/Fail	Notes
1	Navigate the Login page.			As expected.	Pass	
2	Enter Username and Password.	Username: asad2018Isc60 @pu.edu.pk Password: pu@12345				
3	Click on Login button.		Student login will be successful.			
4	Click on Recommendation and fill the requirements.	Semester, Degree, City				
5	Click on Result Button.		System will generate the output and display it to the student.			

Post condition: Student's Recommendation will be stored in the database.

Table 20: TC-04: View Recommendation on SCRS

The below table displays the on-campus courses button test case.

Test Case ID: TC-05	**Test Design By**: Shuja Ur Rehman
Test Module Name: On-Campus Courses Button on home page.	**Test Design Date**: September 21, 2022
Test Priority: High	**Test Executed By**: Syed Asad Ali Bukhari
Test Title: To test On-campus Courses button	**Test Executed Date**: October 25, 2022
Description: Testing the On-campus Courses button	
Pre-condition: Student should have logged into the system and is on the home page where the "On-campus Courses" button is located.	
Dependencies: UC-4	

Step	**Test step**	**Test data**	**Expected result**	**Actual result**	**Status: Pass/Fail**	**Notes**
1	Login to the system	Username: asad2018Isc60@pu.edu.pk Password: pu@12345	Student login will be successful.	As expected.	Pass	
2	Click on On-campus Courses Button.		The system will display the courses			

Post condition: Button is working successfully.

Table 21: TC-05: On-campus Courses Button on home page

The below table displays the difficulty level test case.

Test Case ID: TC-06	**Test Design By**: Shuja Ur Rehman
Test Module Name: Difficulty Level on Recommendation page.	**Test Design Date**: September 21, 2022
Test Priority: High	**Test Executed By**: Syed Asad Ali Bukhari
Test Title: To test Difficulty Level	**Test Executed Date**: October 25, 2022
Description: Testing whether the system shows the difficulty level or not.	
Pre-condition: Student should have logged into the system and is using the recommendation service.	
Dependencies: UC-7	

Step	**Test step**	**Test data**	**Expected result**	**Actual result**	**Status: Pass/Fail**	**Notes**
1	Login to the system	Username: asad2018Isc60@pu.edu.pk Password: pu@12345	Student login will be successful.	As expected.	Pass	
2	Click on Recommendation button					
3	Enter the fields required.					
4	Click on Apply button.		The student’s recommendation should be displayed.			
5	See the difficulty level.		The difficulty level should be visible alongside the courses.			

Post condition: The difficulty level is being displayed properly.

Table 22: TC-06: Difficulty Level on Recommendation Page

The below table displays the view newsfeed test case.

<table>
<tr><td colspan="3"><b>Test Case ID:</b> TC-07</td><td colspan="4"><b>Test Design By</b>: Shuja Ur Rehman</td></tr>
<tr><td colspan="3"><b>Test Module Name</b>: View Newsfeed on Smart Course Recommender System.</td><td colspan="4"><b>Test Design Date</b>: September 21, 2022</td></tr>
<tr><td colspan="3"><b>Test Priority</b>: Medium</td><td colspan="4"><b>Test Executed By</b>: Syed Asad Ali Bukhari</td></tr>
<tr><td colspan="3"><b>Test Title</b>: To test Newsfeed (Home) Button</td><td colspan="4"><b>Test Executed Date</b>: October 25, 2022</td></tr>
<tr><td colspan="3"><b>Description</b>: Check if Newsfeed is visible or not.</td><td colspan="4"></td></tr>
<tr><td colspan="7"><b>Pre-condition:</b> Student should have access to the website.</td></tr>
<tr><td colspan="7"><b>Dependencies</b>: UC-3</td></tr>
<tr><td><b>Step</b></td><td><b>Test step</b></td><td><b>Test data</b></td><td><b>Expected result</b></td><td><b>Actual result</b></td><td><b>Status: Pass/Fail</b></td><td><b>Notes</b></td></tr>
<tr><td><b>1</b></td><td>Navigate the website.</td><td></td><td></td><td rowspan="2">As expected.</td><td rowspan="2">Pass</td><td></td></tr>
<tr><td><b>2</b></td><td>The screen you see is Newsfeed.</td><td></td><td>The Newsfeed should be visible.</td><td></td></tr>
<tr><td colspan="7"><b>Post condition:</b> Student should be able to see the Newsfeed webpage.</td></tr>
</table>

Table 23: TC-07: View Newsfeed on SCRS

The below table displays the view cart test case.

Test Case ID: TC-8	**Test Design By**: Shuja Ur Rehman
Test Module Name: View Cart	**Test Design Date**: September 21, 2022
Test Priority: High	**Test Executed By**: Syed Asad Ali Bukhari
Test Title: To test Cart	**Test Executed Date**: October 25, 2022
Description: Check if Cart is working properly or not.	
Pre-condition: Student should have access to the website and are viewing the online courses section.	
Dependencies: UC-8	

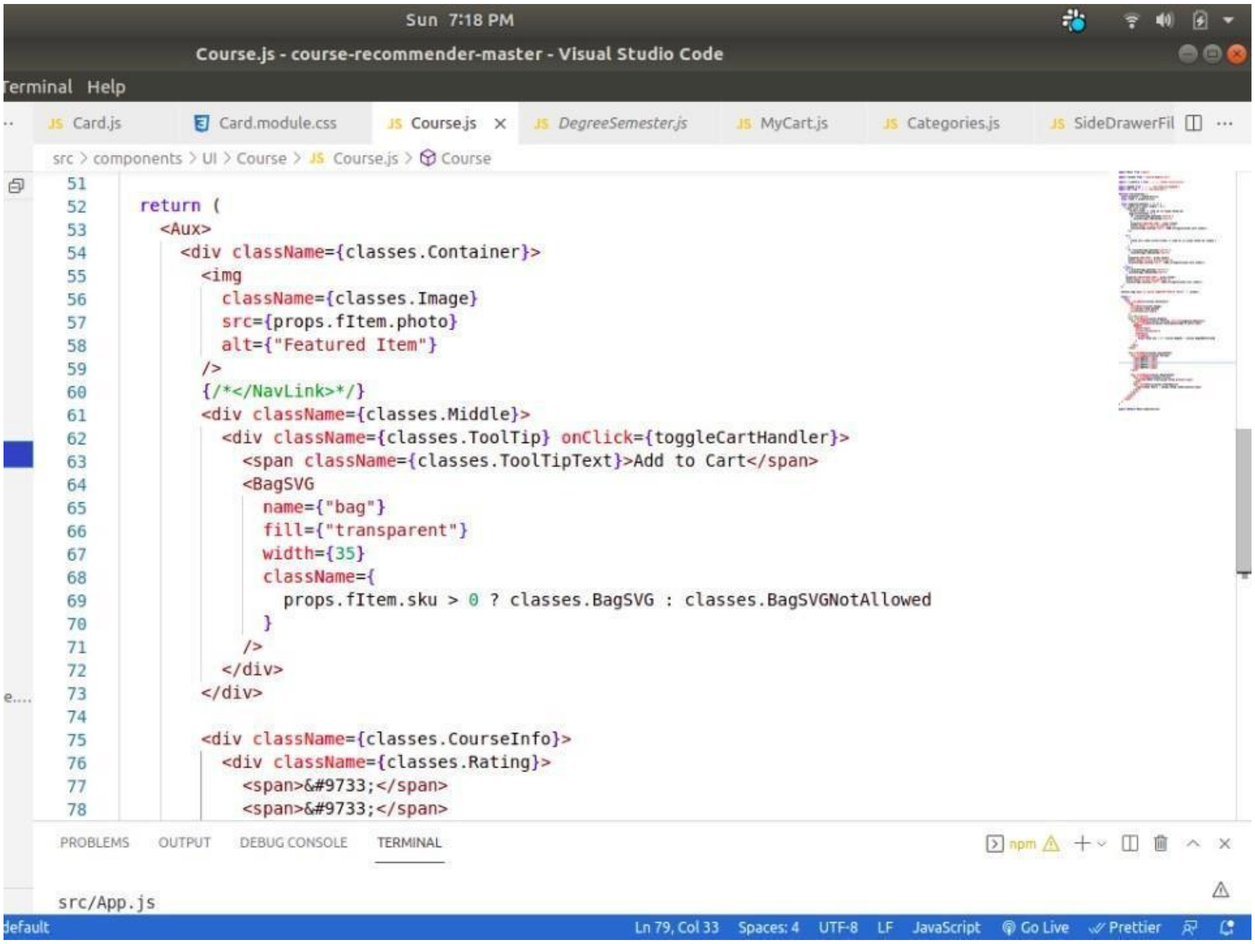
Step	**Test step**	**Test data**	**Expected result**	**Actual result**	**Status: Pass/Fail**	**Notes**
1	Navigate to the Login page and log into the system.	Username: Asad2018Isc60@pu.edu.pk Password: pu@12345		As expected.	Pass	
2	View On-campus Courses and select a course.		Student should be able to select a course from the list.			
3	Add the course in the cart.		The course is added to the cart.			

Post condition: Student should be able to see the Free Courses data.

Table 24: TC-08: View Cart

7.2 Decision Table

7.2.1 Code snippet



```

    return (
      <Aux>
        <div className={classes.Container}>
          <img
            className={classes.Image}
            src={props.fItem.photo}
            alt={"Featured Item"}
          />
          {/*</NavLink>*/}
          <div className={classes.Middle}>
            <div className={classes.ToolTip} onClick={toggleCartHandler}>
              <span className={classes.ToolTipText}>Add to Cart</span>
              <BagSVG
                name={"bag"}
                fill={"transparent"}
                width={35}
                className={
                  props.fItem.sku > 0 ? classes.BagSVG : classes.BagSVGNotAllowed
                }
              />
            </div>
          </div>

          <div className={classes.CourseInfo}>
            <div className={classes.Rating}>
              <span>&#9733;</span>
              <span>&#9733;</span>
```

```
src/App.js
```

The above picture shows a part of the code used in making of the system.

7.2.2 Decision coverage table

a) Sign up Decision Table

The following table shows the actions of the system performed upon certain conditions.

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Student entered Username.	F	T	T	T	T
Student entered Email.	-	F	T	T	T
Student entered Password.	-	-	F	T	T
Student account is registered.	-	-	-	F	T
Actions	Error	Error	Error	Error	Signup is successful.

Table 14: Sign up Decision Table

b) View Cart Decision Table

The following table shows the actions of the system performed upon certain conditions.

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Login to student account.	F	T	T	T	T
Student views the online courses section.	-	F	T	T	T
Click on Add to Cart button.	-	-	F	T	T
Course is added into the cart.	-	-	-	F	T
Actions	Error	Error	Error	Error	Course added successfully.

Table 15: View Cart Decision Table

7.3 Traceability Matrix

7.3.1 RID vs UCID (Requirements vs Use Cases)

The following table shows how much of the two properties overlap in the system.

UCID/RID	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
UC-1	✓	✓	✓											
UC-2	✓	✓	✓											
UC-3	✓	✓												
UC-4								✓	✓	✓	✓			
UC-5								✓	✓	✓	✓			
UC-6	✓	✓		✓	✓	✓	✓							
UC-7	✓	✓								✓		✓		
UC-8	✓	✓								✓		✓		
UC-9													✓	✓
UC-10													✓	✓
UC-11													✓	✓
UC-12													✓	✓

Table 27: RID vs. UCID Table

7.3.2 Test Cases (RID vs TID)

The following table shows how much of the two properties overlap in the system.

TID/RID	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
TC-01	✓													
TC-02	✓	✓												
TC-03	✓	✓	✓											
TC-04						✓	✓✓✓							
TC-05								✓	✓	✓	✓			
TC-06										✓		✓		
TC-07			✓											
TC-08								✓						

Table 28: RID vs. TID Table

8. RESULTS/OUTPUT/STATISTICS

8.1 %completion

Formula: (Fulfilled requirement / Total requirement) *100

95%.

8.2 %accuracy

Formula: (Fulfilled requirement / Total requirement) *100

97%

8.3 %correctness

Formula: (Fulfilled requirement / Total requirement) *100

90%

9. CONCLUSION

In conclusion to this documentation, we have successfully created a Smart Course Recommender System whose purpose is to make recommendations related to the courses that student want to study. Moreover, it provides a choice to choose from online and on-campus courses for different computer science related courses to study from and the on-campus courses that are being taught at PUCIT. We have implemented this project using Django and ReactJS frameworks that have immense significance in the technology industry. We are also using Rest API to transfer data from frontend to backend and vice versa. We collected our own dataset from two main sources that have a huge collection of courses in almost every field.

10. FUTURE WORK

As of right now, we have only selected computer science field for the students to take recommendation from, but in the future, we can expand this area as there are a lot of fields in the world, such as medicine, business, architecture and so on. We can accumulate courses related to these fields and many more students (belonging to these fields) can use the recommendation service. Moreover, we can introduce feedback of the system and of the courses listed in the website.

11. Bibliography

11.1 Research papers

A research paper will be in works once the system is made public.

11.2 Other References

[1] Resnick P. and Varian H., Recommender Systems, Communication of the ACM, 40(3), 56-58, March 1997.

[2] Montaner M., Lopez B., and Josep R., A Taxonomy of Recommender Agents on the Internet, Artificial Intelligence Review, 19, 285-330, 2003.

[3] Bridge D., Product recommendation systems: A new direction. In R.Weber, Wangenheim, C., eds.: Procs. of the Workshop Programme at the Fourth International Conference on Case-Based Reasoning. Vancouver, Canada, 79-86, 2001.

[4] Burke R., Hybrid Recommender Systems: Survey and Experiments, User Modelling and User-Adapted. Interaction 12, 4, 331-370, 2002.