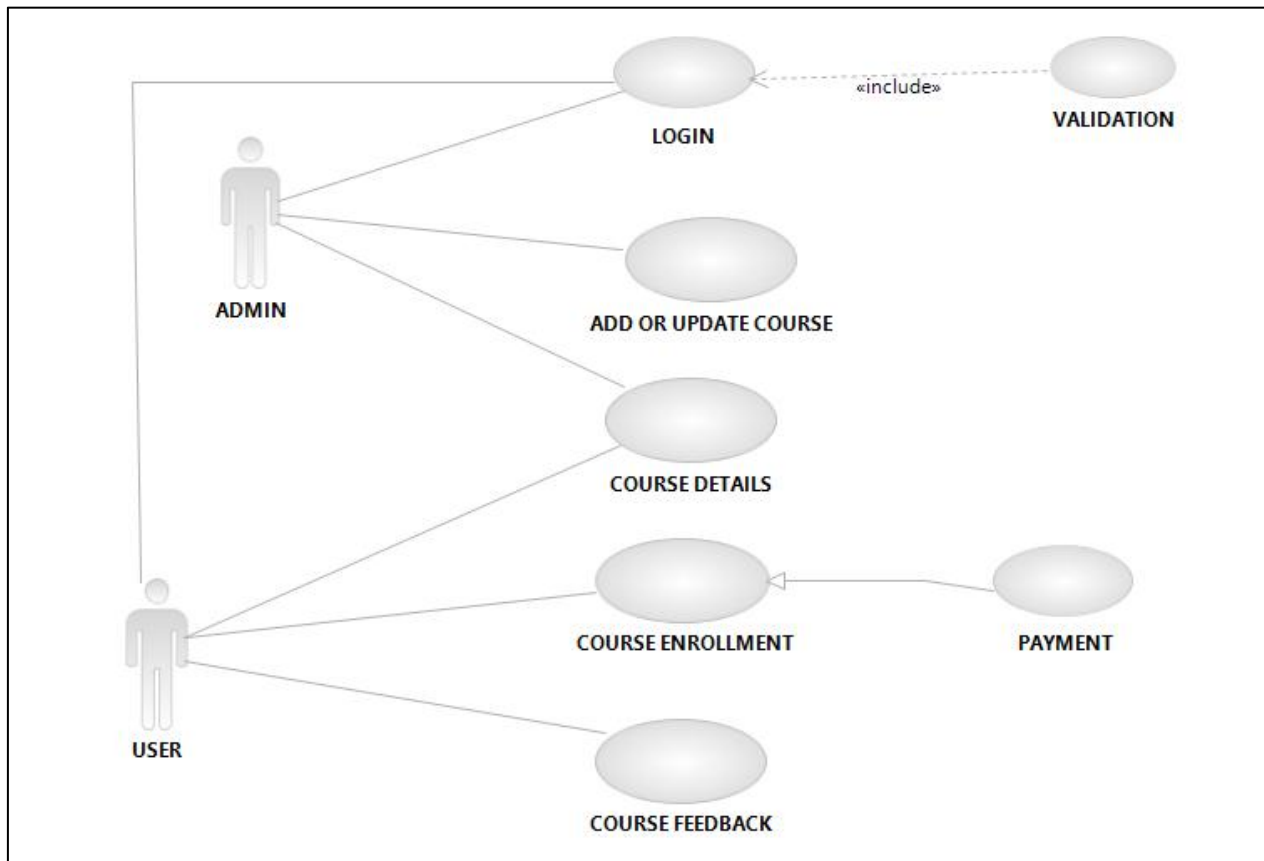
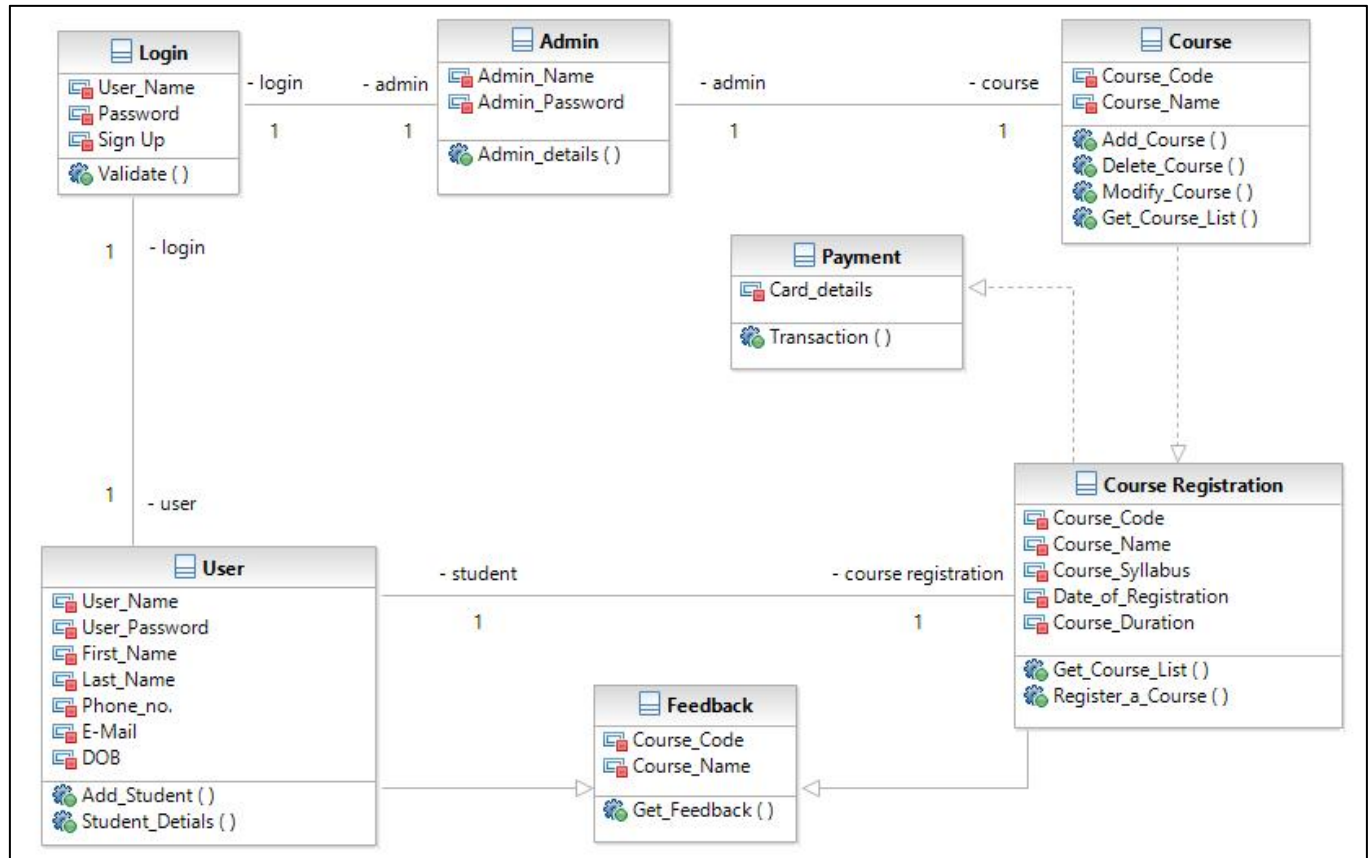


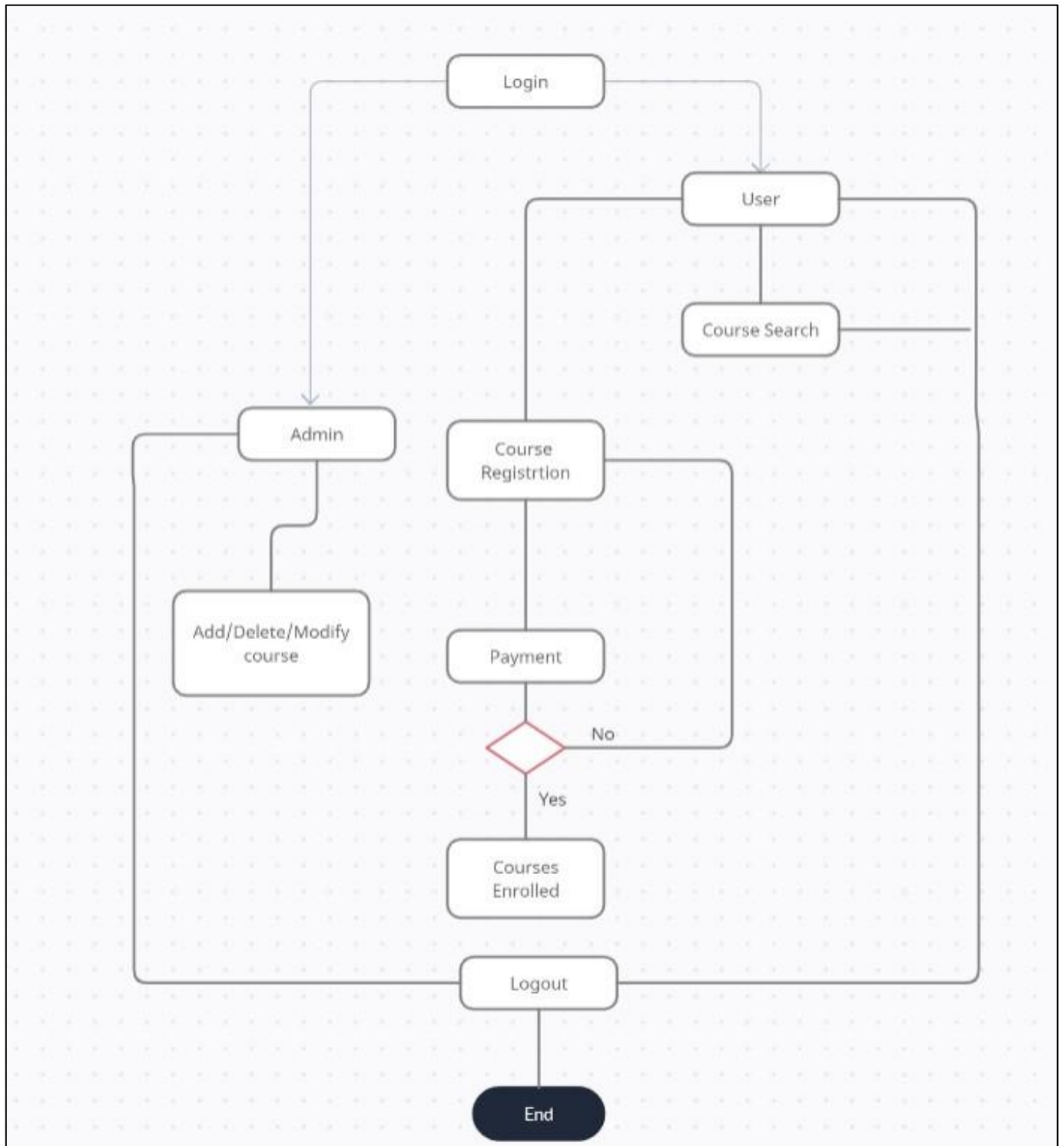
Usecase Diagram:



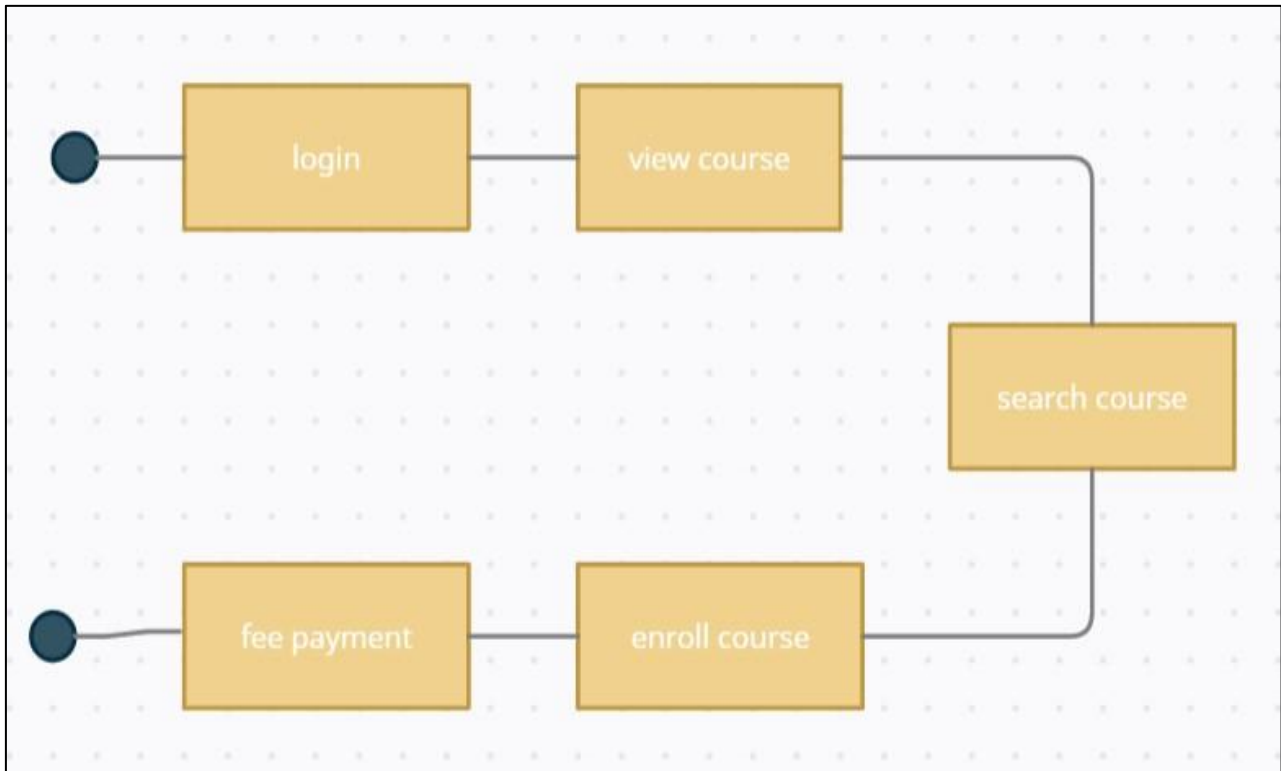
Class Diagram:



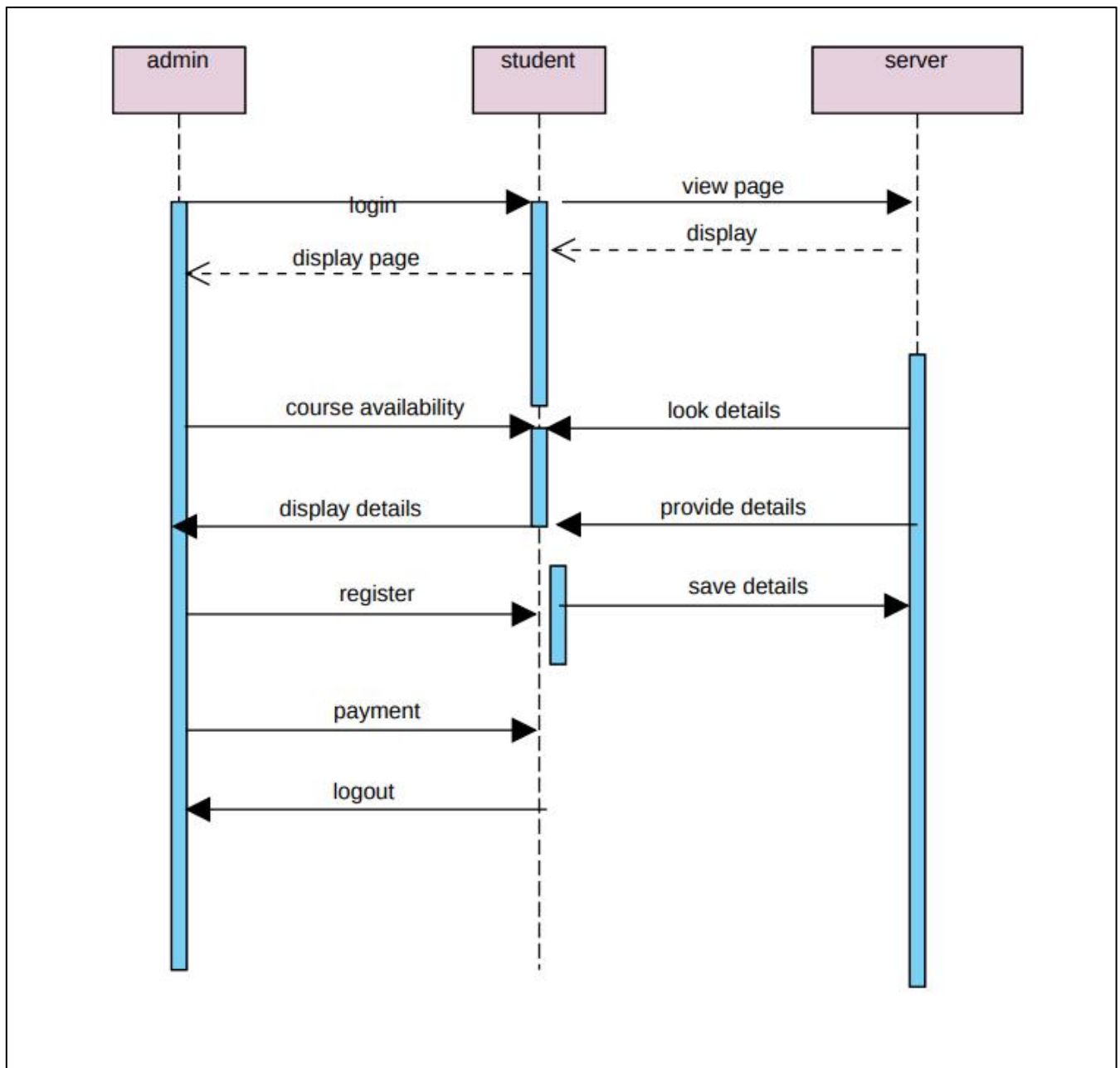
Activity Diagram:



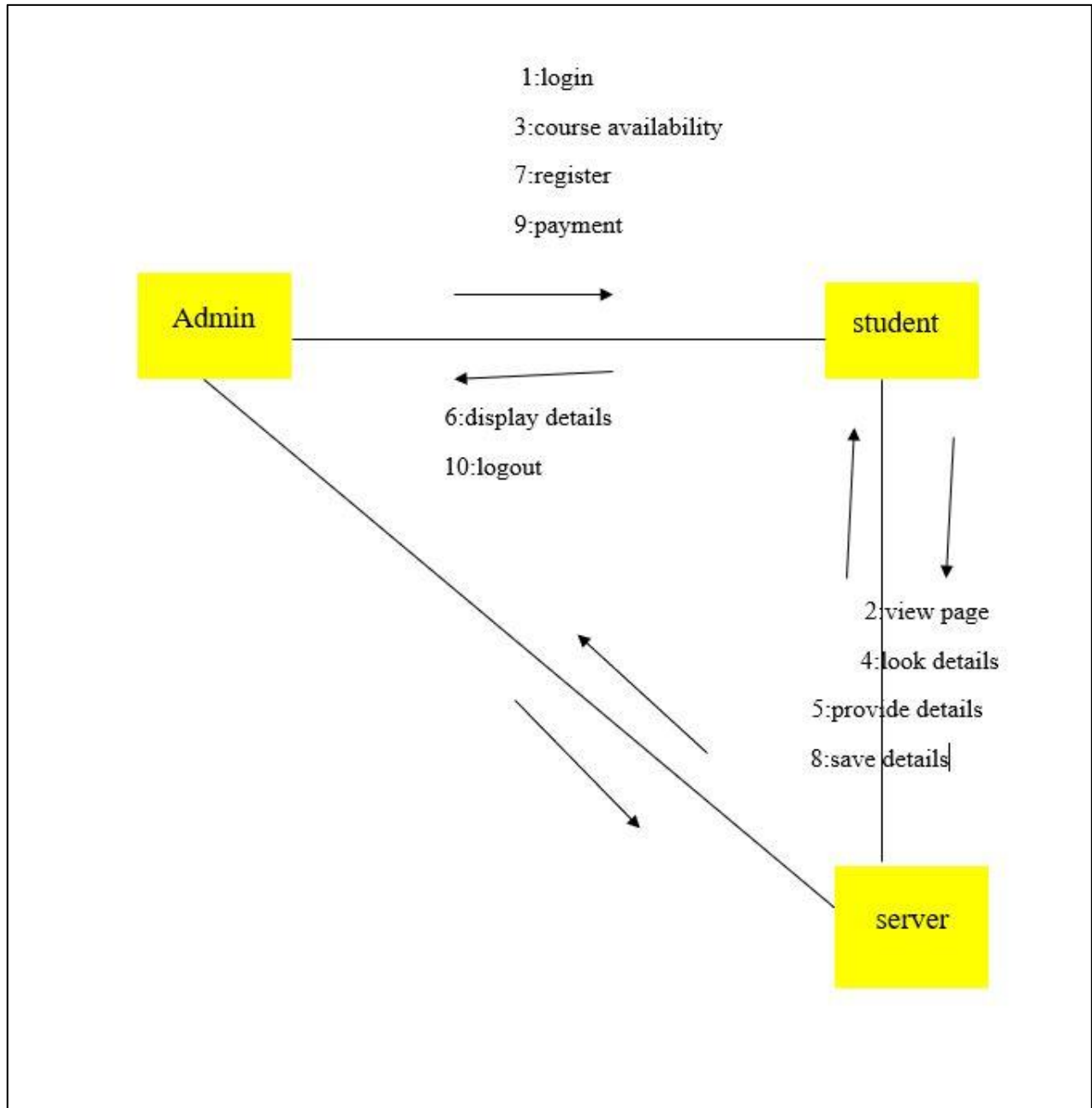
State Diagram:



Sequence Diagram:



Collaboration Diagram:



CODE:

App.js

```
import React,{ useEffect } from 'react'
import { BrowserRouter as Router } from 'react-router-dom';
import { getAllCourse } from './actions/course';
import { useDispatch } from 'react-redux';

import { setCurrentUser } from './actions/currentUser';
import AllRoutes from './AllRoutes';

import './App.scss';

function App() {
  const dispatch =useDispatch()
  useEffect( ()=>{
    dispatch(getAllCourse())
    dispatch(setCurrentUser(JSON.parse(localStorage.getItem('EduLearn'))))

  },[dispatch])

  return (
    <div className="App">
      <Router>
        <AllRoutes />
      </Router>
    </div>
  );
}

export default App;
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.scss';
import App from './App';
import { Provider } from 'react-redux';
import Reducers from './reducers';
import {createStore,applyMiddleware,compose} from 'redux'
import thunk from 'redux-thunk'
import reportWebVitals from './reportWebVitals';
```

```

const store=createStore(Reducers,compose(applyMiddleware(thunk)))

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <Provider store={store}>
    <App />
  </Provider>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();

```

Course.js

```

import * as api from '../api'
import { setMyDetails } from './myDetails'
export const createCourse = (courseData,navigate) => async(dispatch) =>{
  try{
    const {data}=await api.createCourse(courseData)
    navigate('/home')
    dispatch(getAllCourse())
  }catch(err){
    console.log(err.message)
  }
}

export const getAllCourse = () =>async(dispatch) =>{
  try {
    const {data} = await api.getAllCourse();
    dispatch({type:'FETCH_ALL_COURSE',payload:data})
  } catch (err) {
    console.log(err.message)
  }
}

export const enrollCourse = (id,course_id,navigate) => async(dispatch) =>{
  try {

    const {data} = await api.enrollCourse(id,course_id)
    navigate('/home')
    dispatch(setMyDetails())
  }
}

```



```

    } catch (err) {
        console.log(err.message)
    }
}

```

Auth.js

```

import * as api from '../api'
import { setCurrentUser } from './currentUser'
import { setMyDetails } from './myDetails'

export const signUp=(authData,navigate) => async(dispatch) =>{
    try {
        const {data}= await api.signUp(authData)
        dispatch({type:'AUTH',data})
        dispatch(setCurrentUser(JSON.parse(localStorage.getItem('EduLearn'))))
        navigate("/home")
    } catch (err) {
        alert(err.message);
    }
}

export const login= (authData,navigate) => async(dispatch) =>{
    try{
        const {data} = await api.logIn(authData)
        dispatch({type:'AUTH',data})
        dispatch(setCurrentUser(JSON.parse(localStorage.getItem('EduLearn'))))
        navigate("/home")
    }catch(err){
        alert(err.message)
    }
}

export const editProfile = (editData,navigate) =>async(dispatch) =>{
    try {
        const {data} =api.editProfile(editData)
        navigate('/home')
        alert("edit successfull")
        dispatch(setMyDetails())
    } catch (err) {
        console.log(err.message)
    }
}

```

```
export const setMentor = (mentorData,navigate) => async(dispatch) =>{
  try {
    const {data} = api.setMentor(mentorData)
    navigate("/home")
    alert("you are a mentor")
    dispatch(setMyDetails())
  } catch (err) {
    alert(err.message)
  }
}
```

Mydetails.js

```
import * as api from '../api'
```

```
export const defaultDetails = () => async(dispatch)=>{
  try {
    const user=JSON.parse(localStorage.getItem('EduLearn'))
    const result=user.result
    dispatch({type:'SET_DEFAULT_DETAILS',payload:result})
  } catch (err) {
    alert(err.message)
  }
}
```

```
export const setMyDetails = ()=> async(dispatch) =>{
  try {
    const user=JSON.parse(localStorage.getItem('EduLearn'))
    const id = user.result._id
    const {data}= await api.myDetails(id);
    console.log(data)
    dispatch({type:'SET_DETAILS',payload:data})
  } catch (err) {
    alert(err.message)
  }
}
```

Server

Client.js

```
const mongoose=require('mongoose')
const Course = require('../models/Course.js');
const User = require('../models/user.js');

module.exports.getAllCourse = async(req,res)=>{
  try{
    const course=await Course.find().populate('createdBy');
    res.status(200).json(course);
  }catch(err){
    console.log(err.message)
    res.status(500).send("Server error")
  }
}

module.exports.createCourse = async(req,res)=>{
  try{
    const course= new Course({...req.body});
    await course.save();
    res.status(200).json("success")
  }catch(err){

    console.log(err.message)
    res.status(500).json({msg:'internal server error'})
  }
}

module.exports.enrollCourse = async(req,res) =>{
  const {id,course_id} = req.body
  try {
    const user = await User.findById(id);
    user.enrolled.push(course_id)
    await user.save();
    res.status(200).json({msg:"Success"})
  } catch (err) {
    console.log(err.message)
    res.status(500).json({msg:'Internal server error'})
  }
}

module.exports.getEnrolledCourse = async(req,res) =>{
```

```

const {id}=req.params
  try{
    const user=await User.findById(id).populate('enrolled')
    res.status(200).json(user);
  }catch(err){
    console.log(err.message)
    res.status(500).json({msg:"Internal Server error"})
  }
}

```

User.js

```

const mongoose = require('mongoose')
const jwt= require('jsonwebtoken')
const bcrypt = require('bcryptjs')
const User= require('../models/user.js')

```

```

module.exports.signUp = async(req,res)=>{

```

```

  const {name,email,password} = req.body
  try {
    const existinguser=await User.findOne({email}).populate('enrolled')
    if(existinguser){
      return res.status(400).json({message:'User already found..'})
    }
    const hashPassword = await bcrypt.hash(password,12);
    const newUser=new User({name,email,password:hashPassword})
    await newUser.save();
    const token =
    jwt.sign({email:newUser.email,id:newUser._id},'token',{expiresIn:'1h'})
    res.status(200).json({result:newUser,token})
  } catch (err) {
    res.status(500).json('Something went worng...')
  }
}

```

```

module.exports.login = async(req,res) =>{

```

```

  const {email,password} = req.body;
  try{
    const existinguser = await User.findOne({email}).populate('enrolled')
    if(!existinguser){
      return res.status(404).json({message:"User not found..."})
    }
    const isPasswordCrt = await bcrypt.compare(password,existinguser.password)
    if(!isPasswordCrt){

```

```

        return res.status(400).json({message:"Invalid credentials"})
    }
    const token =
    jwt.sign({email:existinguser.email,id:existinguser._id},'token',{expiresIn:'48h'})
    res.status(200).json({result:existinguser,token})
  }catch(err){
    res.status(500).json(err.message)
  }
}

module.exports.editProfile = async(req,res) =>{
  const {name,phone,skills,id} = req.body;
  try {
    const user = await User.findByIdAndUpdate(id,{ $set :
{phone,skills,name}}, {upSert:true})
    await user.save();
    res.status(200).json({msg:"Success"})
  } catch (err) {
    res.status(500).json({msg:"Internal server error"})
  }
}

module.exports.setMentor = async(req,res)=>{
  const {phone,skills,experience,language,isMentor,id} = req.body
  console.log(phone,skills,experience,language,isMentor,id)
  try {
    const user = await
User.findByIdAndUpdate(id,{ $set:{phone,skills,experience,language,isMentor}}, {upsert:true}
)
    await user.save();
    res.status(200).json({msg:"successfull"})
  } catch (err) {
    res.status(500).json({msg:'Internal server error'})
  }
}

module.exports.myDetails = async(req,res) =>{
  const {id} = req.params
  try {
    const user =await User.findById(id).populate('enrolled')
    console.log(user)
    res.status(200).json(user)
  }
}

```

```
} catch (err) {  
  console.log(err.message)  
  res.status(500).json({msg:'User not Found'})  
}  
}
```