Software Requirements Specification

for

EduShare E-Learning Application

**Version 1.2**

**Prepared by Hriday Kaju 15IT212**

**27/02/2018**

**Table of Contents**

**1. Introduction 3**

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Product Scope

1.5 References

**2. Overall Description 4**

2.1 Product Perspective

2.2 Product Functions

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

**3. External Interface Requirements 6**

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

3.4 Communications Interfaces

**4. System Features 7**

4.1 Login/Register

4.2 Create Course

4.3 View Course

4.4 Search/Find Course

4.5 Remove Course(Instructor)

4.6 Delete Course(Student)

4.7 View/Edit Profile

**5. Other Nonfunctional Requirements 10**

5.1 Performance Requirements

5.2 Security Requirements

5.3 Software Quality Attributes

5.4 Business Rules

**Appendix A: Glossary 11**

**Appendix B: Analysis Models 12**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Version 1.1 | 02/27/18 | Added UML Diagrams and updated system features. | 1.1 |
| Version 1.2 | 3/10/18 | Updated System features and requirements | 1.2 |

# Introduction

## Purpose

The purpose of this section is to provide the reader a general background about the e-learning application software EduShare which provides online educational courses for students. The educational courses are created and published by instructors themselves in the same application.

## Document Conventions

The following conventions have been followed while writing this document:

Font: Times New Roman

-Size 18 for Main Headings

-Size 14 for Sub Headings

-Size 12 for the rest of the document. ·

## Intended Audience and Reading Suggestions

This document is meant for professors, students and developers who consider online courses and learning tools as a useful mean for education. This document aims at explaining in an easy manner, the basic idea behind EduShare. It aims to introduce to professors/instructors and students the main features of EduShare and how the developers aim to achieve their goals.

## Product Scope

This document is the Software Requirement Specification for EduShare , an online e-learning application, version 1.1. It describes the functions and performance requirements of EduShare. Using this software an instructor can create and offer courses, maintain assessments and edit courses. A student can subscribe to and take the online courses offered.

## References

-http://ieeexplore.ieee.org/document/720574/- IEEE Recommended Practice for Software Requirements Specifications

# Overall Description

## Product Perspective

The e-Learning tool should enable the instructors to develop rich multimedia presentation courses with presentations, videos, tutorials and tests which are easily available online for students. This software can be used for developing e-learning courses that can later be released on the Internet or delivered in some other electronic medium.

## Product Functions

Edushare has two main user types: instructors and students.

An instructor using EduShare can do the following:

* Create online courses to be offered to students using the application
* Design the courses using presentations, videos and text descriptions. The courses can also include assessment tools such as tests.
* Edit their profile information
* Instructors will have admin privileges and can modify/edit course information.

A student using EduShare can do the following :

* Search and explore the online courses offered among the multiple categories.
* Subscribe to and take a course offered in the application.
* Interact and use each part of a course.
* Taking multiple choice tests to evaluate his/her performance.
* Obtain a certificate from EduShare on completion of a course.
* Edit their profile information.
* View instructor information and contact info.

## User Classes and Characteristics

The two main user classes that will use the software are Instructors and Students.

Instructors essentially have admin rights and can create the courses to be offered to the students accessing the application.

Students will be able to view/find courses to opt for and can take/interact with the course contents in the application.

## Operating Environment

The software will be compatible with most web browsers such as Google Chrome, Firefox, Safari, Microsoft Edge,etc. Any operating system including Windows, Linux, Android and iOS is sufficient. It will be best for laptop/desktop use but will also be responsive and mobile friendly.

## Design and Implementation Constraints

The product should feature a simple and easy to use GUI since it should be easily accessible to users of all ages. The product does not require intensive hardware to be able to use. However the website must be optimized and should be able to load under 5 seconds on the majority of systems/devices. More design constraints may be specified in future versions of SRS.

## User Documentation

There is no specific documentation for the software, but tutorials and help will be provided in the application. The application will also be simple to use and should not require any pre-requisite knowledge or complex documentation.

## Assumptions and Dependencies

The software is not a medium to create content for their courses. The content such as videos, documents, or presentations must be created externally and they will be embedded to the website.

For videos, it is assumed that the instructor has their own videos that can be uploaded on youtube and embedded onto the webpage. Instructors can provide the link of their youtube video to be part of the course lesson.

# External Interface Requirements

## User Interfaces

There will be a common login interface for all users when they enter the application. Then it splits into two main user interfaces, one for instructors and one for students. In the instructor interface there will be a main homepage with a navigation menu with different views for the different functions under the instructor such as creating a course, editing a course, viewing courses, and viewing profile information.

In the student user interface, there will be a main homepage with a navigation menu to access the different views which each provide different functions such as viewing available courses, using currently subscribed courses, and editing profile information.

The homepage in each case will be a dashboard showing useful information. In the case of instructor, it will show the instructor’s created courses. In the case of a student, it will show the explore tab in which the student can add new courses. Clicking on a course in the dashboard will bring up options to other features.

## Hardware Interfaces

The application can be used on any hardware with Windows, Linux, Android or iOS.

## Software Interfaces

For the front end interface, the javascript framework AngularJS will be used. HTTP will be used as the application layer protocol for communication between the front-end and back-end when required in the application. The data returned to the front-end application will be JSON objects/arrays. Node.JS will be used to connect with the NoSQL database MongoDB.

## Communications Interfaces

The application will use HTTP requests to obtain data from the back-end server when required in the form of JSON data. The services to interact with the database will use Python web framework Flask.

# System Features

## Unique Login Page/Register

4.1.1 Description and Priority

When the application is opened, a login page is required for users to login to the application since they each will have unique data associated with them. If the user is not registered, a signup option should also be present in which the user enters his details. This feature is of high priority.

4.1.2 Stimulus/Response Sequences

Sequence of User Actions:

1. User enters his unique email/username and password to login to the application.
2. If successful, the user successfully enters their respective home page depending on if they are a instructor or student.
3. If unsuccessful or the email/username is not found in the database, the user is prompted to register.
4. ‘Register’ action will prompt the user to fill a form with their details with field validation. Upon submitting the form successfully, the user will now be able to login.
5. A user can also register a new account without attempting to login.

4.1.3 Functional Requirements:

REQ-1: Username field required for checking a valid username

REQ-2: Password field required for verifying the password associated with the unique username in the database.

REQ-3: Sign in button to send the data and check the database for verification of the entered details.

REQ-4: Register button required for new users to register to the application.

REQ-5: A register form/page which has fields in which the user enters their name, DOB, email, and whether he/she is signing up as an instructor or as a student.

## Create Course

4.2.1 Description and Priority

It is the feature which allows an instructor using the application to create a new online course. It is of high priority.

4.2.2 Stimulus/Response Sequences

Sequence of User Actions:

1. Enter fields with required data such as Course Title, course subject, difficulty, length, course description.
2. Add the content to the course. The content can either be a video or presentation/pdf.
3. The instructor can add a test to the course. They will add questions each with 4 multiple choice options in the form of text.
4. Instructor can edit course content and information.
5. Submit changes and publish the course.

4.2.3 Functional Requirements

REQ-1: Required fields for course information are title, subject, difficulty, length, description.

REQ-2: For video lessons, the user must provide a youtube link to their video and it will be embedded onto the webpage.

REQ-3: User must be able to upload pdf/ppts to be embedded onto the webpage.

REQ-4: A publish button is shown if the user is creating the course. Upon clicking this, the course should be published and added to the database. Then student users should be able to see the course available in the application. If the instructor is editing the course contents, then a ‘Save changes’ button is shown instead.

* 1. **View Course**

4.3.1 Description and Priority

It is the feature which allows a student user using the application to view and interact with an available course. It is of high priority.

4.3.2 Stimulus/Response Sequences

Sequence of User Actions:

1. User adds a course from the list of available courses.
2. User then selects to view the course from the list of his/her subscribed courses.
3. User can view course description, details and instructor information.
4. User can view and interact with the course(video, presentation).

4.3.3 Functional Requirements

REQ-1: User must be able to see all parts in a course and easily navigate through the course to their liking.

REQ-2: All presentations/embedded videos in the lessons should each have a fixed size or division associated with them in the webpage to have a consistent feel to the application.

* 1. **Search/Find Courses**

4.4.1 Description and Priority

A user should easily be able to find courses which interest him. There will be several filters the user can use to find courses such as category/subject, length, difficulty, instructor, etc. User can also search by name. It is a feature of high priority.

4.4.2 Stimulus/Response Sequences

Sequence of User Actions:

1. User can select filters/categories among the available ones.
2. User enters text in the search bar and presses enter.
3. A list of results with courses will be shown to users.
4. User can select a course from the list to view its details and then add it if they wish.

4.4.3 Functional Requirements

REQ-1: Filters can be of checkbox type, or selectable dropdown. Filters include the subject category(list of all subjects in the database),difficulty, instructor. More filters may be specified in the later versions of SRS.

REQ-2: Number of results obtained should be displayed. In case no results are found, user should be prompted to change filters/search terms and search again.

REQ-3: Upon adding a course, the course should then be added to the users list of subscribed courses.

* 1. **Remove Course(Instructor)**

4.5.1 Description and Priority

An instructor should be able to remove/delete a course which they have created. It is a feature of high priority.

4.5.2 Stimulus/Response Sequences

Sequence of User Actions:

1. Instructor selects a delete option for the specific course they want to remove.
2. Instructor is asked to confirm
3. If any students are currently subscribed, the course will be removed from their list of subscribed courses.
4. If instructor confirms their choice, the course is removed from the database.

4.5.3 Functional Requirements

REQ-1: Instructor should be shown a confirmation dialog when they select ‘delete’ option.

REQ-2: If any students are subscribed to the course, they should also be deleted from the database.

* 1. **Delete Course(Student)**

4.6.1 Description and Priority

A student should be able to remove/delete a course which they have added. It is a feature of high priority.

4.6.2 Stimulus/Response Sequences

Sequence of User Actions:

1. Student selects a delete option for the specific course they want to remove.
2. Student is asked to confirm.
3. Course is removed from Student’s “My Courses” list.

4.6.3 Functional Requirements

REQ-1: Student should be shown a confirmation dialog when they select ‘delete’ option.

* 1. **Edit Profile Information**

4.7.1 Description and Priority

A user should be able to view and edit their profile information.

4.7.2 Stimulus/Response Sequences

Sequence of User Actions:

1. User selects “View Profile” tab.
2. User selects ‘Edit’ option.
3. User can modify profile options and save changes.

4.7.3 Functional Requirements

REQ-1: Both student and instructor can modify their profile information.

REQ-2: The fields in the Profile page are the same as the fields entered from the user while registering.

# Other Nonfunctional Requirements

## Performance Requirements

There are performance requirements for the product under various circumstances. The home page should load under five seconds for user convenience and retainability. There should be no hiccups or glitches while integrating and embedding videos, pdfs or presentations in the lessons of the application. It should have a consistent look.

## Security Requirements

To ensure secure logins, the users’ hash of the password will be compared to a hash stored in the database. A standard encryption algorithm md5 will be used.

## Software Quality Attributes

There are few quality attributes which the product will have to both customer as well as developer are as follows.

1. Adaptability - The product should be flexible and be able to adapt to and seamlessly integrate future requirements and changes.
2. Interoperability - It is the quality of the software to operate on multiple platforms as much as possible be consistent in every system.
3. Portability - The product is portable since it is an online service which might be operated anywhere and on any device.
4. Reliability - The product is reliable since it provides instant processing of the features explained as in system features and it has good response time.
5. Robustness - It should be built to work under all conditions. Since the product has not yet reached its implementation stage, it will be updated in later versions of SRS.
   1. **Business Rules**

These are the operating principles that the product will follow.

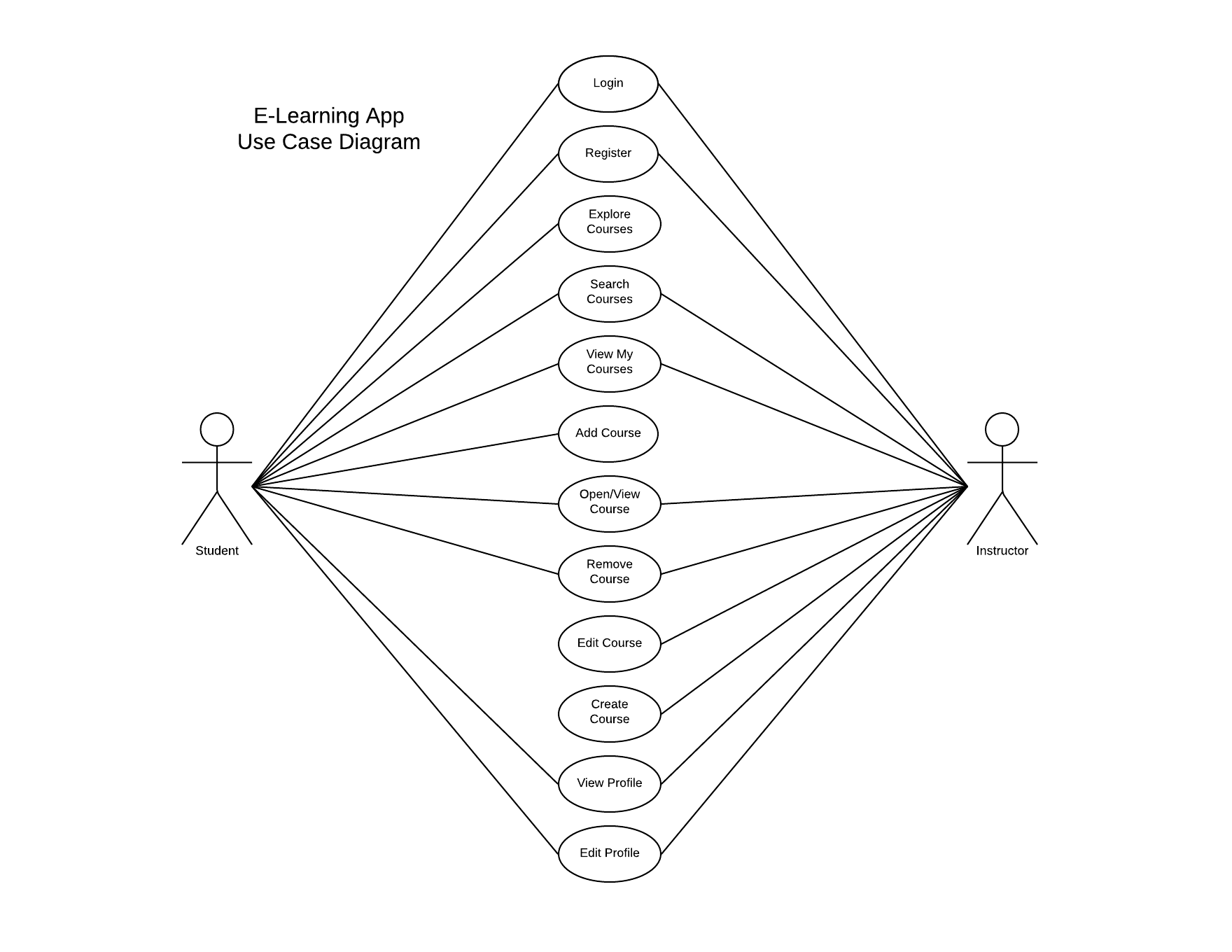
1. Make the product to have high rankings on search engines to attract more users and develop a diverse and large user base.
2. Build links with other similar companies which are hosting similar e-learning software.
3. Keep on updating the product according to user feedback and latest trends.
4. Use analytics for analysis of our product.

**Appendix A: Glossary**

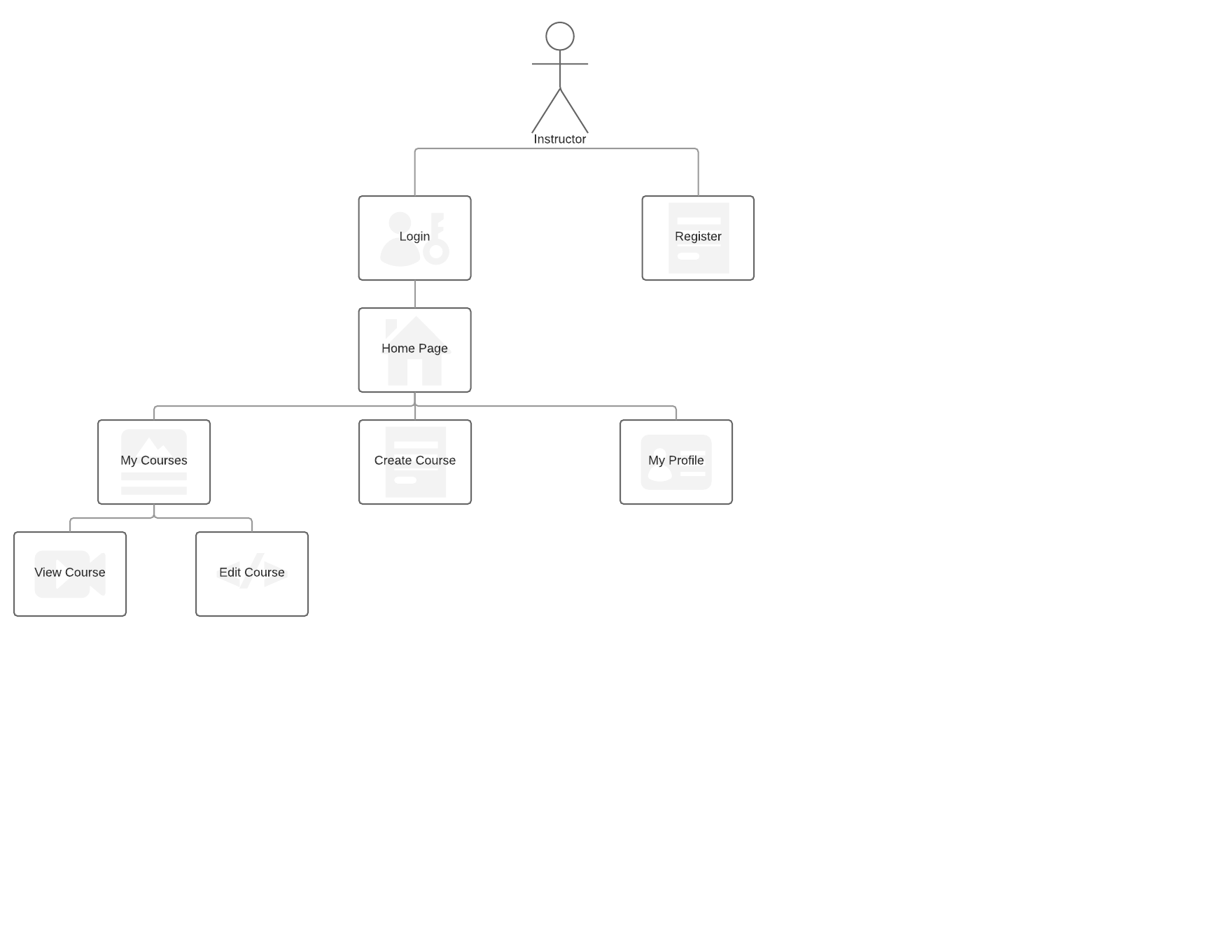
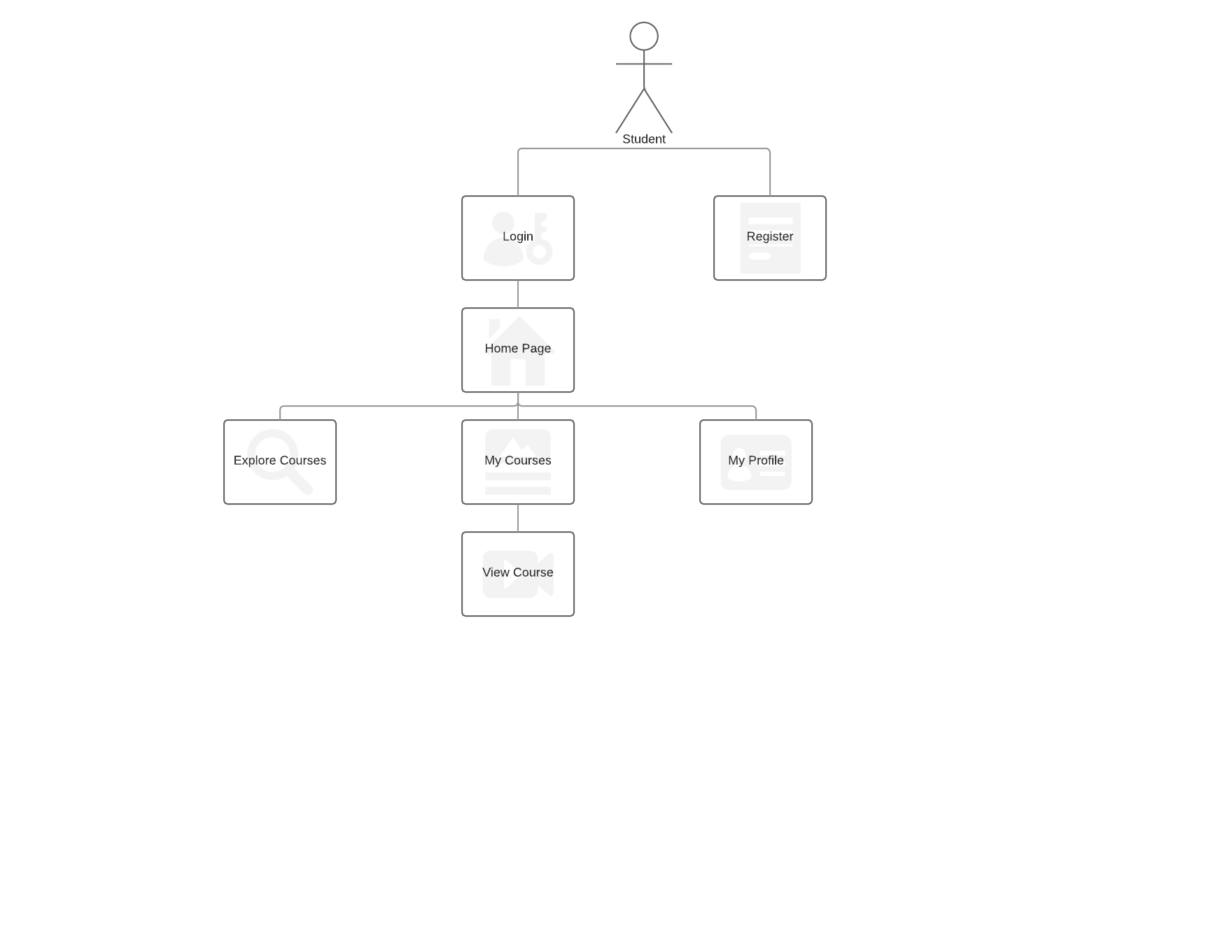
1. SRS - Software Requirement Specification.
2. HTTP - Hyper Text Transfer Protocol.
3. GUI - Graphical User Interface.
4. JSON- JavaScript Object Notation

**Appendix B: Analysis Models**

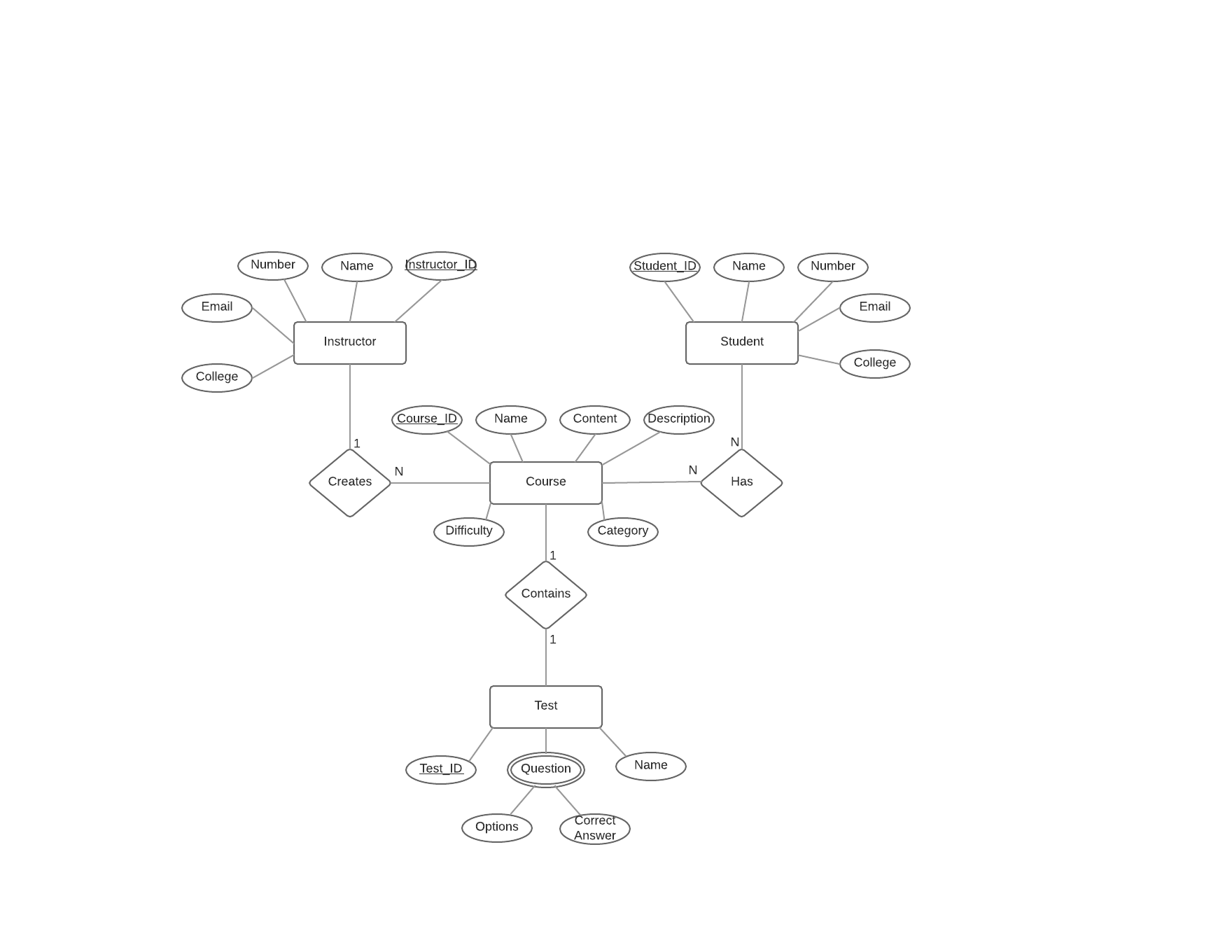
1. Use Case Diagram



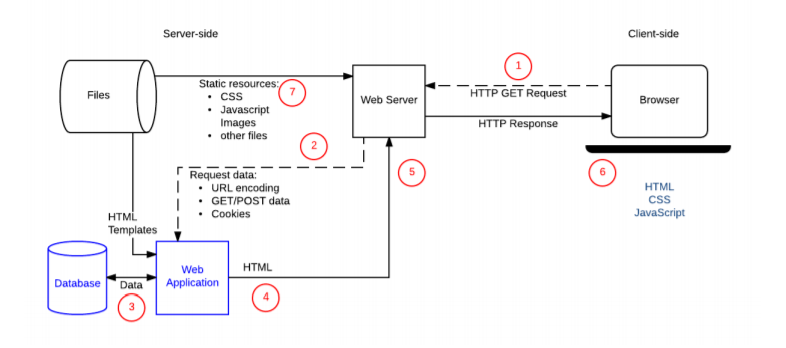
2. Navigation Diagrams

****

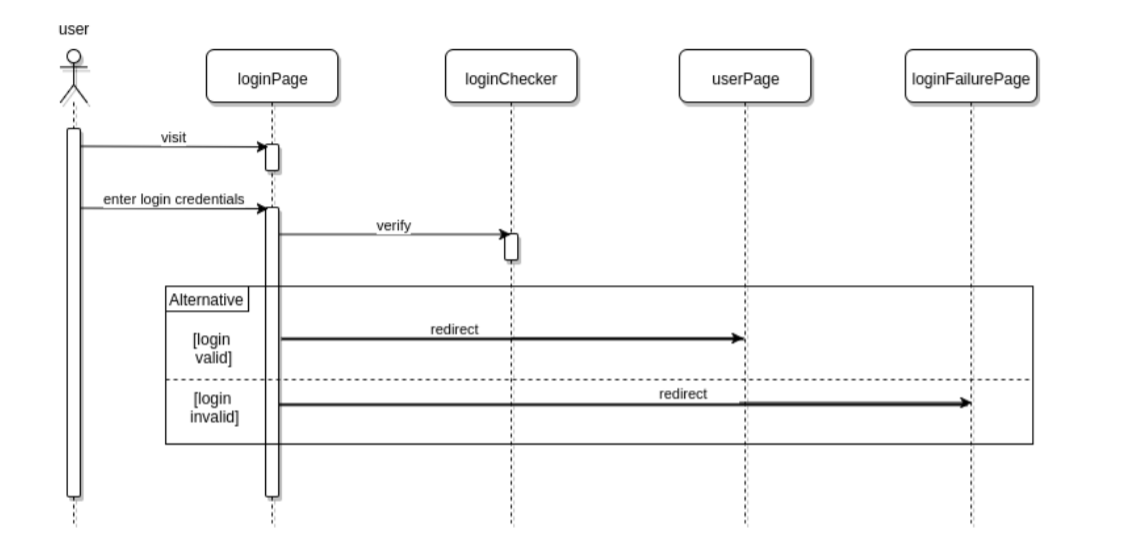
3. Entity-Relationship Diagram



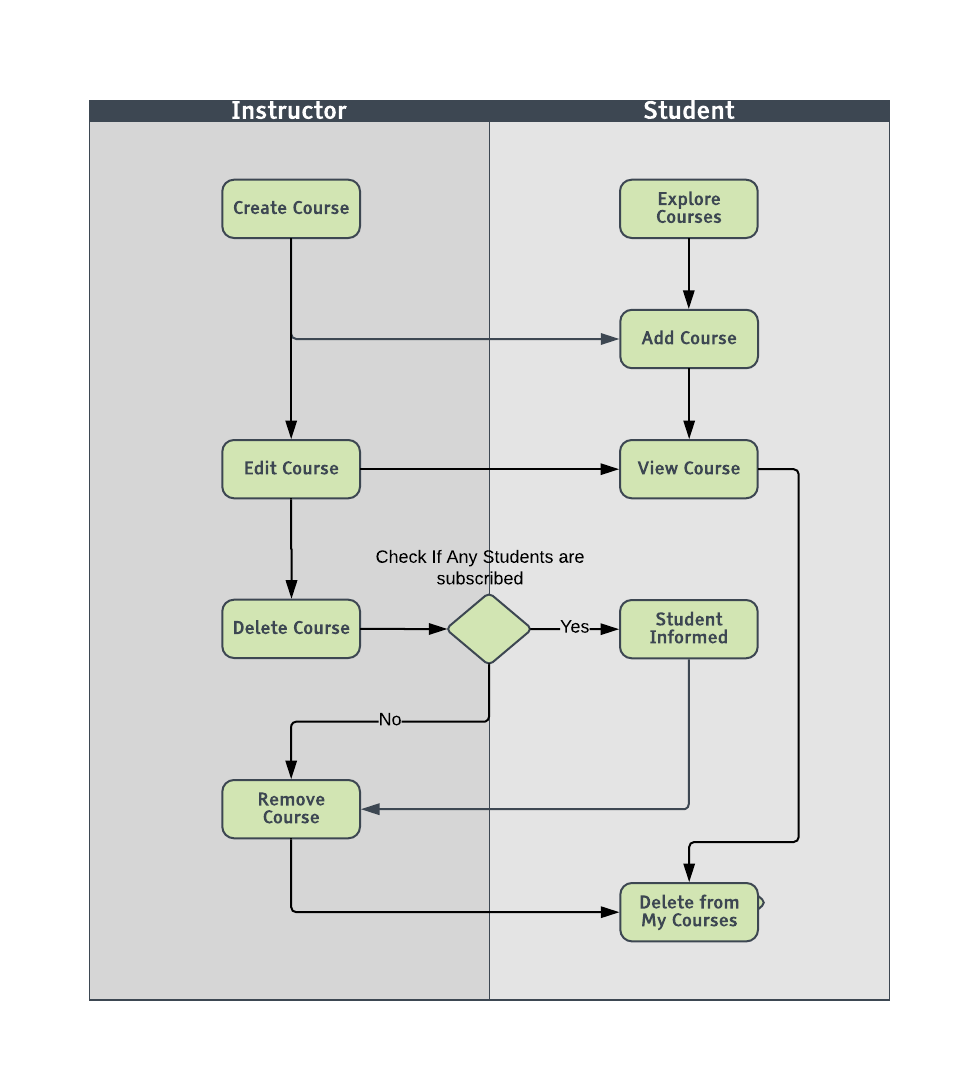
4. Architecture Diagram



5. Sequence Diagram for Login



6. Swim Flow Diagram



7. Activity Diagrams

