EECS 3311 Project Iteration 1

Group 5 (StudyLink)

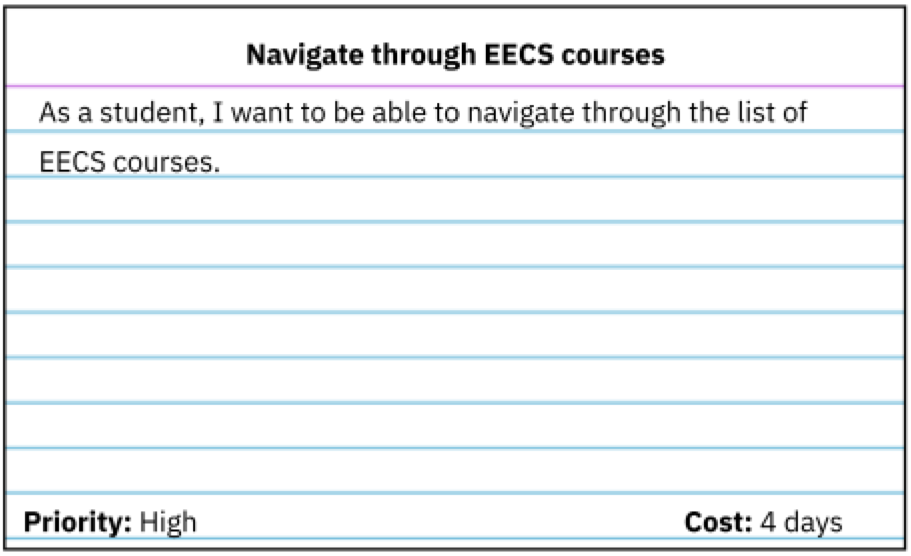
Team Members

Manasvi Jain 218218321
Angela Manalo 215726771
Yuvtesh Mann 217614736
Kevin Chang 217720939
Ammiel Cruz 211077971

Github link: https://github.com/chang-kevin/EECS_3311

Big Stories

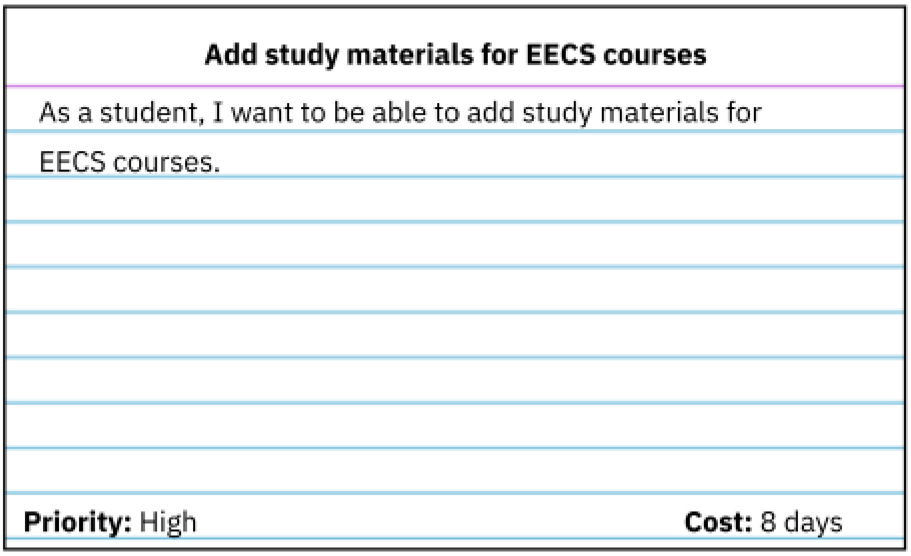
Iteration 1



Navigate through EECS courses

As a student, I want to be able to navigate through the list of EECS courses.

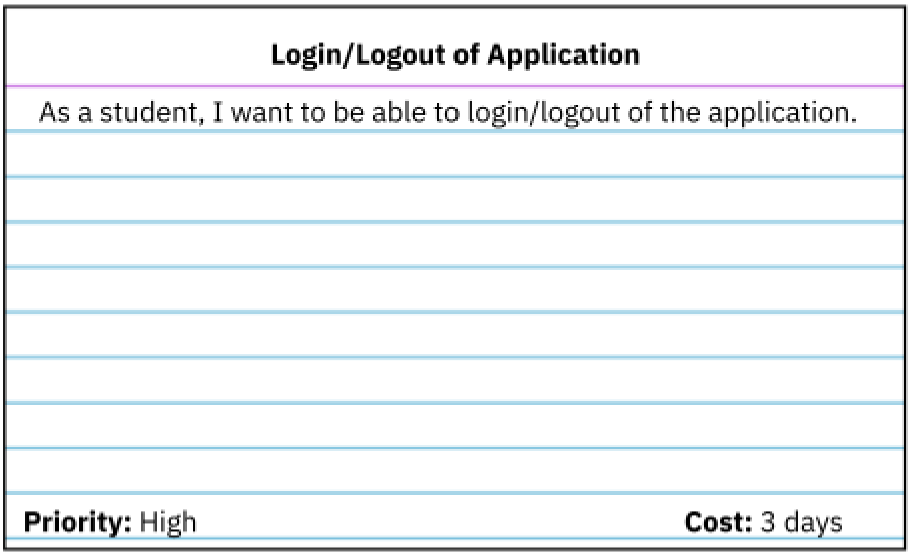
Priority: High **Cost:** 4 days



Add study materials for EECS courses

As a student, I want to be able to add study materials for EECS courses.

Priority: High **Cost:** 8 days

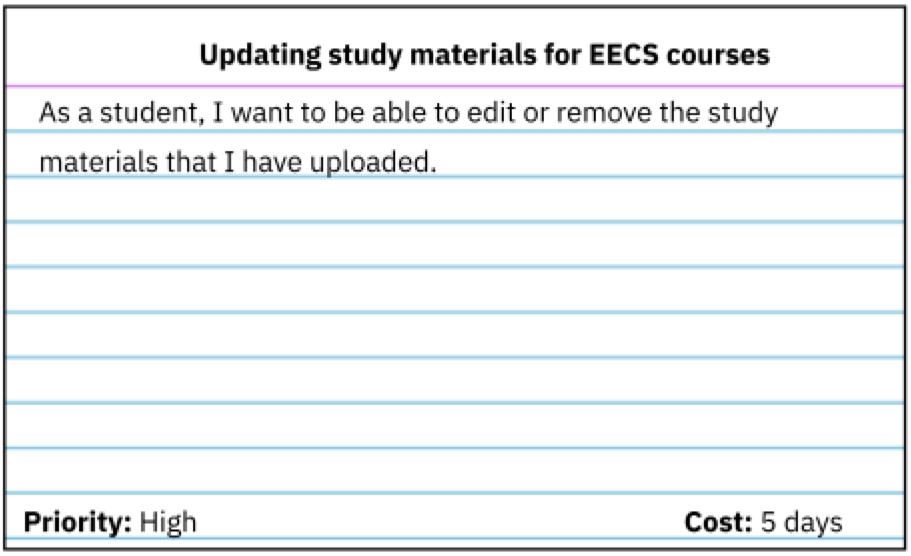


Login/Logout of Application

As a student, I want to be able to login/logout of the application.

Priority: High **Cost:** 3 days

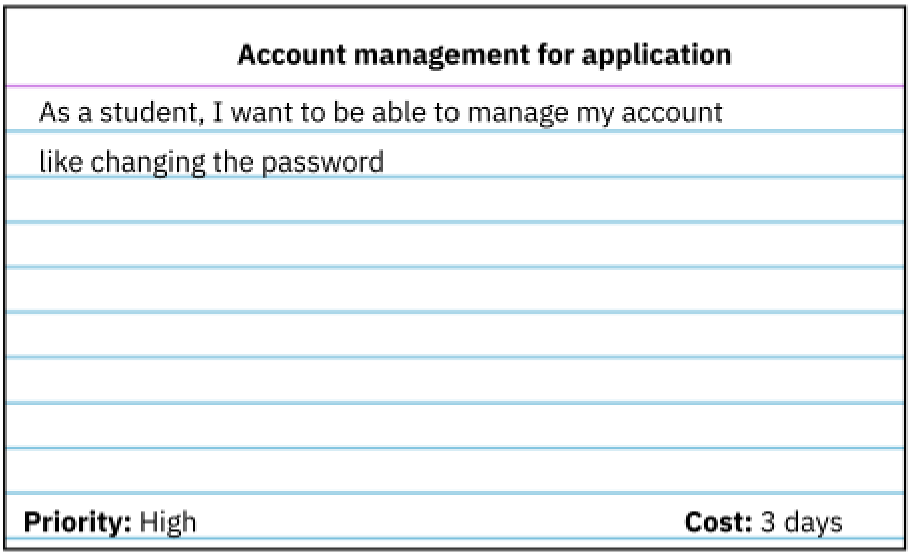
Iteration 2



Updating study materials for EECS courses

As a student, I want to be able to edit or remove the study materials that I have uploaded.

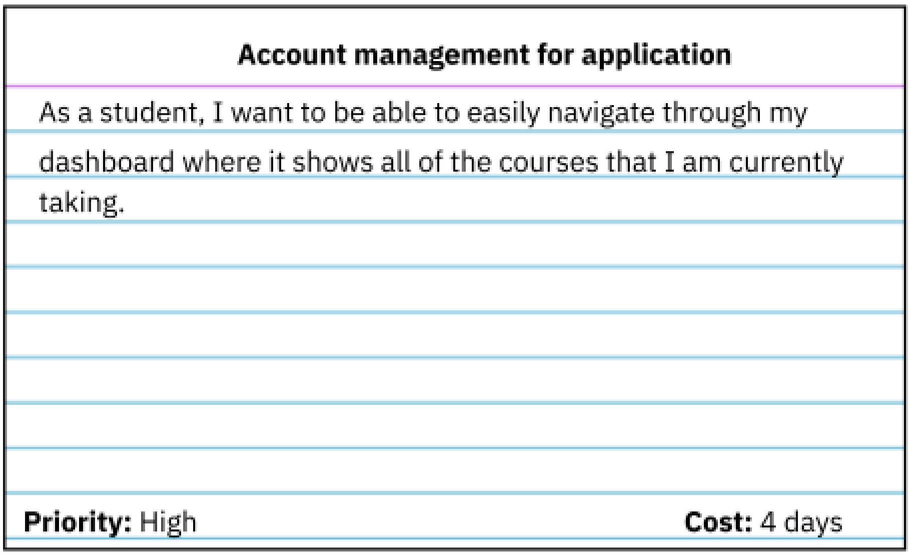
Priority: High **Cost:** 5 days



Account management for application

As a student, I want to be able to manage my account like changing the password

Priority: High **Cost:** 3 days



Account management for application

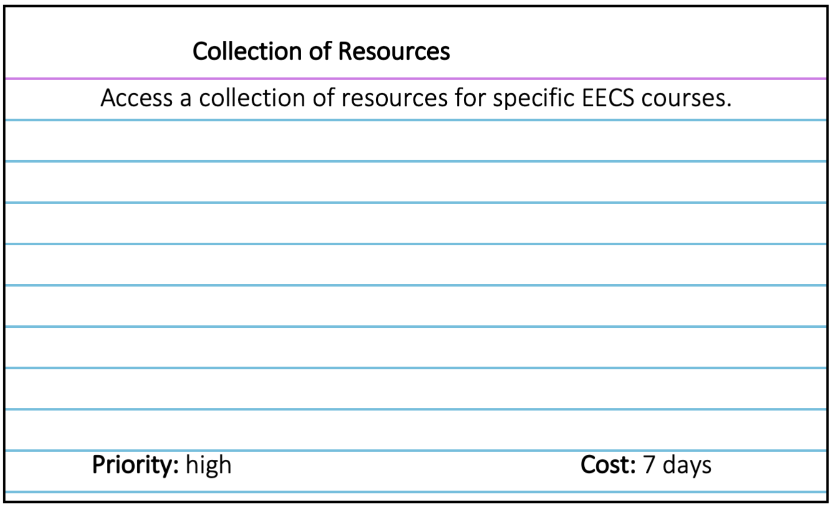
As a student, I want to be able to easily navigate through my dashboard where it shows all of the courses that I am currently taking.

Priority: High **Cost:** 4 days

The rating/reporting system was not implemented for the study materials in iteration 1. Iteration 1 was about releasing the product as early as possible with the most marketable features to get maximum revenue. The rating/reporting system is not our most marketable feature so if we had spent time to build that feature it would delay the release of our product. Also due to the time constraint, we had to prioritize and complete higher priority features. We plan to implement this feature in iteration 2.

Updated Plan For Iteration 1

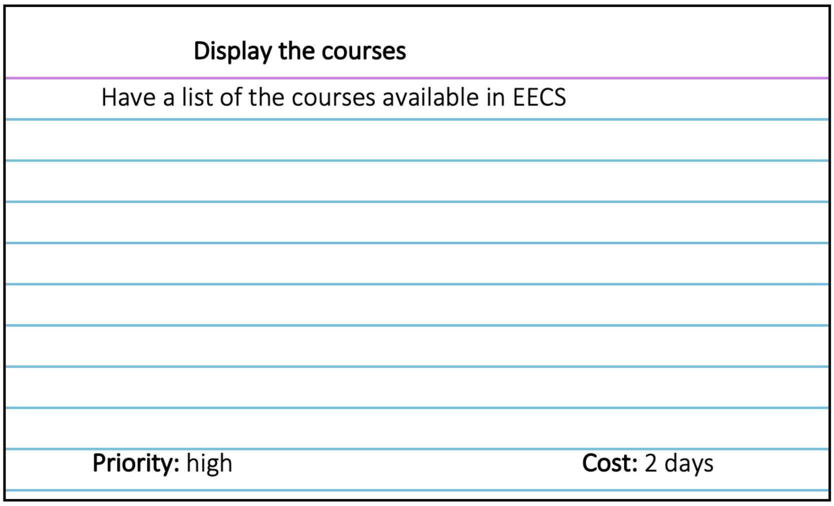
Navigating through resources



Collection of Resources

Access a collection of resources for specific EECS courses.

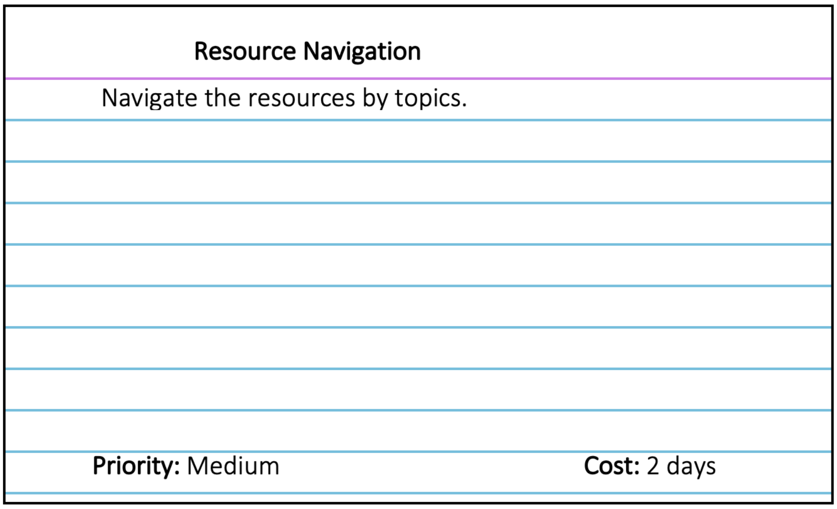
Priority: high **Cost:** 7 days



Display the courses

Have a list of the courses available in EECS

Priority: high **Cost:** 2 days



Resource Navigation

Navigate the resources by topics.

Priority: Medium **Cost:** 2 days

Adding study materials

Upload study materials to the app

Upload study materials for EECS courses.

Rating the study materials

Rate the study material from a scale of 1 – 5 based on how helpful it is to me

Priority: Medium **Cost:** 3 days

Reporting the study materials

Report a study material if it violates the terms and conditions.

Priority: Medium **Cost:** 3 days

Login/logout

Sign Up

Create an account with email and password

Priority: Medium **Cost:** 3 days

Login

Login into the account using email and password

Priority: Medium **Cost:** 3 days

Forgot/change password

Change the password using email

Priority: Medium **Cost:** 3 days

Development tasks assigned in iteration 1

Manasvi:

- Sign Up
- AddFile

Angela:

- ThirdYearCourses
- Dashboard

Yuvtesh:

- Userlist
- CourseList
- User
- Course

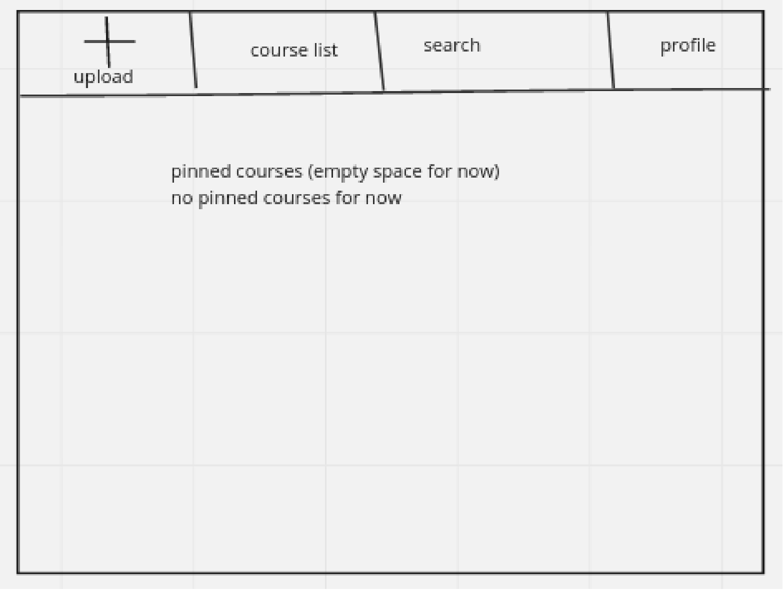
Ammiel:

- Login
- ForgotPassword

Kevin

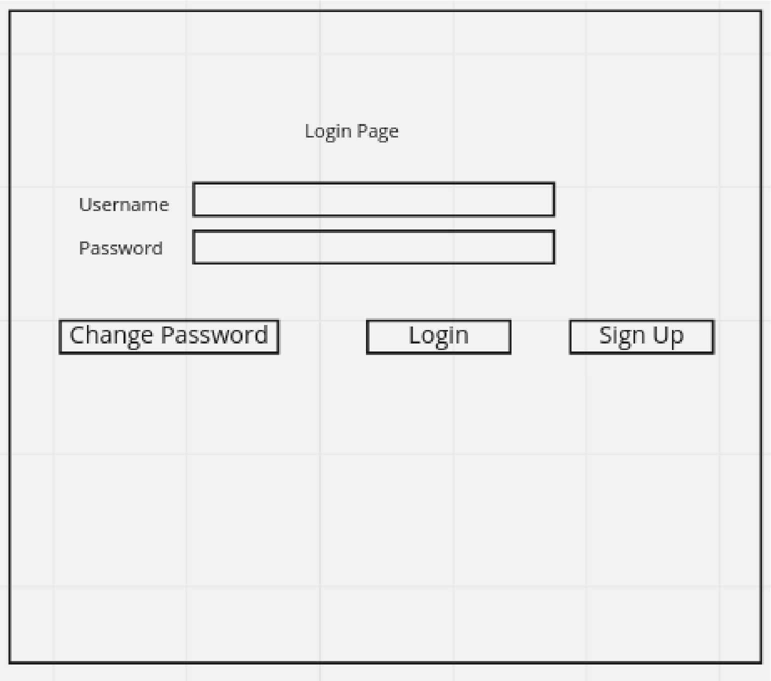
- Dashboard
- CourseResources

Dashboard skeleton picture as discussed by the group

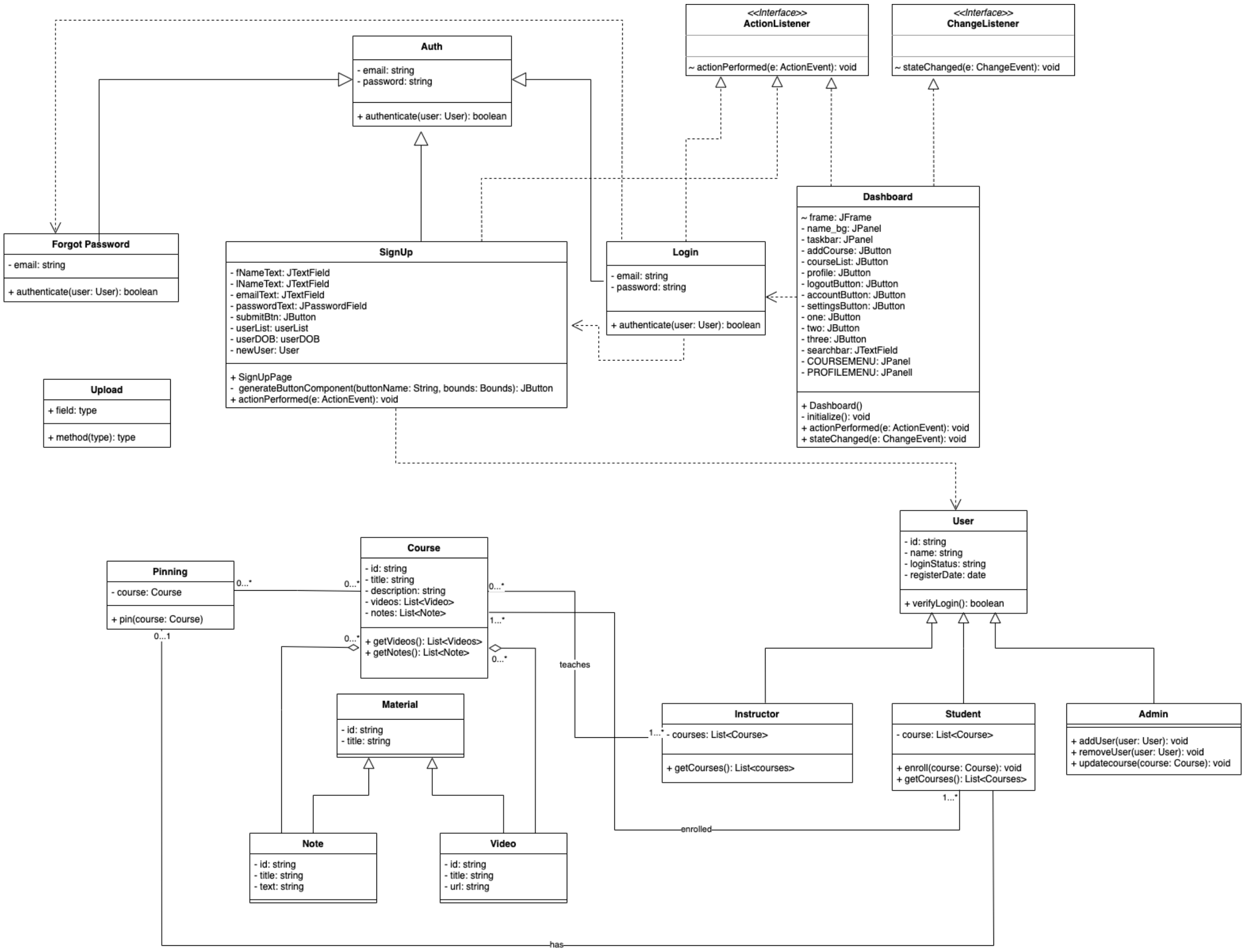


pinned courses (empty space for now)
no pinned courses for now

Login Page



Class Diagram



The User class for example represents a user with a username and password. It has getter methods to retrieve the username and password. The Login class takes a User object as input and performs the login action. It is also capable of redirecting the user to the forgot password page, and signup page. This example separates the data (User class) from the actions (Login/Logout/Signup) and follows the single responsibility principle where each class has a single, well-defined responsibility. In addition, it promotes loose coupling between classes, allowing for easier maintenance and additions in the future.

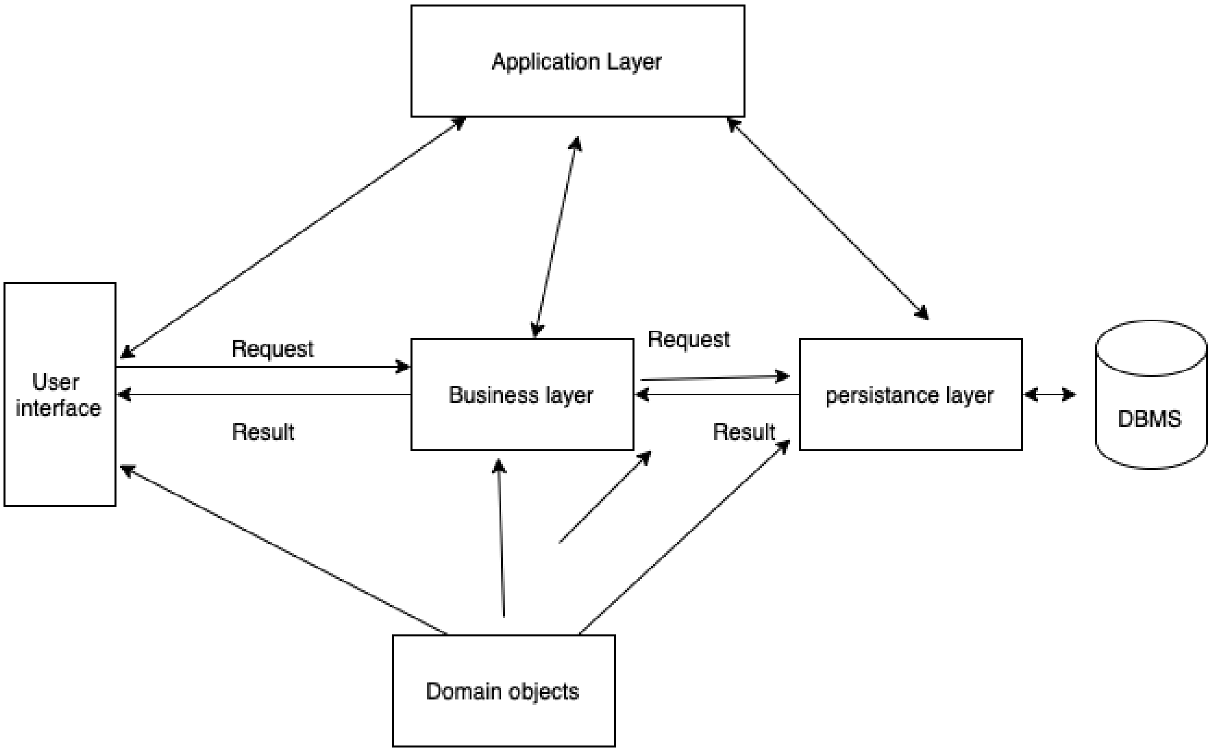
Platform Architecture

In this diagram, the arrows represent the flow of data and requests between the components of the platform.:

- The user is able to access the platform through the front-end/UI.
- The front-end then sends requests to the backend for data, such as course materials or user information.
- The backend retrieves the data from the database through the persistence layer, and returns it to the front-end.
- The front-end may also request data from a content delivery network (CDN) to reduce the load on the backend and improve load times for the user.
- The backend handles the user authentication through the authentication component

The Business Layer is responsible for implementing the business logic of the application. It sits between the frontend and backend where all the complex calculations, data processing and other business-related tasks are performed. The classes AddFile and ThirdYearCourses which are responsible for collecting the study materials organised for the students to conveniently have an archive for specific topics.

The Domain objects represent the entities in the platform, such as a User, or a Course. These objects wrap around the data and behaviour of the entities and provide a way to interact with the data.



The Persistence layer is responsible for storing and retrieving data from the database by providing an interface to the Business Layer for accessing and manipulating data. In our platform, this maps to the UserList and CourseList classes where UI classes are able to retrieve information via these classes.