EECS 3311 Project Iteration 3

Group 5 (StudyLink)

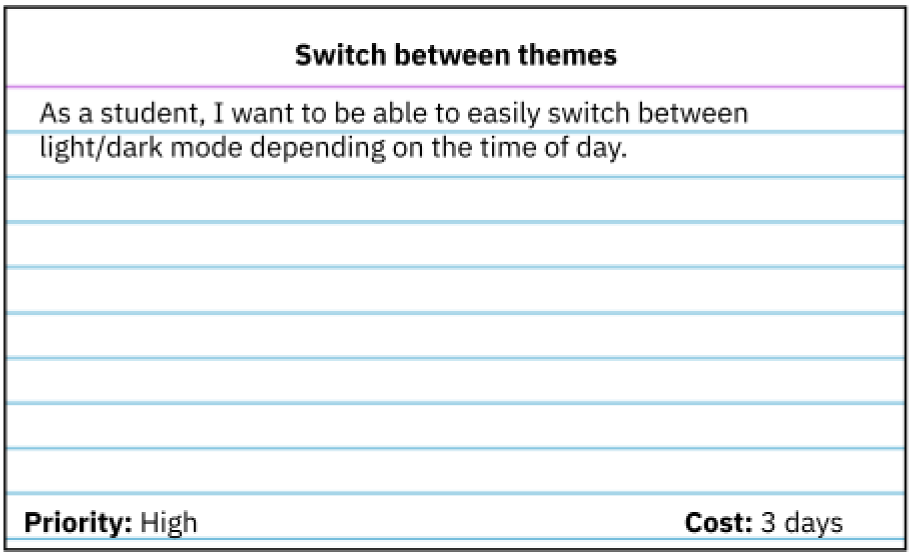
Team Members

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

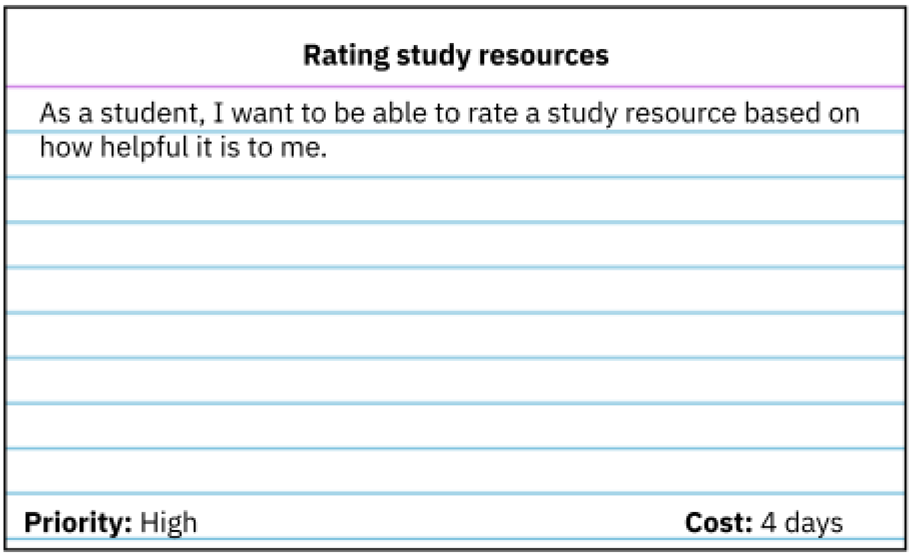
Iteration 3



Switch between themes

As a student, I want to be able to easily switch between light/dark mode depending on the time of day.

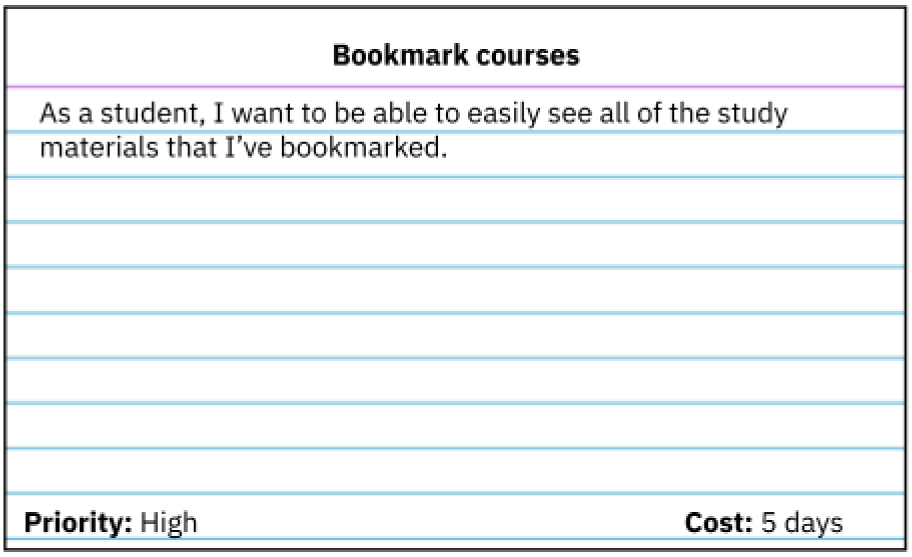
Priority: High **Cost:** 3 days



Rating study resources

As a student, I want to be able to rate a study resource based on how helpful it is to me.

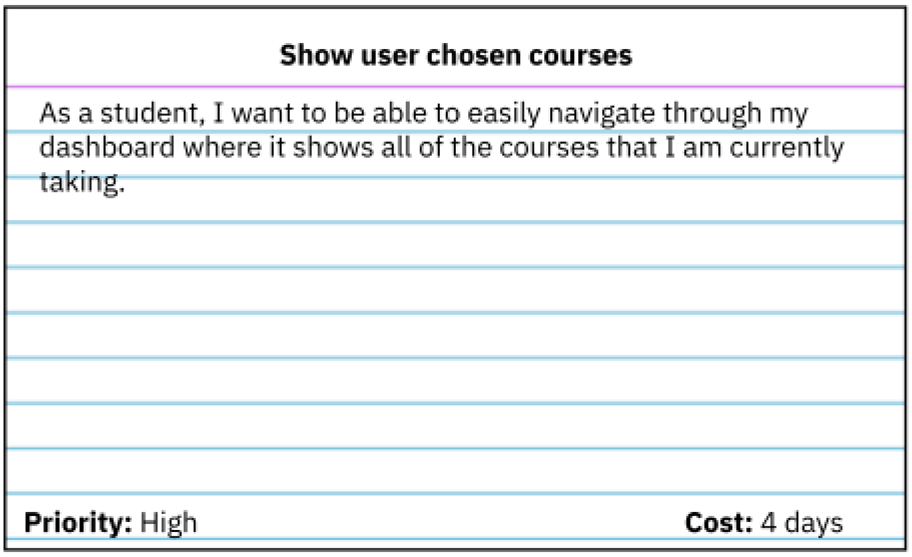
Priority: High **Cost:** 4 days



Bookmark courses

As a student, I want to be able to easily see all of the study materials that I’ve bookmarked.

Priority: High **Cost:** 5 days



Show user chosen courses

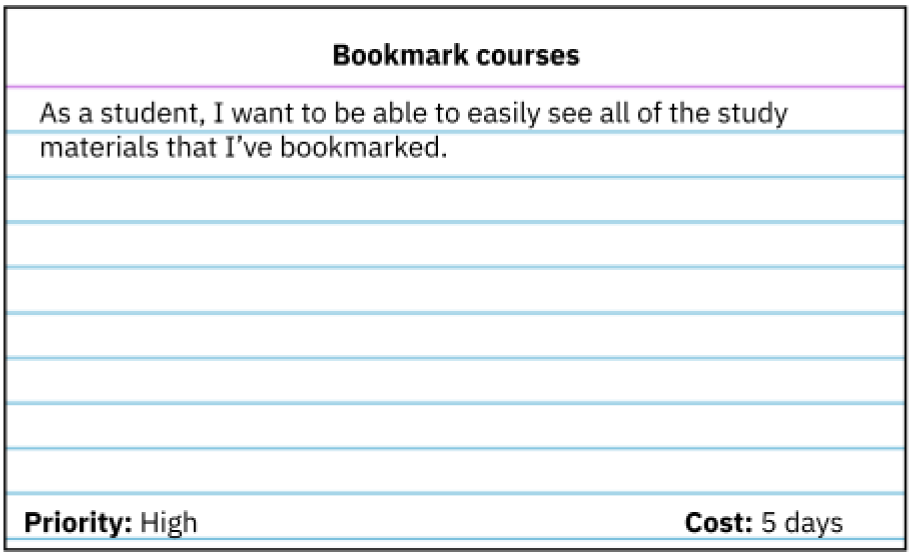
As a student, I want to be able to easily navigate through my dashboard where it shows all of the courses that I am currently taking.

Priority: High **Cost:** 4 days

The switch between themes and rating study resources were not implemented for the in iteration 3. The switching betweens themes did not add anything to the overall purpose of our system so we decided not to implement it, instead we updated the UI which displays properties of light and dark themes as a default theme. The rating of study resources was not implemented in Iteration 3 because rating study materials turned out to be insignificant in terms of our overall purpose of the system.

Updated Plan For Iteration 3

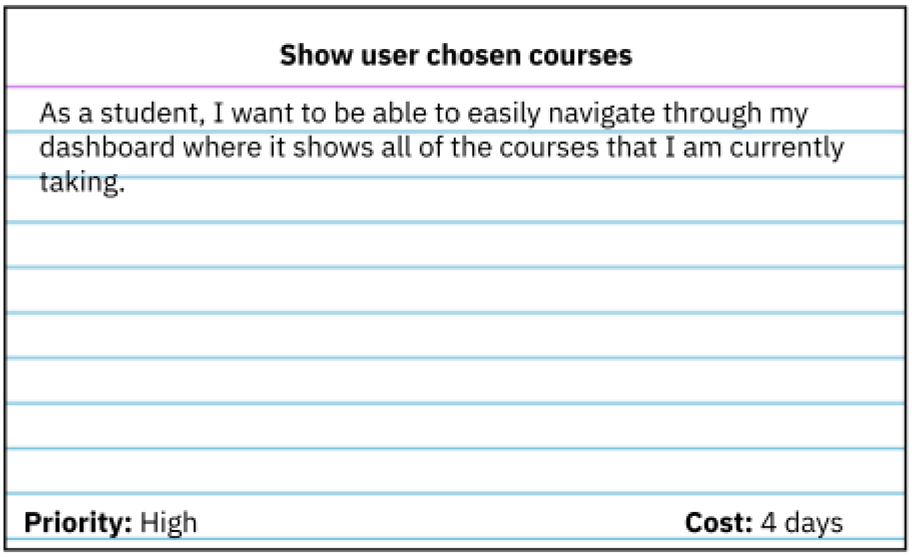
Big user stories



Bookmark courses

As a student, I want to be able to easily see all of the study materials that I've bookmarked.

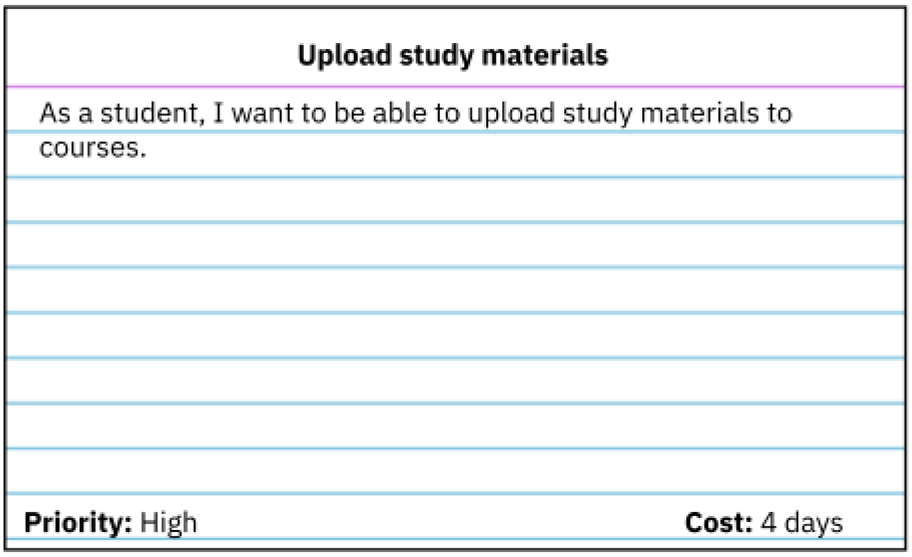
Priority: High **Cost:** 5 days



Show user chosen courses

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

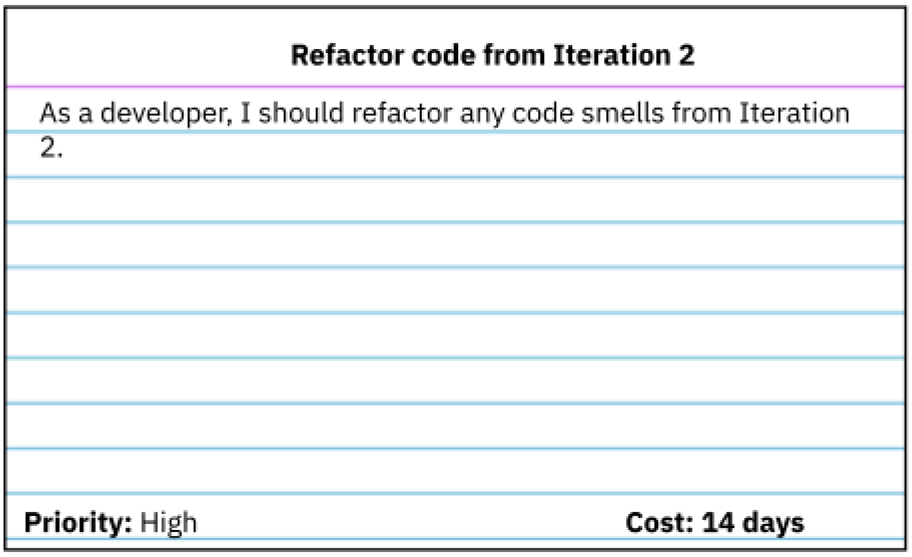


Upload study materials

As a student, I want to be able to upload study materials to courses.

Priority: High **Cost:** 4 days

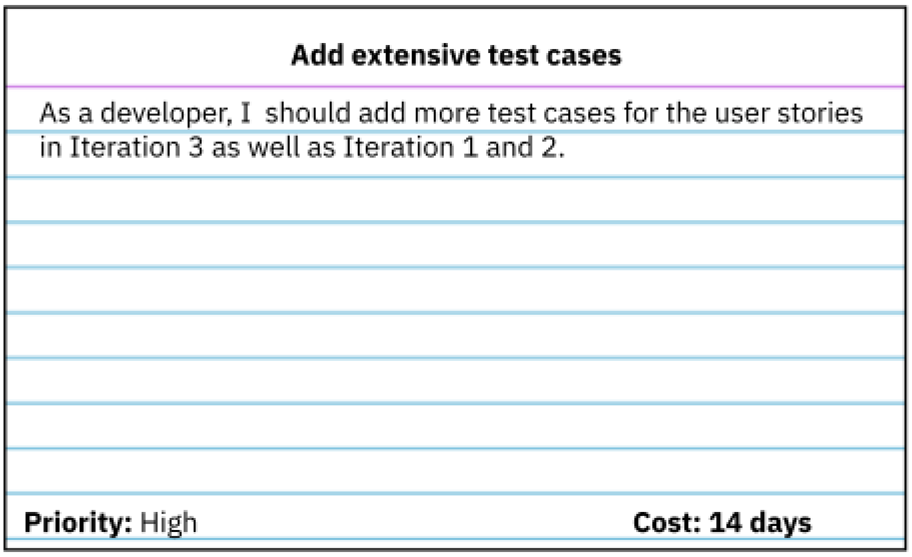
Developer stories



Refactor code from Iteration 2

As a developer, I should refactor any code smells from Iteration 2.

Priority: High **Cost: 14 days**



Add extensive test cases

As a developer, I should add more test cases for the user stories in Iteration 3 as well as Iteration 1 and 2.

Priority: High **Cost: 14 days**

Development tasks assigned in iteration 3

Manasvi:

- Upload Function
- Refactoring code

Angela:

- Bookmark feature
- Refactoring code

Yuvtesh:

- Integration and JUnit testing
- Database

Ammiel:

- Updating database
- Testing
- Refactoring code

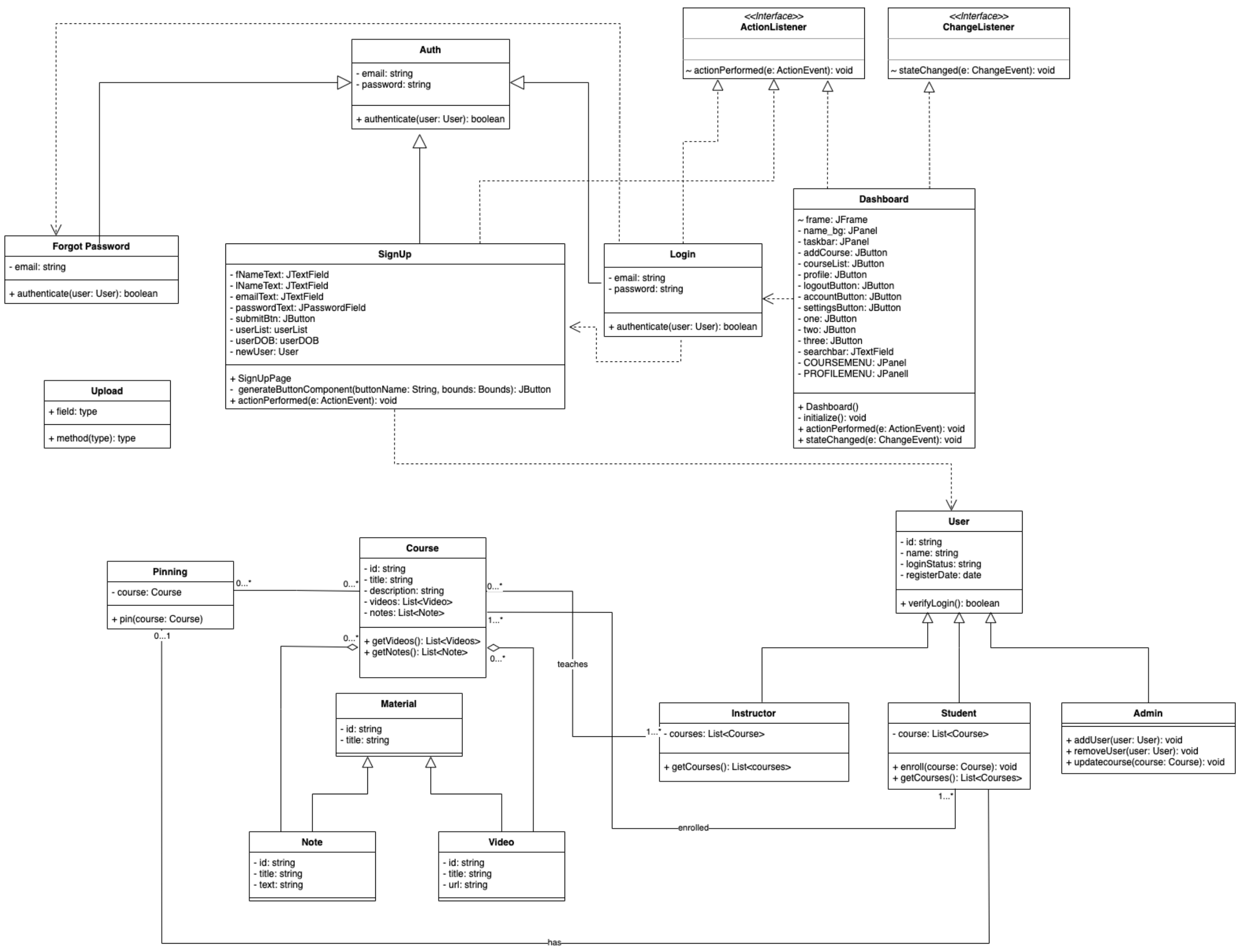
Kevin

- Topics and Study Materials
- Refactoring code

Actual time taken to complete the developmental tasks

- Upload Function: 15 days
- Refactoring code: 15 days
- Bookmark feature: 7 days
- Integration and JUnit testing: 16 days
- Updating database: 14 days
- Topics and Study Materials: 14 days

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.

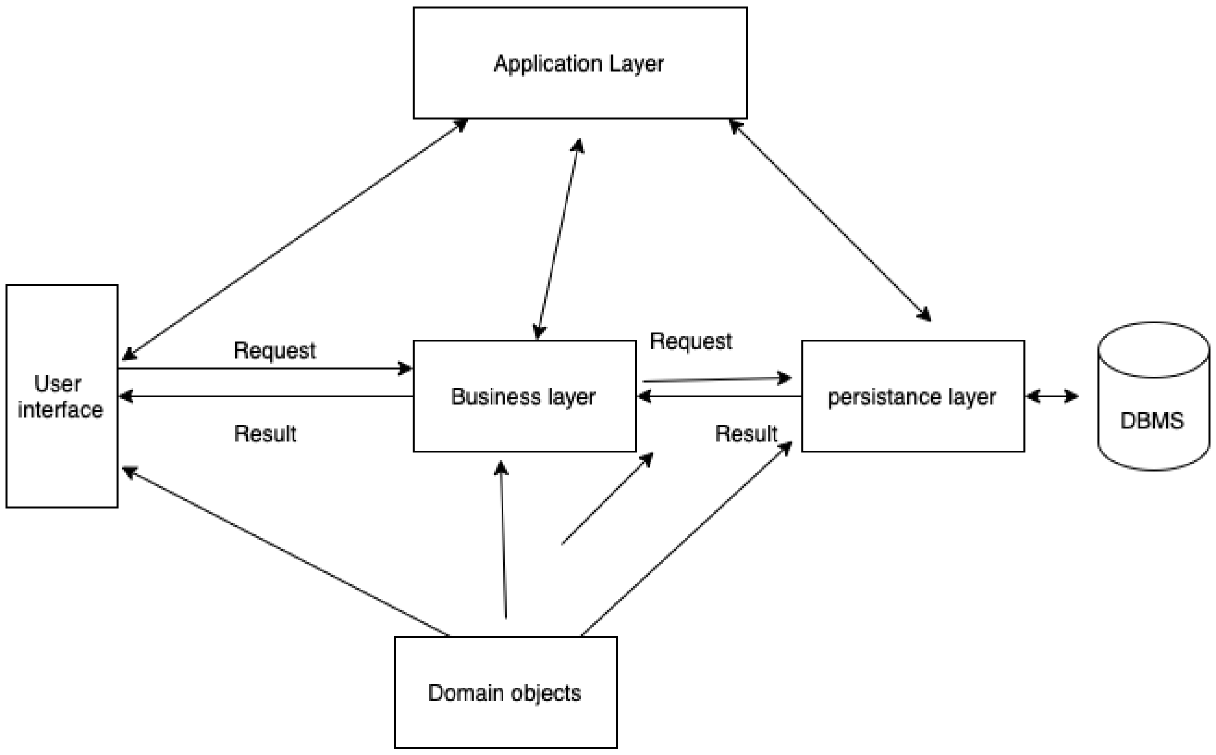
Platform Architecture

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.

Refactoring

Our team had to refactor portions of our codebase to utilize design patterns such as Singleton,