System Design Document V3

Exam Jam

Team hireme

March 2023

Table of Contents

CRC Cards

Database Models

<table>
<tr><th colspan="3">Course</th></tr>
<tr><td><b>Parent Class</b>
● Model</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b></td></tr>
<tr><td colspan="2"><b>Responsibilities</b>
● Database model that represents a course at UofT</td><td><b>Fields</b>
● courseCode: string
● title: string
● description: string
● programArea: string[]
● campus: string[]</td></tr>
</table>

<table>
<tr><th colspan="3">Exam</th></tr>
<tr><td><b>Parent Class</b>
● Model</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Course</td></tr>
<tr><td colspan="2"><b>Responsibilities</b>
● Database model that represents an exam belonging to a course</td><td><b>Fields</b>
● examId: string
● courseCode: string
● link: string
● data: string | Buffer</td></tr>
</table>

<table>
<tr><th colspan="3">Post</th></tr>
<tr><td><b>Parent Class</b>
● Model</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Exam
● Vote
● Comment</td></tr>
<tr><td colspan="2"><b>Responsibilities</b>
● Database model that represents a discussion post belonging to an exam</td><td><b>Fields</b>
● examId: string
● postId: string
● author: string
● body: string
● images: string[]
● upvotes: number
● downvotes: number
● tags: string[]</td></tr>
</table>

<table>
<tr><th colspan="3">PiazzaPost</th></tr>
<tr><td><b>Parent Class</b>
● Model</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● PiazzaComment</td></tr>
</table>

<table>
<tr><td><b>Responsibilities</b>
● Database model that represents a post scraped from a piazza forum</td><td><b>Fields</b>
● courseCode: string
● forumId: string
● postNumber: number
● title: string
● content: string
● createdAt: Date</td></tr>
</table>

<table>
<tr><th colspan="3">PiazzaComment</th></tr>
<tr><td><b>Parent Class</b>
● Model</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● PiazzaPost</td></tr>
<tr><td colspan="2"><b>Responsibilities</b>
● Database model that represents a comment under a piazza post</td><td><b>Fields</b>
● id: string
● postId: string
● type: Enum(s_answer, i_answer, followup, feedback)
● content: string
● parentId: string | null
● children: PiazzaComment[]</td></tr>
</table>

<table>
<tr><th colspan="3">Tag</th></tr>
<tr><td><b>Parent Class</b>
● Model</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Course
● Post</td></tr>
<tr><td colspan="2"><b>Responsibilities</b>
● Database model that represents a tag that can be attached to a Post</td><td><b>Fields</b>
● courseCode: string
● postId: string
● name: string</td></tr>
</table>

<table>
<tr><th colspan="3">Comment</th></tr>
<tr><td><b>Parent Class</b>
● Model</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Post
● Vote</td></tr>
<tr><td colspan="2"><b>Responsibilities</b>
● Database model that represents a comment under a post</td><td><b>Fields</b>
● postId: string
● commentId: string
● parentCommentId: string | null
● author: string
● body: string
● upvotes: number
● downvotes: number</td></tr>
</table>

<table>
<tr><th colspan="3">User</th></tr>
<tr><td><b>Parent Class</b>
● Model</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
●</td></tr>
<tr><td colspan="2"><b>Responsibilities</b>
● Database model that represents application users</td><td><b>Fields</b>
● username: string
● email: string
● active: boolean
● password: string
● isModerator: boolean</td></tr>
</table>

<table>
<tr><th colspan="3">Vote</th></tr>
<tr><td><b>Parent Class</b>
● Model</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Comment
● Post</td></tr>
<tr><td colspan="2"><b>Responsibilities</b>
● Database model that represents the a collection of votes made by a user on a specific item</td><td><b>Fields</b>
● userEmail: string
● type: up | down
● itemId: string</td></tr>
</table>

<table>
<tr><th colspan="3">Bookmark</th></tr>
<tr><td><b>Parent Class</b>
● Model</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Course
● Exam
● Post
● User</td></tr>
<tr><td colspan="2"><b>Responsibilities</b>
● Database model that represents either a course, exam, or post that the user has bookmarked.</td><td><b>Fields</b>
● userEmail: string;
● type: string;
● courseCode: string;
● examId: string;
● postId: string;</td></tr>
</table>

Backend

<table>
<tr><th colspan="3">App</th></tr>
<tr><td><b>Parent Class</b>
● None</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Everything</td></tr>
</table>

<table>
<tr><td><b>Responsibilities</b>
● The main entry point into the backend API</td></tr>
</table>

<table>
<tr><th colspan="3">Server</th></tr>
<tr><td><b>Parent Class</b>
● None</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● App</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● Responsible for running the App on a specific port/host</td></tr>
</table>

<table>
<tr><th colspan="3">UsersController</th></tr>
<tr><td><b>Parent Class</b>
● BaseController</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● User
● Visited</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● Responsible for everything related to a users profile
○ Managing fields such as username, password etc
○ Account deletion
○ History management</td></tr>
</table>

<table>
<tr><th colspan="3">AuthController</th></tr>
<tr><td><b>Parent Class</b>
● BaseController</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● User</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● Responsible for handling authentication i.e. register, login, and logout</td></tr>
</table>

<table>
<tr><th colspan="3">ExamsController</th></tr>
<tr><td><b>Parent Class</b>
● BaseController</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Exam</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● Primarily responsible for querying exam files and metadata from the database</td></tr>
</table>

<table>
<tr><th colspan="3">CoursesController</th></tr>
<tr><td><b>Parent Class</b>
● BaseController</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Course</td></tr>
</table>

Responsibilities ● Primarily responsible for querying course information from the database

PostsController		
Parent Class ● BaseController	**Sub Class** ● None	**Collaborations** ● Post ● Comment ● Tag
Responsibilities ● Responsible for managing posts (Primarily basic CRUD)		

CommentsController		
Parent Class ● BaseController	**Sub Class** ● None	**Collaborations** ● Post ● Comment
Responsibilities ● Responsible for managing comments (Primarily basic CRUD)		

Frontend

<table>
<tr><th colspan="3">App Page</th></tr>
<tr><td><b>Parent Class</b>
● None</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● All pages</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● The entry point of the application into the frontend
● Responsible for routing the browser to different pages</td></tr>
</table>

<table>
<tr><th colspan="3">Register Page</th></tr>
<tr><td><b>Parent Class</b>
● None</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Login Page</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● Provides a way for users to create accounts by filling out a form
● Will also supply a way for users to reset their forgotten password or resend account verification emails</td></tr>
</table>

<table>
<tr><th colspan="3">Login Page</th></tr>
<tr><td><b>Parent Class</b>
● None</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Register Page</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● Provides a way for users to authenticate themselves by logging in to their accounts</td></tr>
</table>

<table>
<tr><th colspan="3">Navbar</th></tr>
<tr><td><b>Parent Class</b>
● None</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● All Pages</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● Provides an easy way for users to navigate to different pages in the app</td></tr>
</table>

<table>
<tr><th colspan="3">Dashboard Page</th></tr>
<tr><td><b>Parent Class</b>
● None</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Course Search Page
● Course Page
● Navbar</td></tr>
<tr><td colspan="3"><b>Responsibilities</b></td></tr>
</table>

- The page that users are redirected to after logging in
- Contains bookmarked/enrolled courses and recent activity

Course Search Page		
Parent Class ● None	**Sub Class** ● None	**Collaborations** ● Course Page ● Navbar
Responsibilities ● Clicking on an icon on the navbar takes users to this page ● Provides a search bar for users to search for courses ○ A paginated list of course results appear		

Course Page		
Parent Class ● None	**Sub Class** ● None	**Collaborations** ● Course Search Page ● Navbar ● Exams Page ● Posts Feed ● Piazza Posts Feed
Responsibilities ● After clicking on either a bookmarked or a search result course, the user is redirected to this page ● Contains a table with links to all exams for that specific course ● Contains a feed or posts related to the course ● Contains a link that redirects users to all exams for that specific course		

Exam Page		
Parent Class ● None	**Sub Class** ● None	**Collaborations** ● Course Page ● Navbar ● Exam Page ● Posts Feed
Responsibilities ● After a user is redirected to this page from the page of a specific course, the exam is displayed in an embedded pdf viewer ● Underneath the pdf viewer, there is a feed of posts that have been made related to that specific exam		

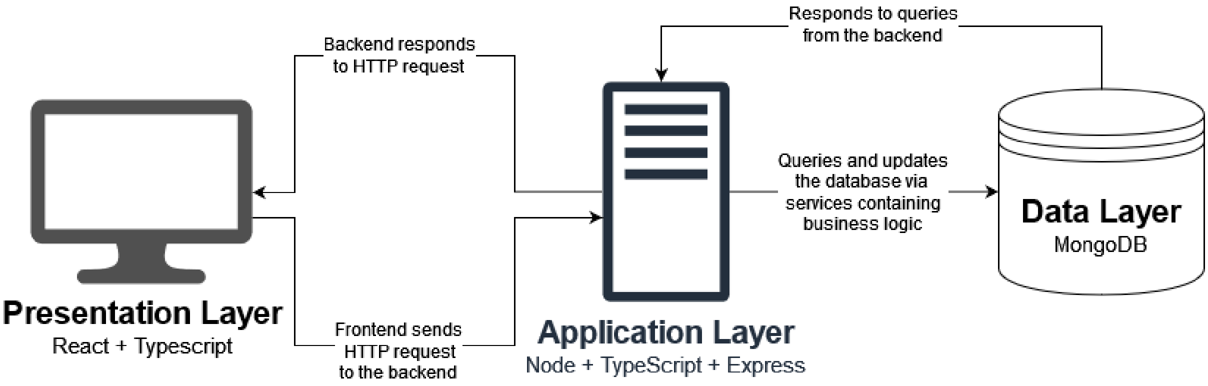
Posts Feed

<table>
<tr><td><b>Parent Class</b>
● None</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Course Page
● Exam Page</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● Responsible for displaying a feed of discussion posts related to a course or exam.
● Posts are displayed in a paginated manner and can be filtered or sorted</td></tr>
</table>

<table>
<tr><th colspan="3">Post Page</th></tr>
<tr><td><b>Parent Class</b>
● None</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Course Page
● Posts Feed</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● Responsible for displaying the information of a crowd-sourced discussion post including the title, content, and all comments.</td></tr>
</table>

<table>
<tr><th colspan="3">Piazza Post Page</th></tr>
<tr><td><b>Parent Class</b>
● None</td><td><b>Sub Class</b>
● None</td><td><b>Collaborations</b>
● Course Page
● Post Feed</td></tr>
<tr><td colspan="3"><b>Responsibilities</b>
● Responsible for displaying the information of a piazza post including the title, content, any student/instructor answers, and all followup discussions.</td></tr>
</table>

System Architecture Diagram


Backend responds to HTTP request
Responds to queries from the backend
Queries and updates the database via services containing business logic
Presentation Layer
React + Typescript
Frontend sends HTTP request to the backend
Application Layer
Node + TypeScript + Express
Data Layer
MongoDB

Reference

- https://www.linuxjournal.com/article/3508

Explanation

- Our application will follow the 3-tiered architecture
  - Presentation: Frontend with React and TypeScript
  - Application: Backend API with TypeScript, Node, and Express.js, and Python for simple scripts
  - Data Layer: Database primarily using MongoDB
- The frontend will only ever communicate with the backend which will act as a middleman between the frontend and the database

Error Handling & Exceptional Cases

- There will be two explicit layers of schema validation and one implicit layer.
  - The two explicit layers are:
    - Frontend form validation using the yup npm library
      - Primarily for a better UX
    - Backend API schema validation also using yup + TSOA
      - TSOA checks for the presence of required items and yup checks for properties of each field (i.e. a password is 8 characters long or an email follows a certain regex)
    - Database validation (on insertion)
      - We are using the mongoose ODM which allows us to define a structure/schema for a specific model. It takes care of validating all database objects on insertion or update queries.
- Generic error handling will make use of HTTP status codes and error messages