

Focal Point: Yuxuan Zhao, yz4397

Team Members:

Jin Liu, jl6276

Yiming Wang, yw3931

Noah Zhou, zz2900

Bon Li, bl2944

TA: Rattandeep Singh (rs4478)

1. Introduction: Description of your service.

- Class Tutor Platform
 - Our Class Tutor ChatBot serves as a pioneering platform designed to bolster students' understanding of class materials and concepts. Employing the state-of-the-art capabilities of the Chatglm2 framework, this bot harnesses advanced natural language processing to deliver immediate, precise responses to student queries. In addition, our platform features an expansive textbook database, offering students pertinent textbook details and procurement links. A unique selling point remains our insightful dashboard, showcasing class performance trends from previous years.

2. Personas and Roles: Potential Users and Roles in your service.

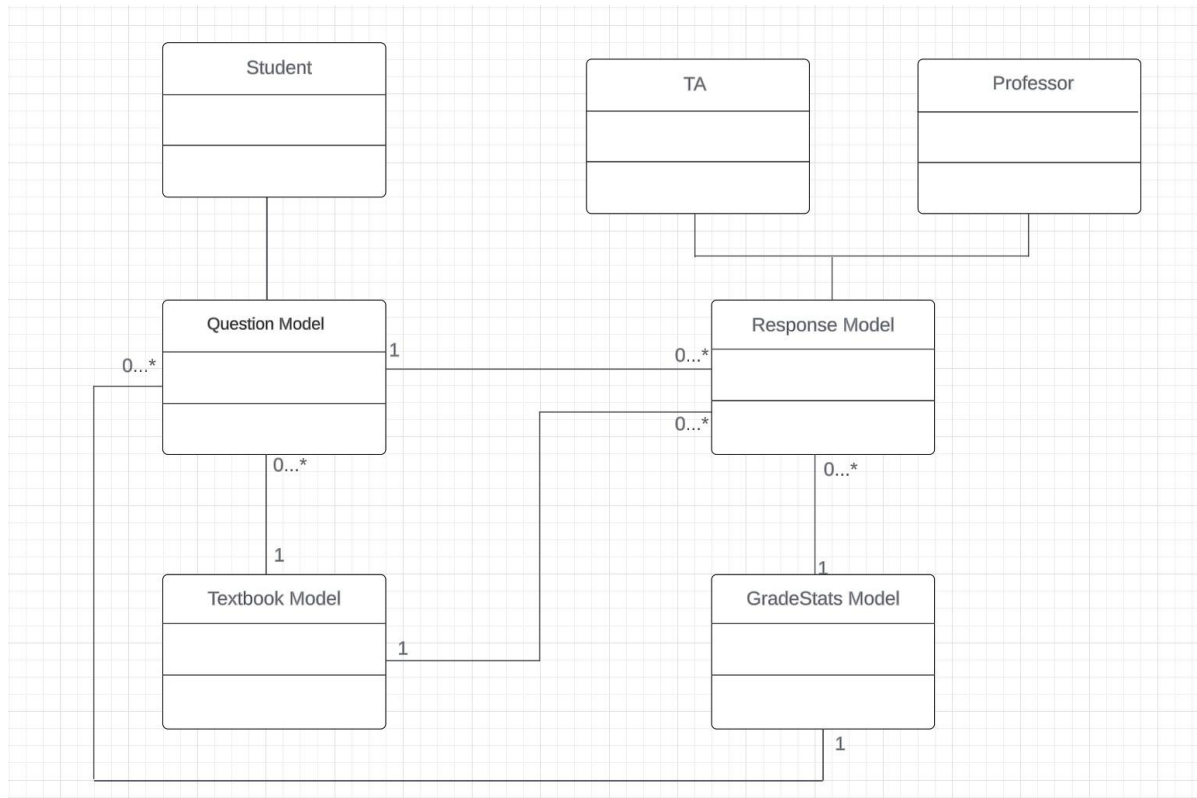
- Students:
 - Use the WebUI to ask questions about class material
 - Receive real-time responses to their queries
 - Search textbooks by keyword.
 - View textbook details and purchase links.
 - View own historical course performance statistics on dashboard.
- TA: Prep latest class material
 - Prepares the latest class material
 - Feed the chatbot with frequently asked questions and their answers
- Instructor:
 - Collects student questions to improve lecture quality and content

- Monitors bot's interactions to identify areas where students are struggling
- View performance changes over the years for the teaching courses on dashboard.

3. Key Features: Main features of your application

- Virtual TA:
 - Provides real-time responses to students' questions
 - Supports multimedia content (videos, diagrams)
 - Offers links to additional resources
- Textbook look up:
 - An up-to-date textbook database providing detailed information and purchase links.
- Dashboard of Past Class:
 - Enables students to view historical grade distributions.
 - Grants educators insights to refine lecture content.

4. Domain Model Diagram: Visualization of your data in your microservices (Please Indicate what microservice each model belongs to by surrounding them in the same box.)



5. Resource Paths: API Paths for your application for each microservice.

- /api/docs: OpenAPI documentation
- ChatBot Service:
 - [POST] /api/chatbot/questions: Submit a new question
 - [GET] /api/chatbot/responses/{questionId}: Get response for a specific question
- Analytics Service:
 - [GET] /api/analytics/interactions: Retrieve all interactions
 - [GET] /api/analytics/topics: Retrieve topic-based analytics
- Textbook Service:
 - [POST] /api/textbook/add: Add a new textbook item with descriptions
 - [GET] /api/textbook/search: Search the textbooks given some keywords
 - [GET] /api/textbook/view/{textbookId}: View details of a textbook.
 - [GET] /api/textbook/buy/{textbookId}: Buy a textbook via purchasing link

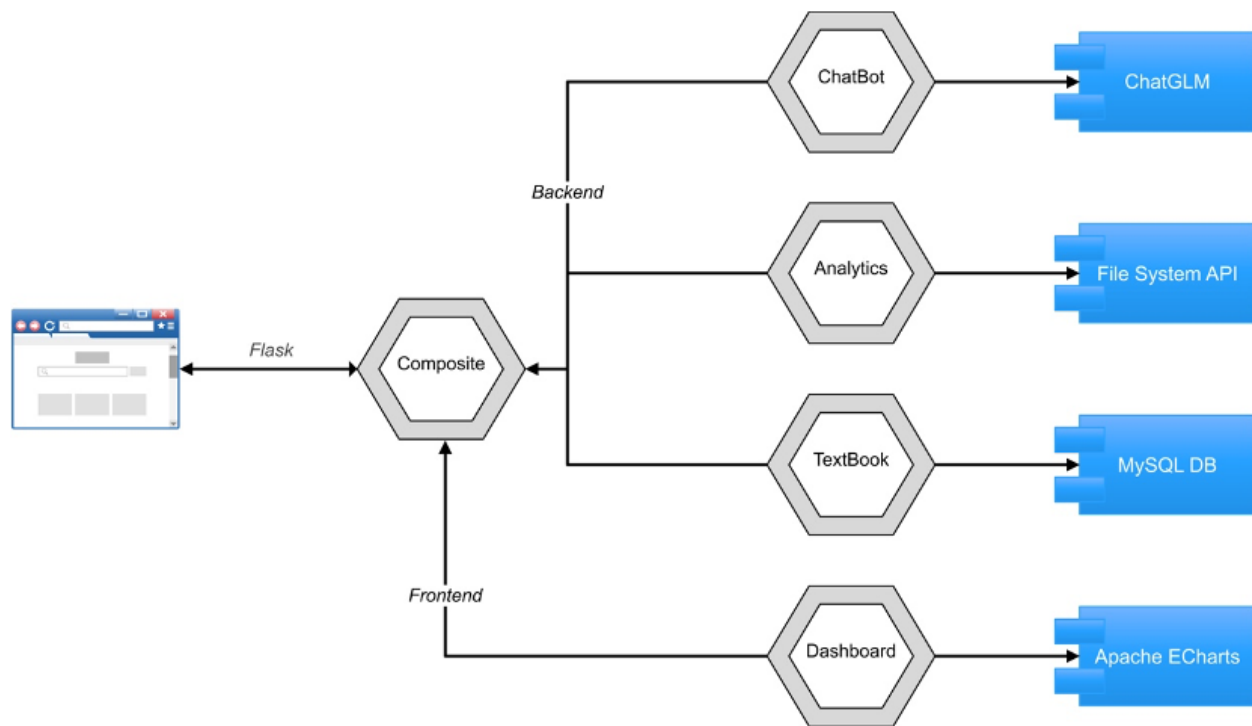
- Dashboard Service:
 - [GET] /api/dashboard/{studentId}: View a student's basic information and grade performance
 - [GET] /api/dashboard/{courseId}: View class performance statistics for a course in recent years

6. Logical View: High-level visualization of how each microservice will be deployed and functioning.

We have adopted a logical view rooted in the principles of Hexagonal Architecture. In accordance with the delineated Resource Path, this architectural paradigm is manifest as a division into three discrete microservices, namely ChatBot, Textbook, and Dashboard. These microservices collectively represent the entirety of the project, encompassing both its front-end and back-end components.

Within the shading area of each hexagon(i.e. microservice), we find the integration points for external APIs. Specifically:

- In the context of back-end microservices, ChatBot establishes a connection with ChatGLM, while Analytics interfaces with the File System API for the purposes of file manipulation and transfer. The Textbook service, on the other hand, leverages the MySQL database as its primary data repository.
- For front-end microservices, our design contemplates the integration of Apache Echarts to facilitate the presentation of diverse visual data representations.



7. Physical View: How each microservice instance will be deployed on the Cloud Infrastructure.

The physical view shows how various microservices are deployed on cloud infrastructure. Here's a detailed breakdown:

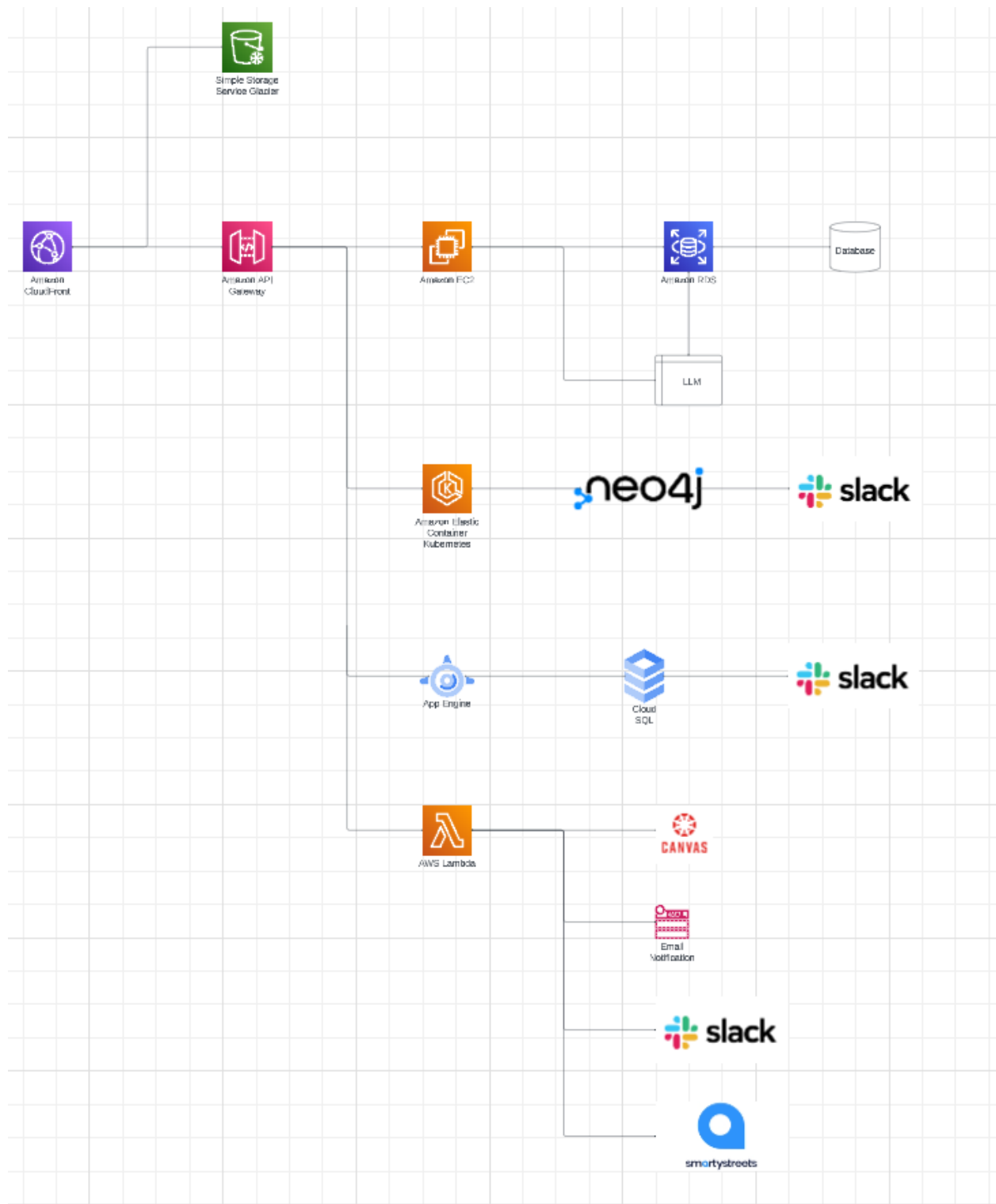
Amazon CloudFront : This is a content delivery network offered by AWS. Typically used for serving static content or redirecting traffic. It appears to be deployed at the edge to cater to global traffic. It is deployed with the help of Simple Storage Service (S3) + Service Operator. Amazon S3 is a storage service by AWS. It's coupled with a 'Service Operator', indicating there might be a specific service that manages S3 operations.

Virtual TA microservice: The Virtual TA microservice is deployed by using Amazon EC2, which runs on a virtual server in the AWS cloud. The Amazon RDS is connected to manage the database connection. The Larger Language Model (LLM) is connected through API and provides answers for clients of microservice. Assisting with the GitHub API get way the code can be loaded and modified.

Past class microservice: this microservice is deployed by using Amazon Elastic Container Kubernetes (EKS), which connects to neo4j and slack database and visualizes the searching result for the client.

Book look up microservice: it is deployed by using Google's App Engine, which is a platform-as-a-service that allows apps to be deployed without managing the underlying infrastructure. Cloud SQL is Google's managed SQL database service, which manages the book database in the cloud.

The AWS Lambda, integrated with Canvas, email notification, Slack, and SmartyStreets, supports the main microservices, which provides the verification and notification for users.



8. Codebase:

Microservice-1 Repo: <https://github.com/ChuckYuxuan/Microservice-1>

Microservice-2 Repo: <https://github.com/ChuckYuxuan/Microservice-2>

Microservice-3 Repo: <https://github.com/ChuckYuxuan/Microservice-3>

Front Page Repo: https://github.com/ChuckYuxuan/Front_Page

Project Board: <https://github.com/users/ChuckYuxuan/projects/3>

9. Links to deployment:

Front: <http://cloudcomputing23fall.s3-website.us-east-2.amazonaws.com/>

Microservice - 1:EC2

<http://ec2-3-147-242-212.us-east-2.compute.amazonaws.com:8011/>

Microservice - 2 Google App Engine:

<https://microservice-3-402703.uk.r.appspot.com/>

Microservice - 3 EC2 with Dockerhub

<http://ec2-3-18-212-113.us-east-2.compute.amazonaws.com:8013/>