EXAM TEAM FOR ENROLLMENT

Yaren Delfin Ozen, Orkhan Abilov, Disha Dass, Jack Felix Smirnov, Vireak Tosa

QUALITY SCENARIOS

- Portability
- Security
- Testability
- Modifiability
- Performance

QUALITY SCENARIO - PORTABILITY

Web Browser

Desktop App

QUALITY SCENARIO - PORTABILITY

QUALITY SCENARIO – PORTABILITY

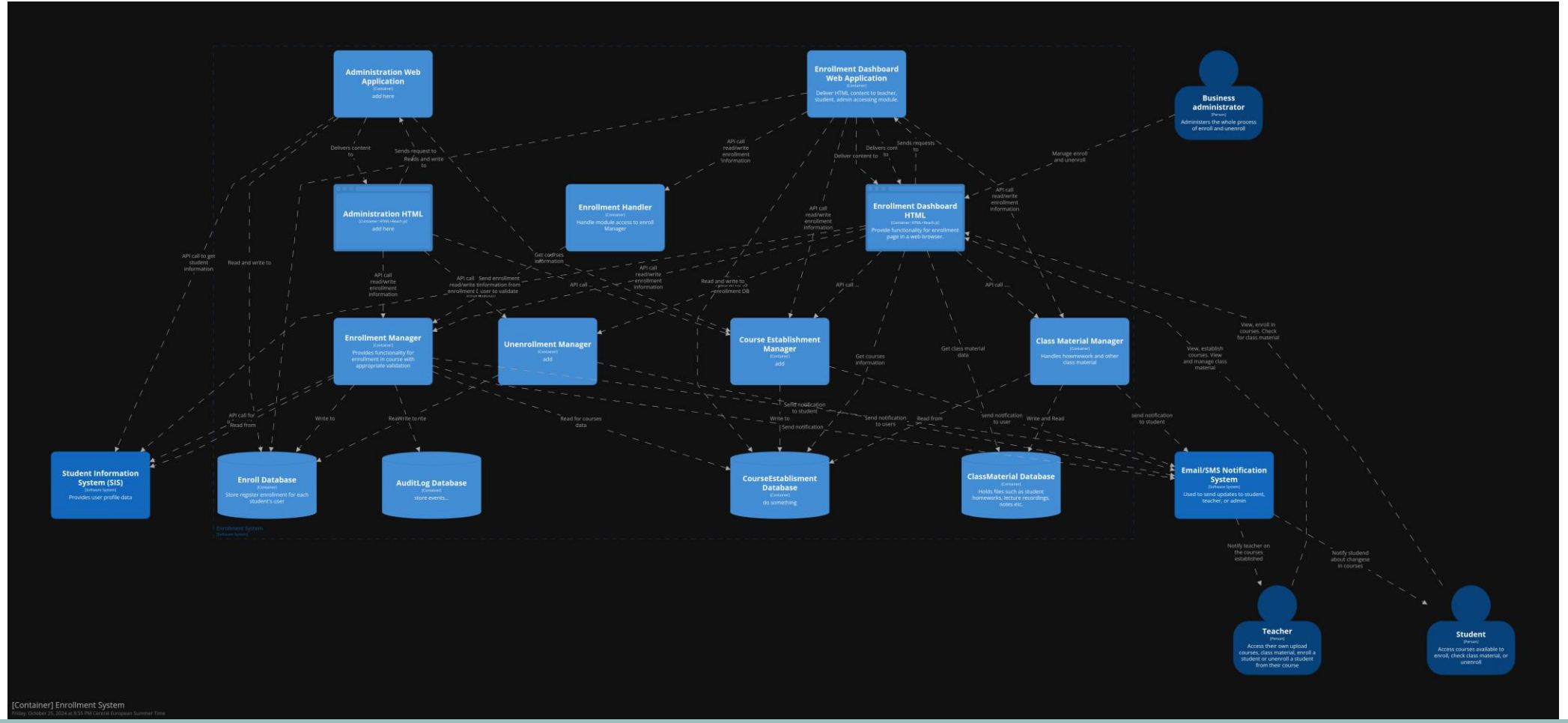
BY JACK FELIX SMIRNOV

- **Source:** The system owner.

- **Stimulus:** A requirement emerges to extend the current web-based enrollment system to operate as a desktop application.

- **Artifact:** The enrollment system, including the front-end user interface and back-end APIs.

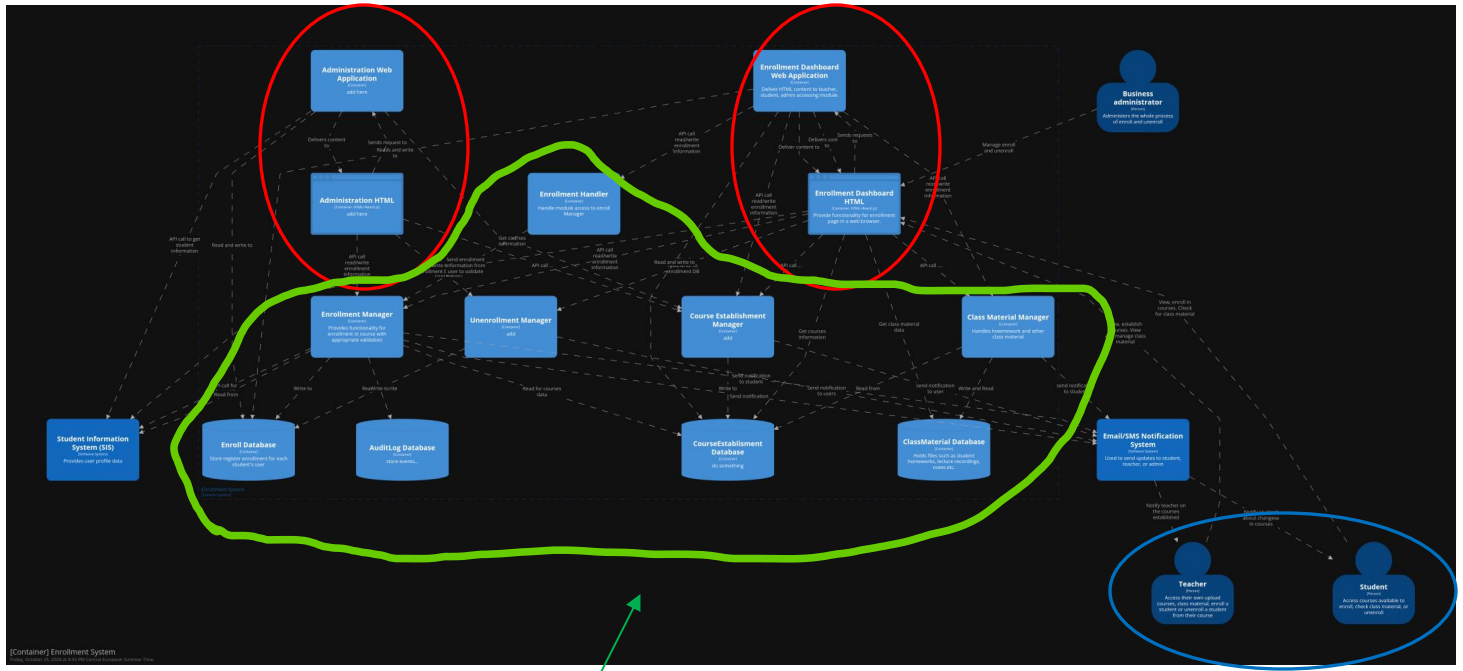
- **Environment:** The web application currently runs on a browser, while the desktop application must run on Windows.



QUALITY SCENARIO - PORTABILITY

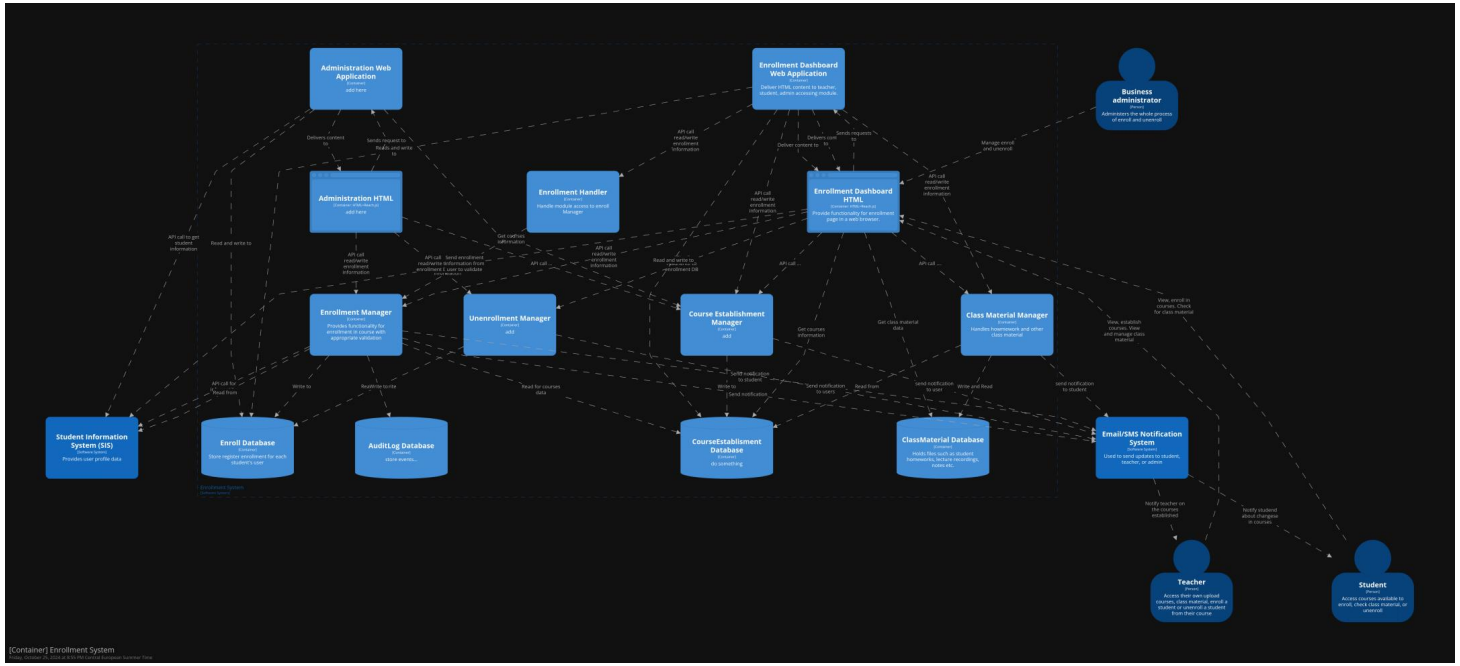
QUALITY SCENARIO - PORTABILITY

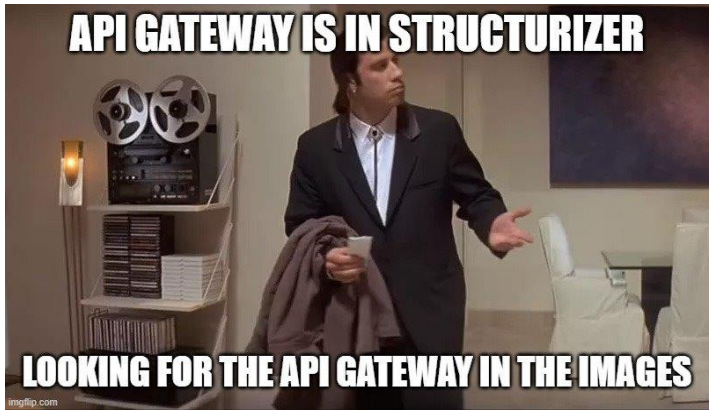
- No clear separation between the front-end and the back-end.
- It appears that the user need to use the back-end to interact with the front-end



QUALITY SCENARIO - PORTABILITY

- Communication with the back-end services would be simpler through an API Gateway container, which is missing.

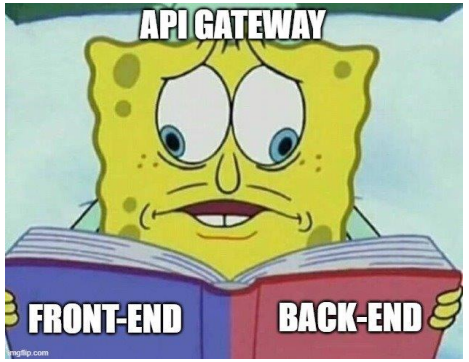




QUALITY SCENARIO - PORTABILITY

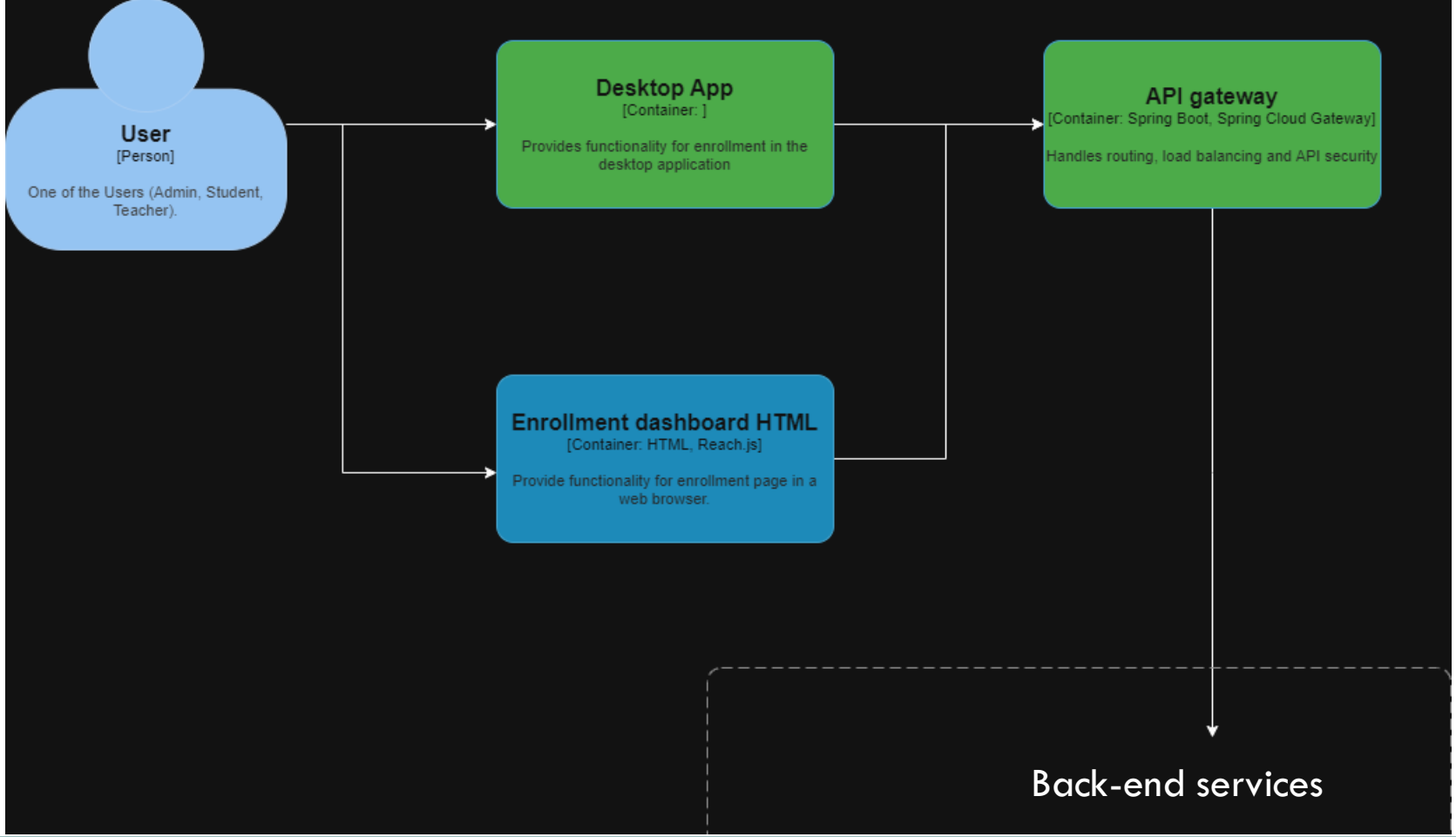
Response:

- The system will clearly separate between the front-end and the back-end services



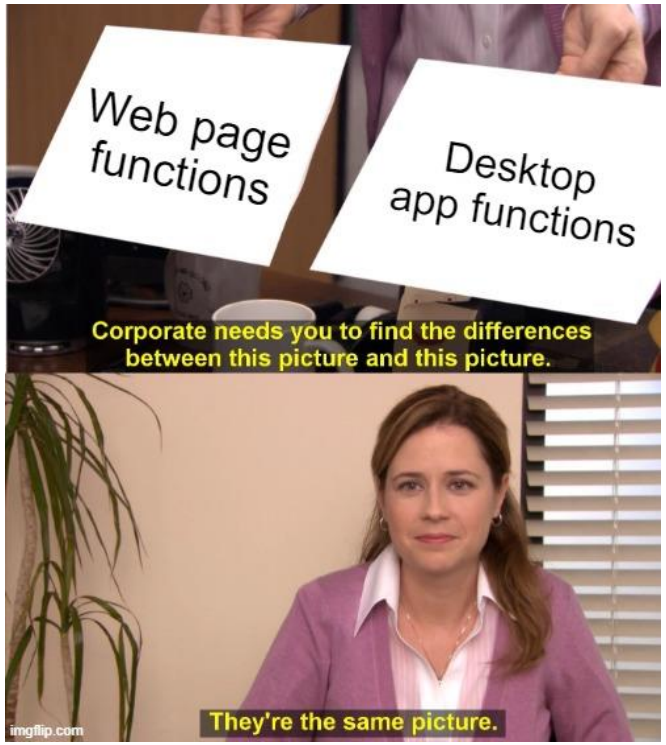
- The system's core functionality is successfully adapted to a desktop application.

- The back-end APIs remains unchanged and are reused without modification.



PORTABILITY UPDATE

QUALITY SCENARIO - PORTABILITY



Response Measure:

- The desktop application is deployed and tested on Windows.

- All web features are functional without additional back-end changes.

QUALITY SCENARIO — SECURITY

BY YAREN DELFIN OZEN

Context

Stimulus: Unauthorized user.

Behavior: Attempts to bypass validation to create a course with invalid data.

Artifact: Course Validator.

Goal: Prevent unauthorized users from compromising the integrity of the system by introducing invalid or malicious data.

QUALITY SCENARIO – SECURITY

Change

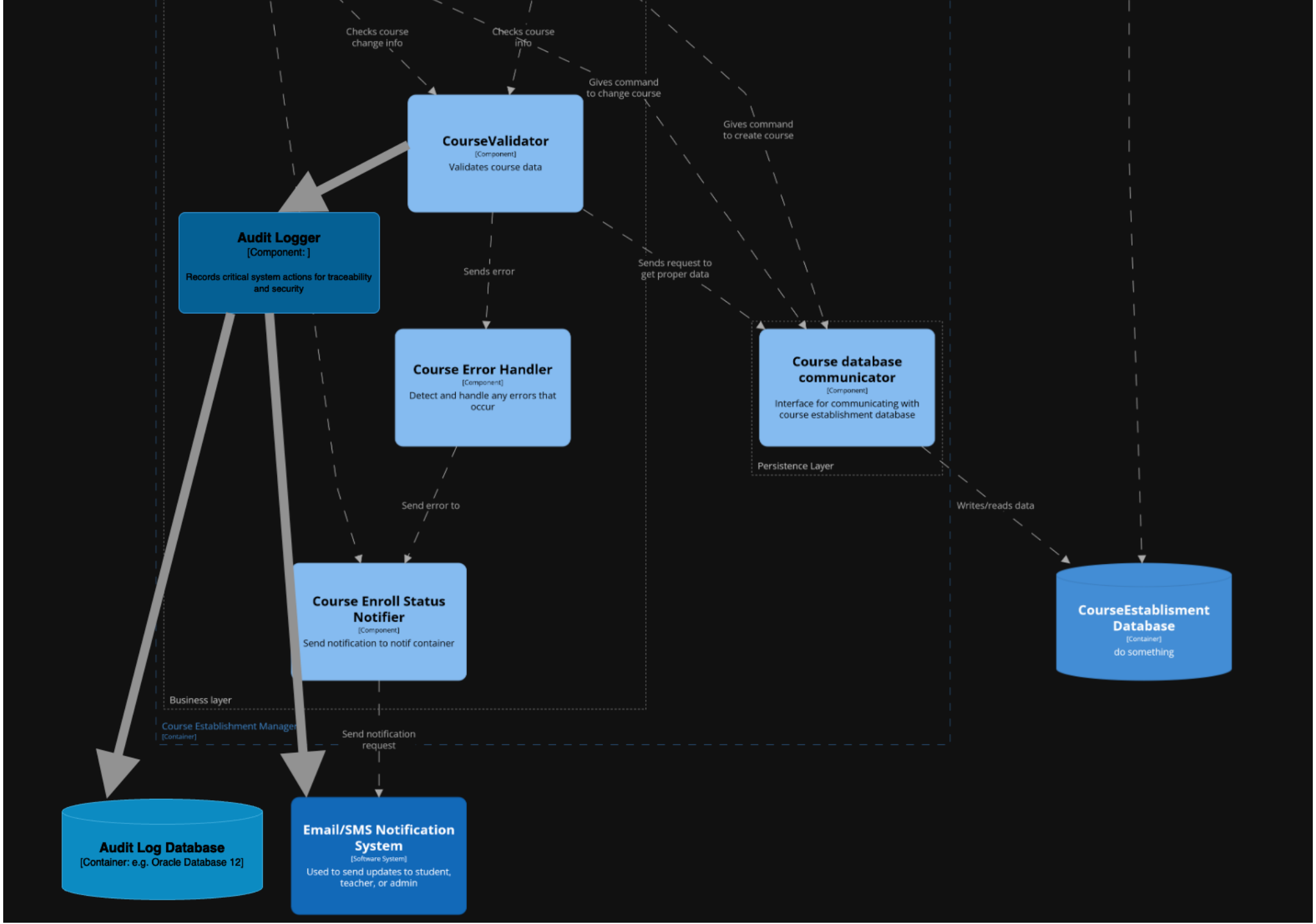
Add audit logger component

Responsibilities:

- Detects and rejects invalid input.
- Logs failed validation attempts.
- Notifies the system administrator of suspicious activities.

Key Features of the change

- Input validation:
  - Strictly enforces data integrity rules and blocks any data that fails validation criteria.
- Audit Trail:
  - Logs all failed validation attempts for monitoring and analysis.
  - Ensures accountability by capturing relevant details (e.g., timestamp, user ID, input data).
- Real-Time Alerts:
  - Immediately notifies administrators about suspicious or repeated failed attempts.
  - Enables timely intervention to address potential security threats.

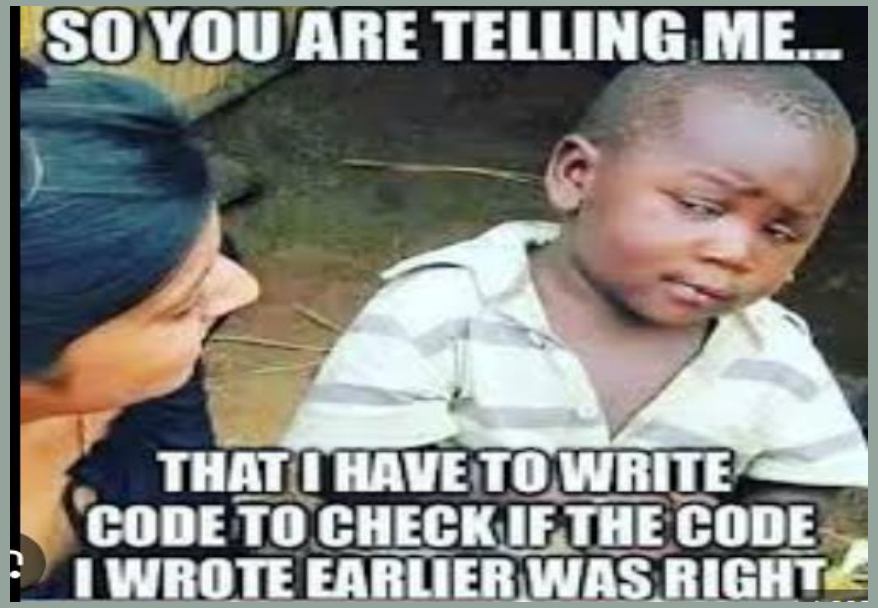

Checks course change info
Checks course info
Gives command to change course
Gives command to create course
CourseValidator
[Component]
Validates course data
Audit Logger
[Component:]
Records critical system actions for traceability and security
Sends error
Sends request to get proper data
Course Error Handler
[Component]
Detect and handle any errors that occur
Course database communicator
[Component]
Interface for communicating with course establishment database
Persistence Layer
Send error to
Writes/reads data
Course Enroll Status Notifier
[Component]
Send notification to notif container
CourseEstablisment Database
[Container]
do something
Business layer
Course Establishment Manager
[Container]
Send notification request
Audit Log Database
[Container: e.g. Oracle Database 12]
Email/SMS Notification System
[Software System]
Used to send updates to student, teacher, or admin

QUALITY SCENARIO – SECURITY

Performance Goal

- Key Metric: Detection and rejection time.
  - Target: Less than 2 seconds to detect, reject, and log unauthorized input.

QUALITY SCENARIO - TESTABILITY BY DISHA DASS



Stimulus

A tester prepares and executes a set of tests to validate the **Homework Controller** and its associated components (e.g., **Homework Verifier**) under controlled conditions.

Source of Stimulus

The **tester** (a user or an automated testing framework) initiates the tests by submitting file upload scenarios.

Artifact

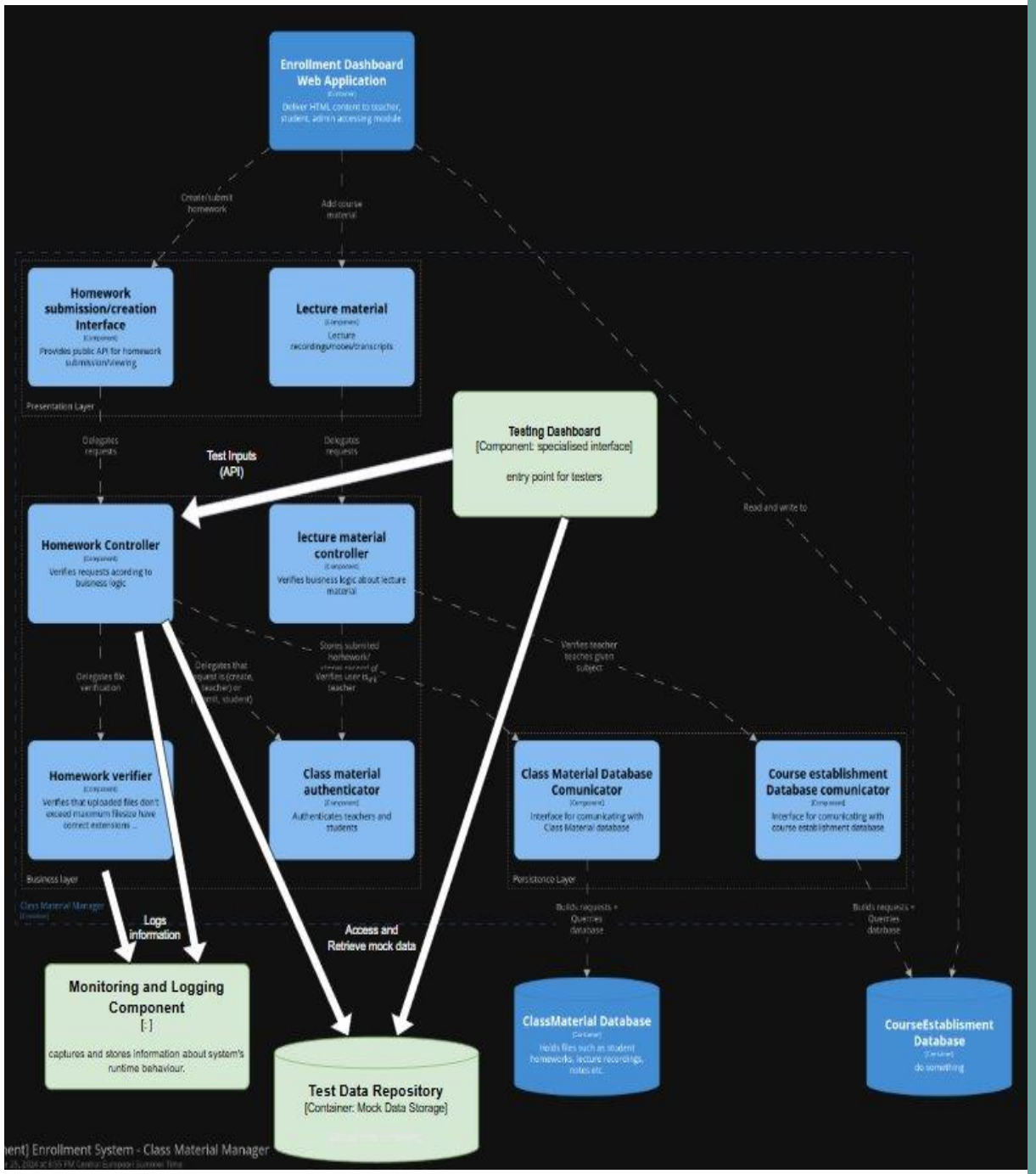
The Homework Controller and Homework Verifier.

Environment

The tests are performed in a **dedicated testing environment.**

Response

- The **Homework Controller** processes the test inputs, delegating validation to the **Homework Verifier**.
- The system generates clear, observable outputs for each test case.
- Logs are generated



Response Measure

- **Fault Detection:** The system quickly identifies and handles invalid files.
- **Coverage:** All defined validation rules (e.g., file size, format) are tested.
- **Observability:** Detailed logs capture every decision, ensuring faults are traceable.

Testability Tactic: Controllability

Focus on adding **specialized interfaces** (APIs) to control and simulate system inputs during tests. This allows testers to:

- Inject controlled inputs (e.g., mock file uploads) into the **Homework Controller**.
- Monitor the system's response to these inputs in real-time.

Implementation in the System

- Specialized Testing API
- Input Simulation
- Immediate Feedback

QUALITY SCENARIO MODIFIABILITY

BY ORKHAN ABILOV

Stimulus

A new feature request requires tracking submission timestamps for assignments and saving a version history for homework updates. Which requires modifying the system logic for submission and update.

Source of Stimulus

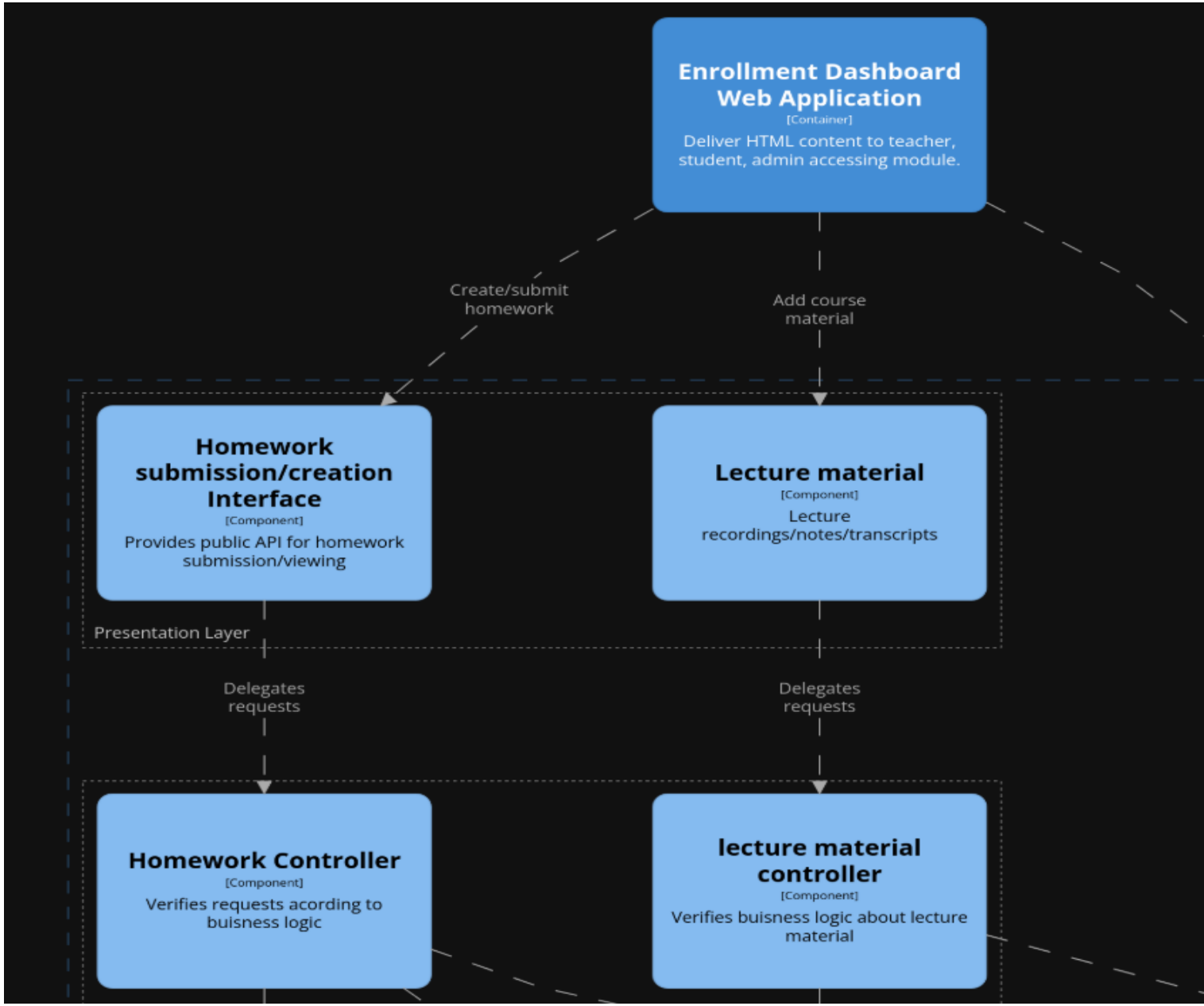
Students and teachers request the feature.

Environment

During active semester usage, with ongoing homework submissions and updates.

Artifact

- Homework submission/creation interface
- Homework Controller
- Class Material Database Communicator



Response

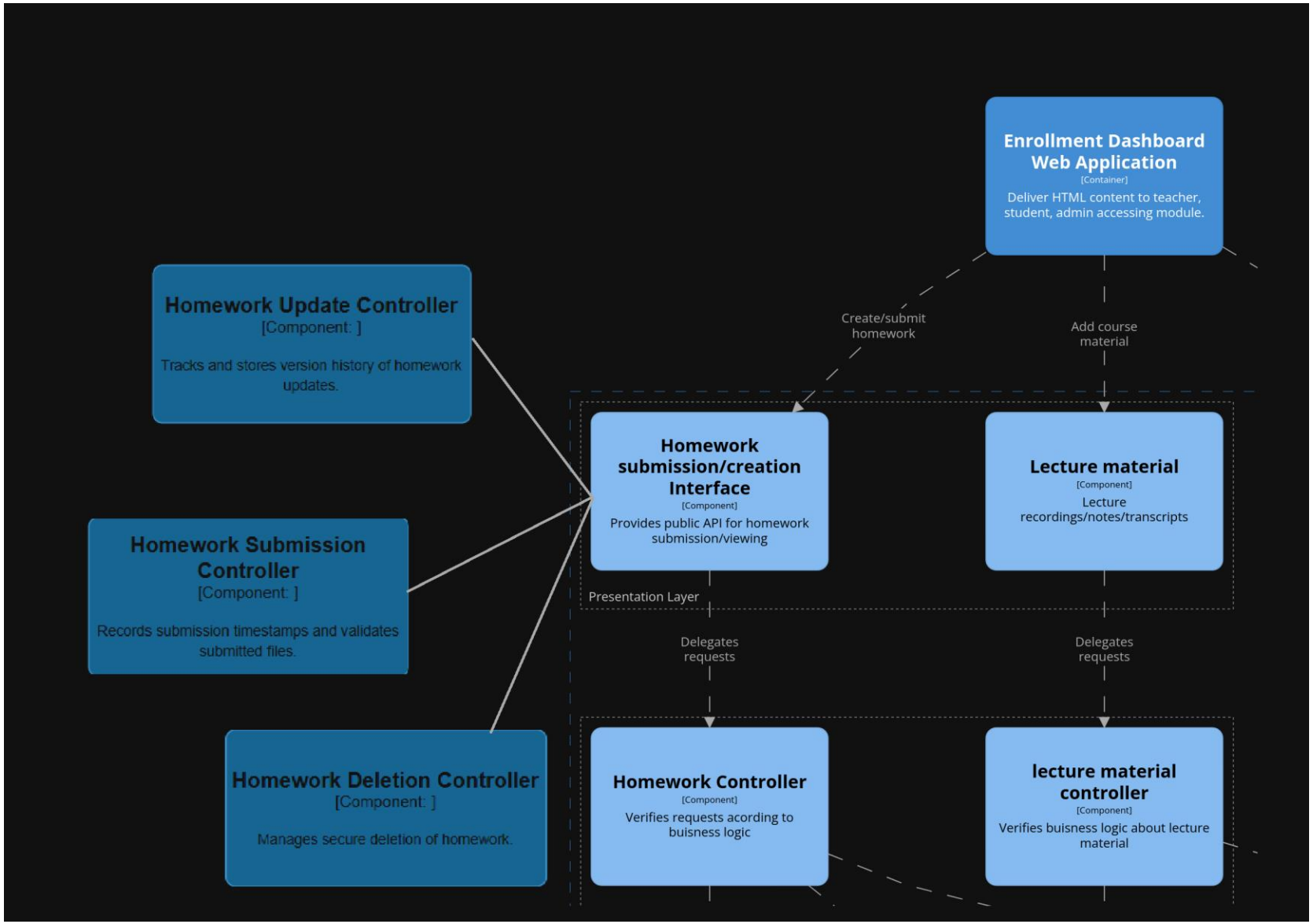
1. Create new separate controllers:

   - Homework Submission Controller

   - Homework Update Controller

   - Homework Deletion Controller

2. Adjust the Homework submission/creation interface to route requests to the appropriate controllers.


Enrollment Dashboard Web Application
[Container]
Deliver HTML content to teacher, student, admin accessing module.
Create/submit homework
Add course material
Homework Update Controller
[Component:]
Tracks and stores version history of homework updates.
Homework submission/creation Interface
[Component]
Provides public API for homework submission/viewing
Lecture material
[Component]
Lecture recordings/notes/transcripts
Homework Submission Controller
[Component:]
Records submission timestamps and validates submitted files.
Presentation Layer
Delegates requests
Delegates requests
Homework Deletion Controller
[Component:]
Manages secure deletion of homework.
Homework Controller
[Component]
Verifies requests acording to buisness logic
lecture material controller
[Component]
Verifies buisness logic about lecture material

Response Measure

- The changes are implemented without disrupting existing functionalities.

- New features (submission tracking and version history) are completed within.

QUALITY SCENARIO PERFORMANCE

BY VITEAK TOSA

Stimulus

The system receives a large number of concurrent requests for processing.

Source of Stimulus

A peak period with multiple users submitting course data simultaneously.

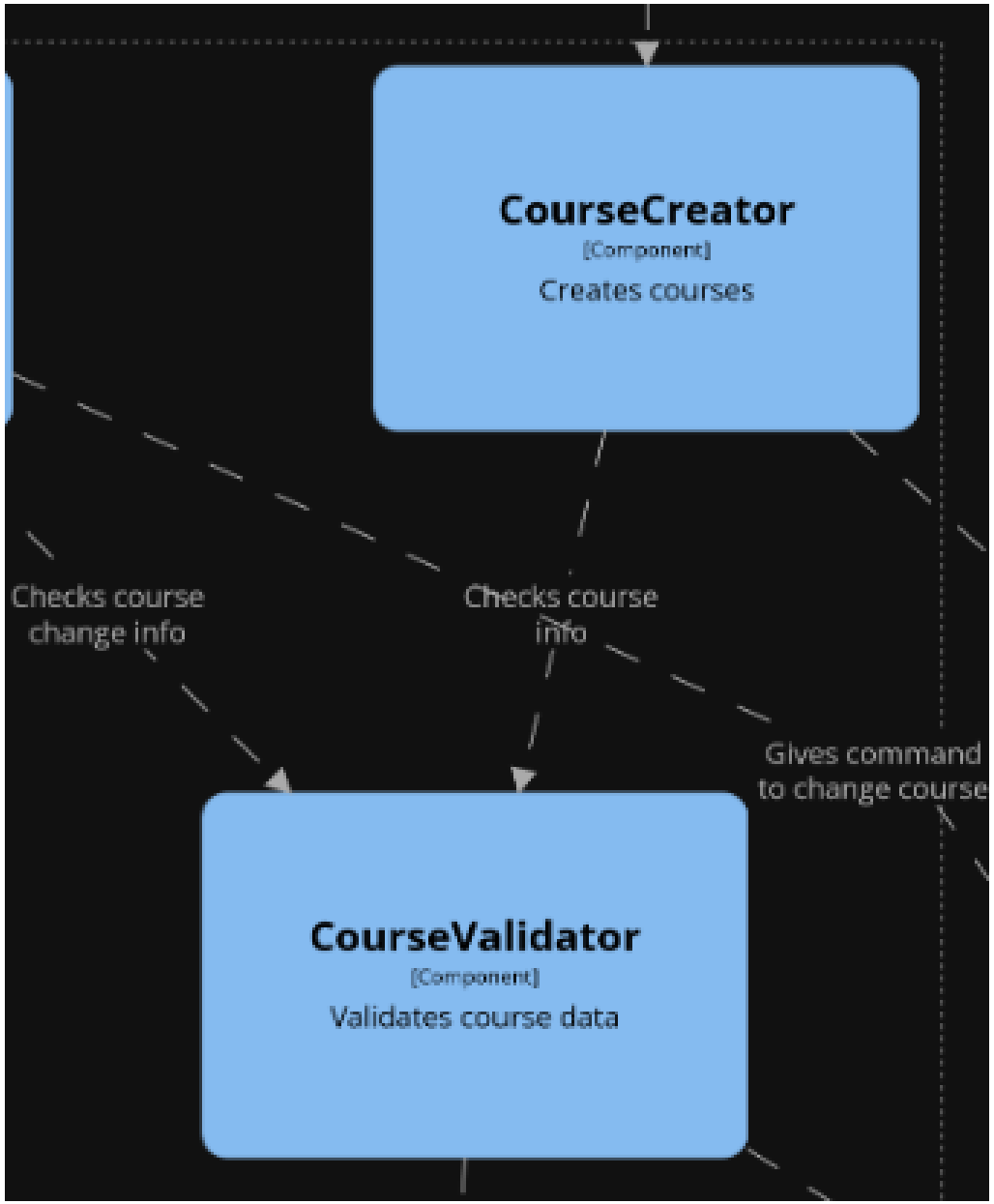
Environment

Production environment with high concurrency.

Artifact:

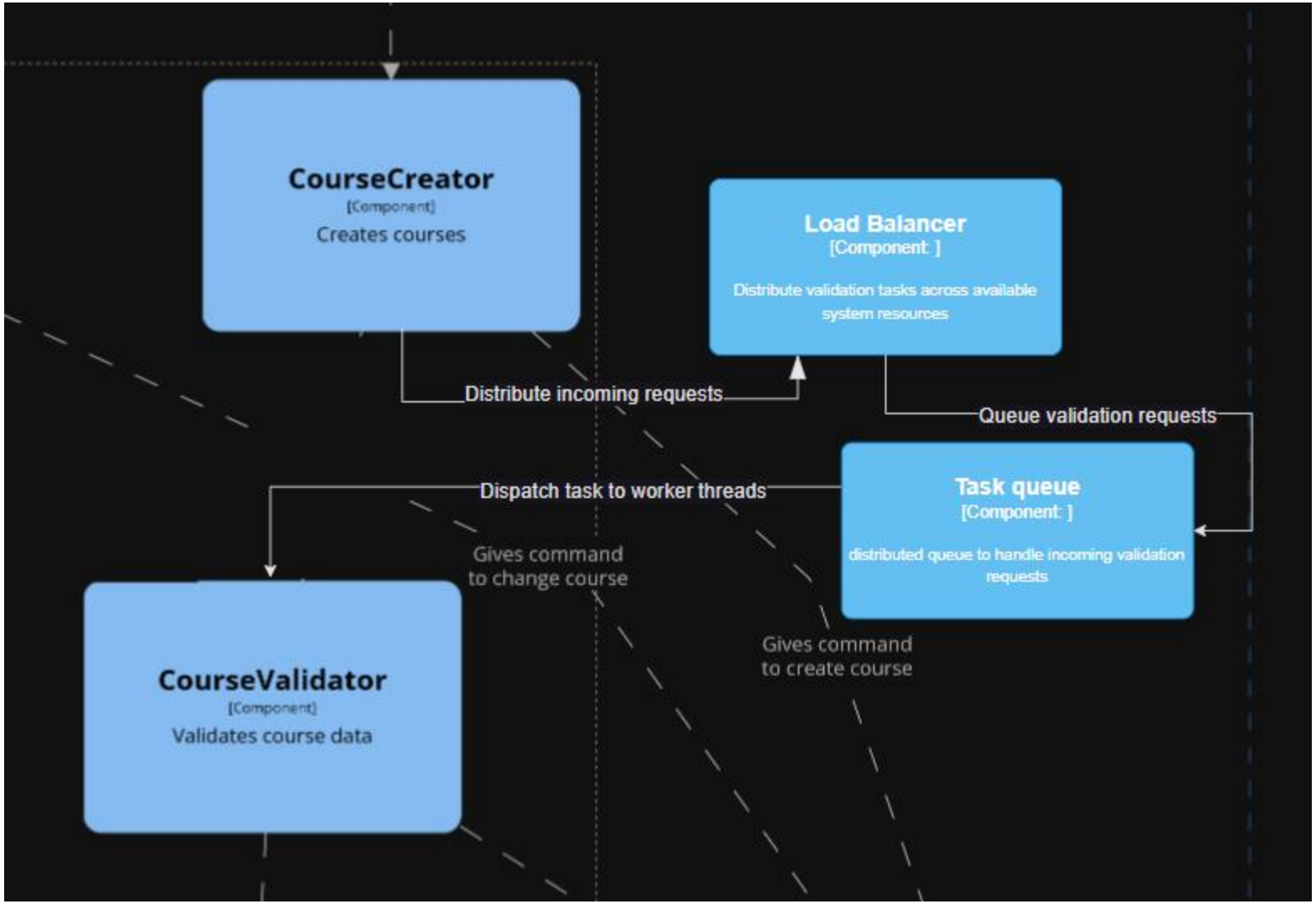
- **CourseCreator**

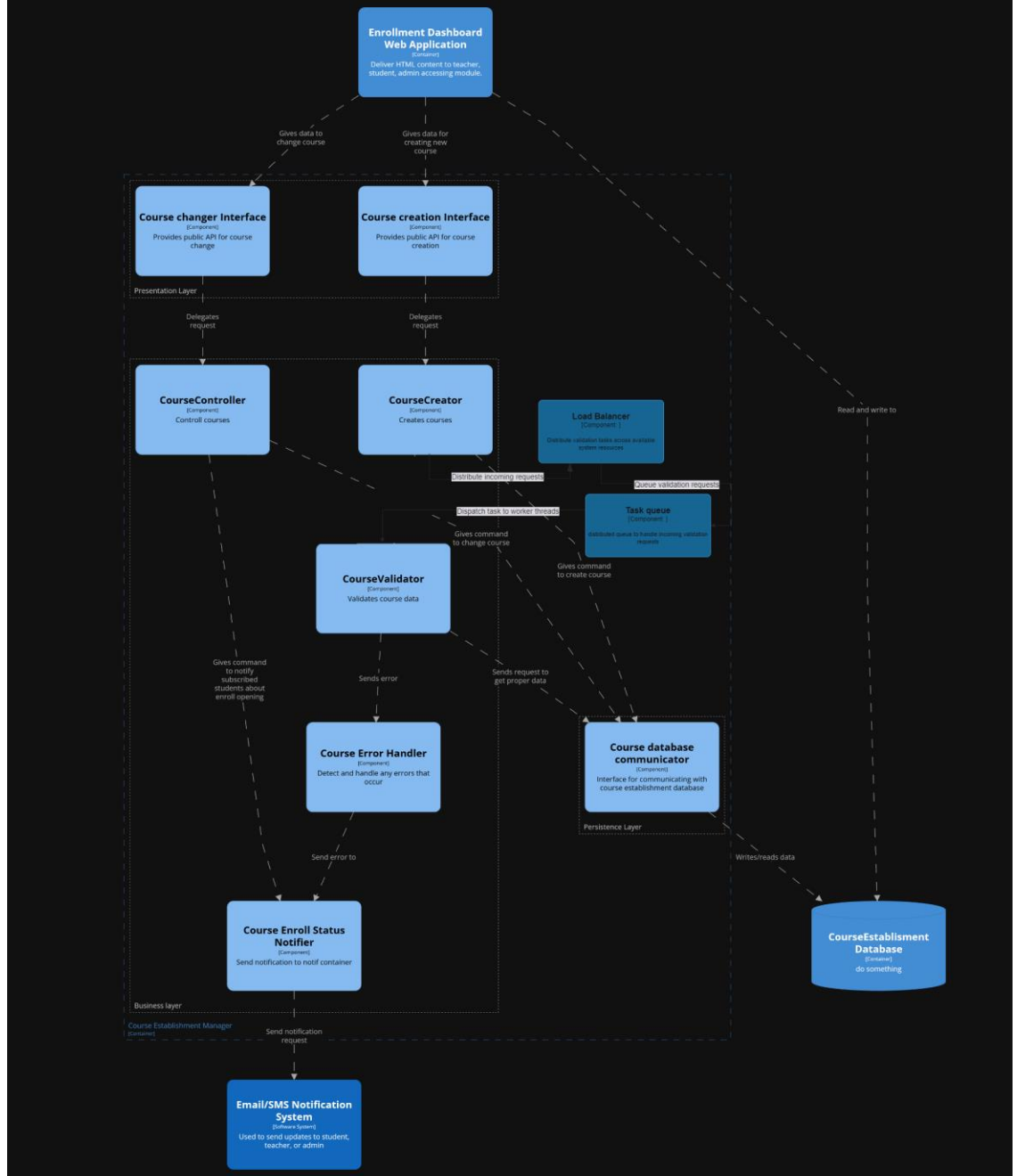
- **CourseValidator**



Response:

- Requests are queued using a **task queue.**
- The **CourseValidator** processes the requests in parallel using a pool of worker threads.
- Users are notified of their request status (e.g., queued or processed).





[Component] Enrollment System - Course Establishment Manager

неділя, 10 листопада 2024 р. о 23:51:1 за центральноєвропейським стандартним часом

Response Measure:

- Throughput is maintained at a consistent rate (e.g., 100 requests per second).

- All requests are processed within a specified timeframe (e.g., 60 seconds).

REPOSITORY LINK

https://github.com/Metalystn/Exam-repository

THANK YOU!