IOWA STATE UNIVERSITY

Senior Design Poster Presentation
Fall 2021

Department of Software Engineering

SDDEC21-19

Team Members: Andrew Dort, Emma Paskey, Joshua Slagle, Kira Pierce, Zach Borchard
Advisor: Nicholas Fila
Client: Henry Duwe

Canvas LTI Student Climate Dashboard

Introduction & Motivation

Problem statement:

Instructors currently create journey maps to chart student resonance with the goal of understanding and identifying shared experiences among students within a course. Gauging student resonance can help instructors when planning for subsequent semesters. However, gathering data and building a graph is an intensive, subjective, and time consuming process.

Solution:

Provide instructors with a software tool to automate the most time-consuming aspects of the journey-mapping process (data gathering & visualization). Users are provided with an interactive charting tool, and have the ability to aggregate student feedback and statistics from Canvas. Users can also customize how this data is interpreted and displayed.

High Level Design Requirements

Functional

- The system should be able to create Journey Map from data.
- The system should automatically categorize the students into similar groups.
- Professor should be able to view class Journey Map.
- Professors should be able to view journey maps with only a specific set of variables taken into account.

Non-Functional

- Data integration should be modular for future extensions.
- Student data should not be accessible by other students.
- The system should be easily extensible.

Operating Environment

- Services are containerized for universal* compatibility
- Kubernetes handles networking / service availability.

Design Approach

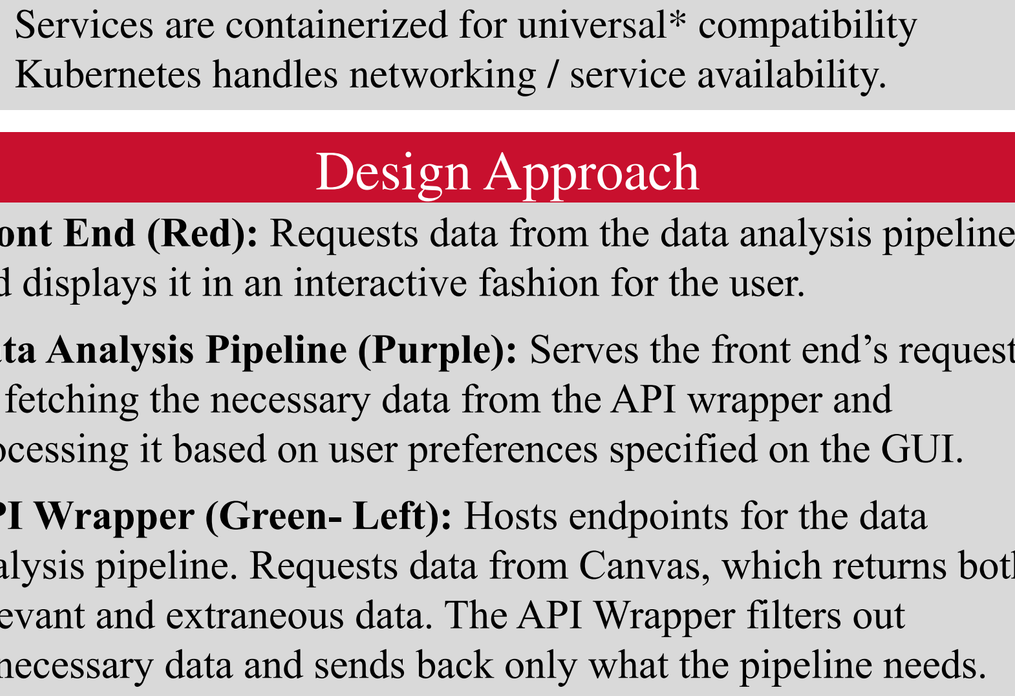
Front End (Red): Requests data from the data analysis pipeline and displays it in an interactive fashion for the user.

Data Analysis Pipeline (Purple): Serves the front end's requests by fetching the necessary data from the API wrapper and processing it based on user preferences specified on the GUI.

API Wrapper (Green- Left): Hosts endpoints for the data analysis pipeline. Requests data from Canvas, which returns both relevant and extraneous data. The API Wrapper filters out unnecessary data and sends back only what the pipeline needs.

Data Storage (Green - Right): Data caching on API endpoints for speed / minimizing API calls to Canvas as well as storage and endpoints for all non-canvas data that the application needs to run.

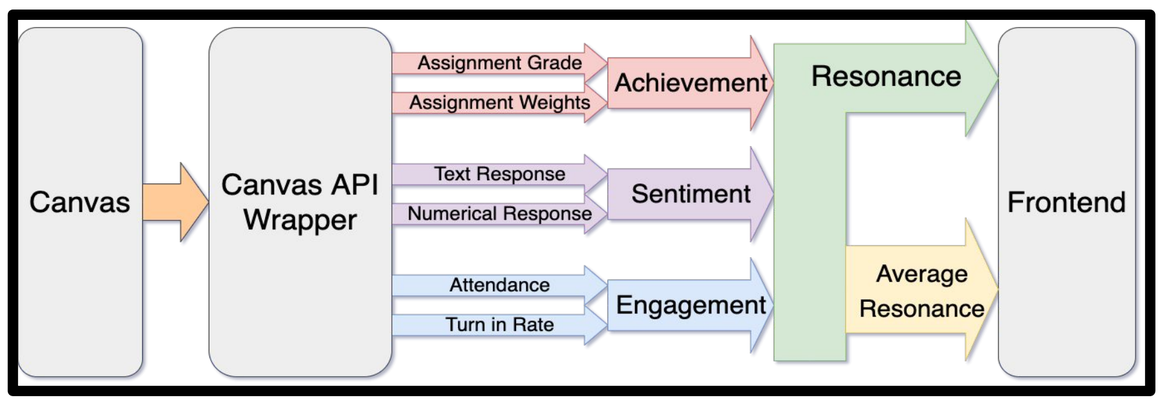
Concept Sketches



Testing & Standards

- **Testing**
  - Full integration testing between all tools. The Data analysis pipeline , frontend-ui, and the .NET core canvas API wrapper.
  - Using HTTP request tools such as Postman and Insomnia on all endpoints for our application.
- **Standards**
  - Storage of sensitive tokens in Kubernetes Secrets.
  - Pagination & Caching for data-heavy API Endpoints.
  - RFC HTTP Standardized Response Status Codes

Use Case Flow

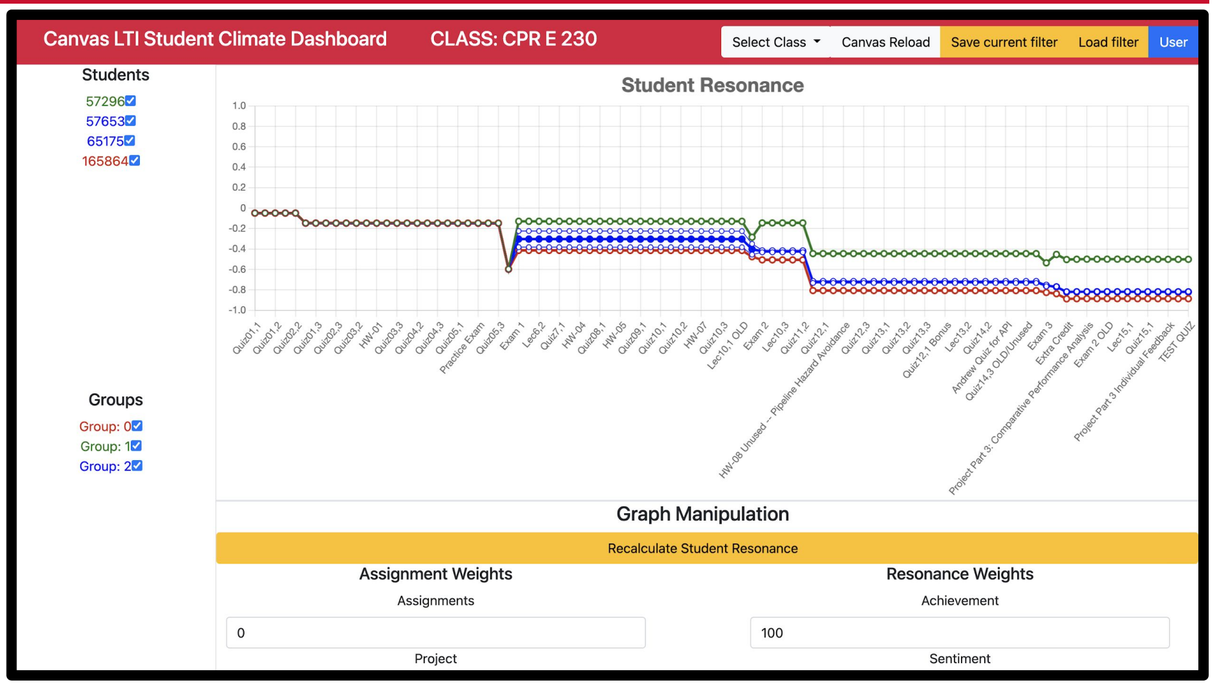


Intended Users & Uses

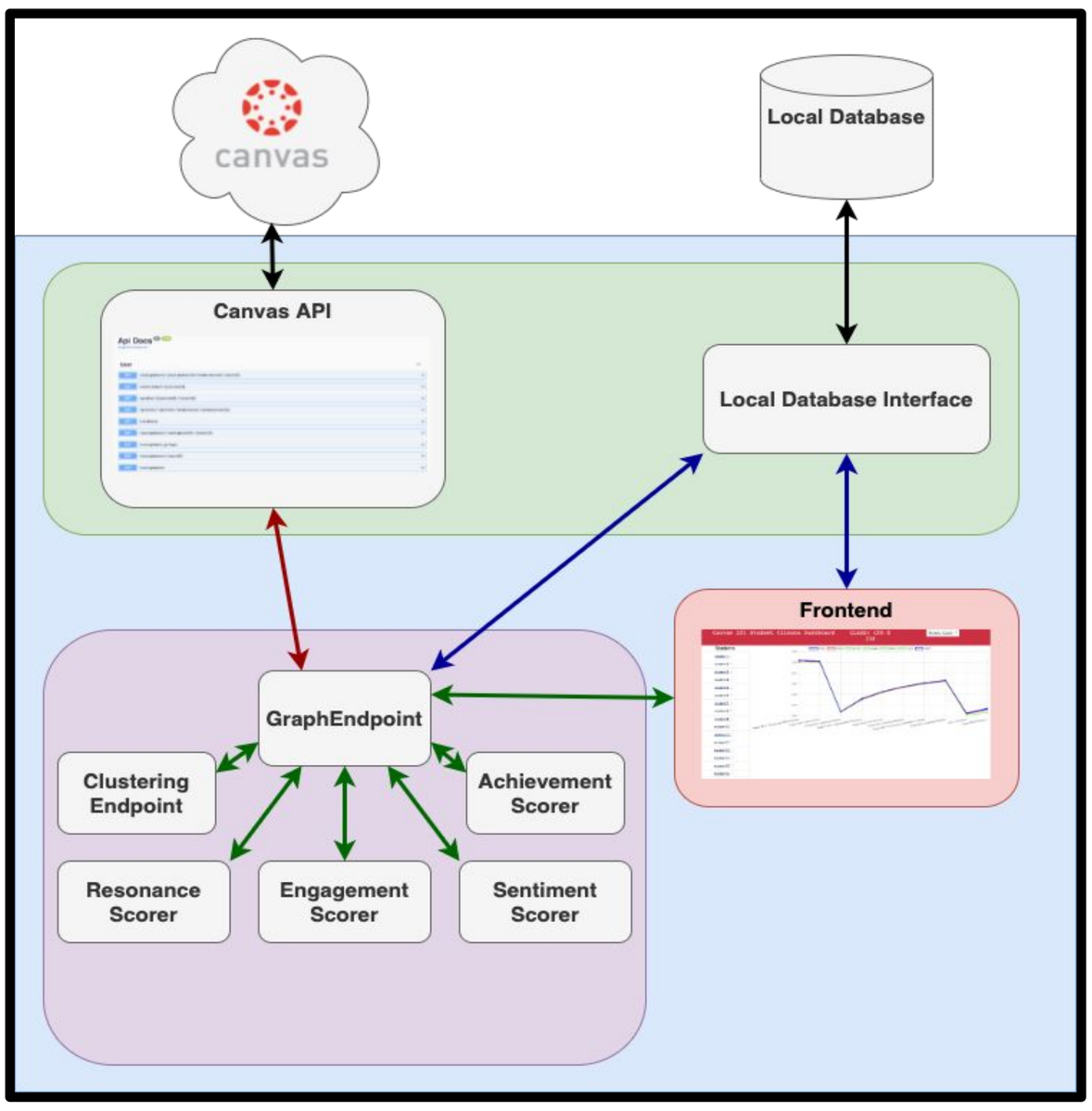
Intended Users: Course Instructors and Co-Instructors
Intended Uses:

- Chart student resonance as course progresses.
- Find common points of student feedback and course experience (student resonance).
- Use findings to shape approach to course planning.

Product Main Interface Screen Shots



System Diagram



Technical Details

- **Application Infrastructure**:
  - Containerized Microservices
  - Kubernetes Cluster Container Management
  - MySQL Database
- **Technologies Leveraged:**
  - Docker
  - Kubernetes
  - .NET Core
  - Flask
  - Servicestack
- **Communication:**
  - Protobufs (Green)
  - JSON (Red)
  - TCP/IP (Blue)
- **Programming Languages**
  - Python
  - C#
  - Javascript
- **Resources Used**
  - Virtual Machine from ETG
  - Mock Canvas Course