

Реализация элементов, расширяющих функциональные возможности HwProj 2.0

Усик Владислав Сергеевич, 18.Б11-ММ

Научный руководитель: доц. каф. СП, к.т.н. Ю.В. Литвинов

СПБГУ, 2021

Введение

- В университете большой поток информации между студентами и преподавателями
- Есть сложный бизнес-процесс проведения практик
- Хорошо бы автоматизировать

HwProj

- Имеет ошибки и недочеты
- Разработан на устаревших технологиях
- Практическая невозможность поддержки и расширения

Аналоги

- HwProj
- BlackBoard
- Stepik
- EasyChair
- Moodle

Постановка задачи

Цель работы — разработка сервиса практик и его подключение к HwProj 2.0

Задачи:

- Изучение предметной области и требований
- Разработка сервиса практик
- Реализация общения между сервисами
- Подключение авторизации через существующий сервис
- Реализация алгоритма для сервиса практик, автоматически распределяющего рецензентов на практики

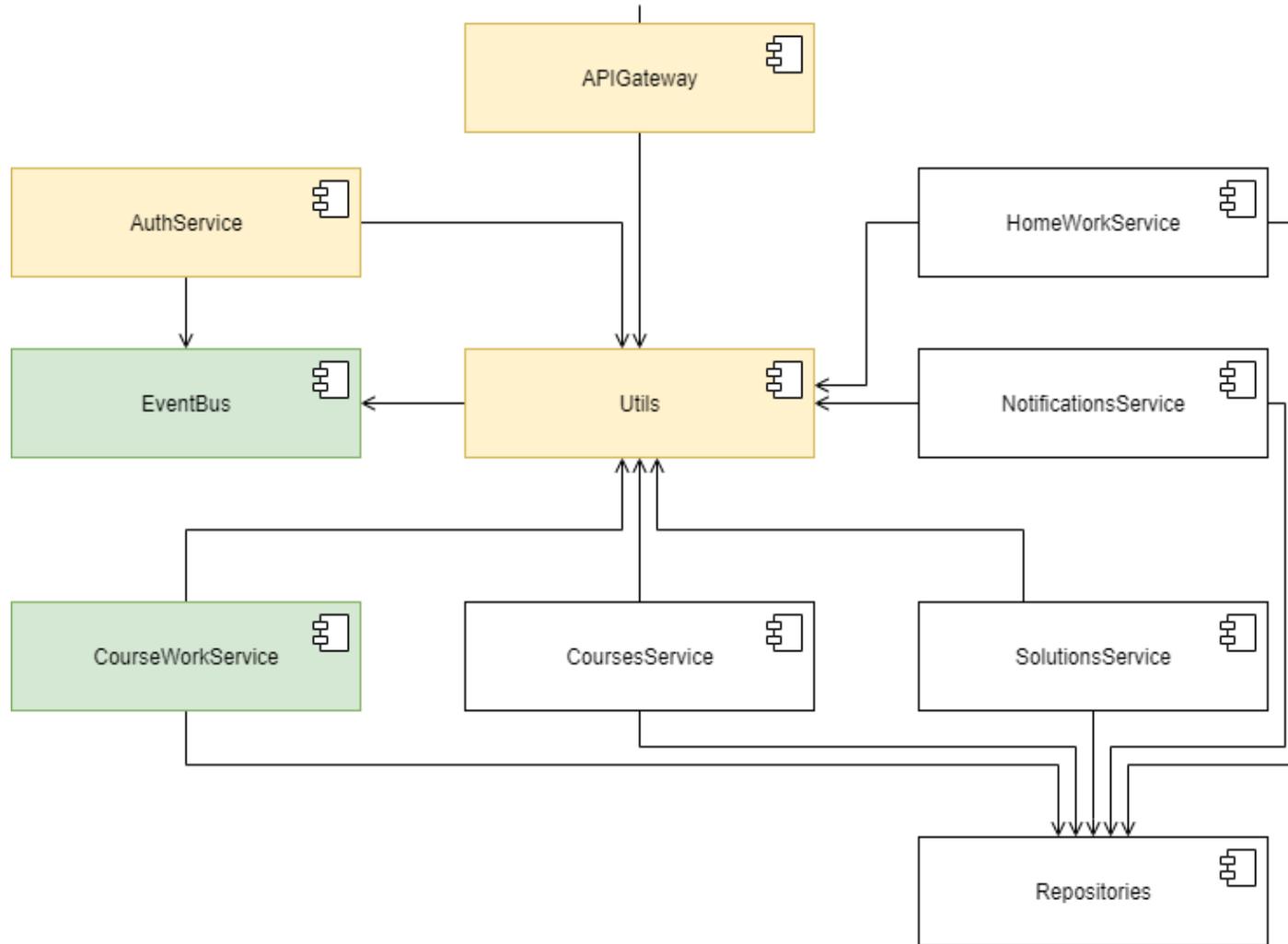
Бизнес-требования

- CRUD-операции над практиками и запросы различных выборок практик
- Прием/подача заявок
- Учет дедлайнов
- Хранение файлов
- Система ставок для выбора рецензентов
- Автоматическое распределение рецензентов с возможностью ручной корректировки

Технологии

- ASP.Net Core
- Entity Framework Core
- RabbitMQ
- Ocelot

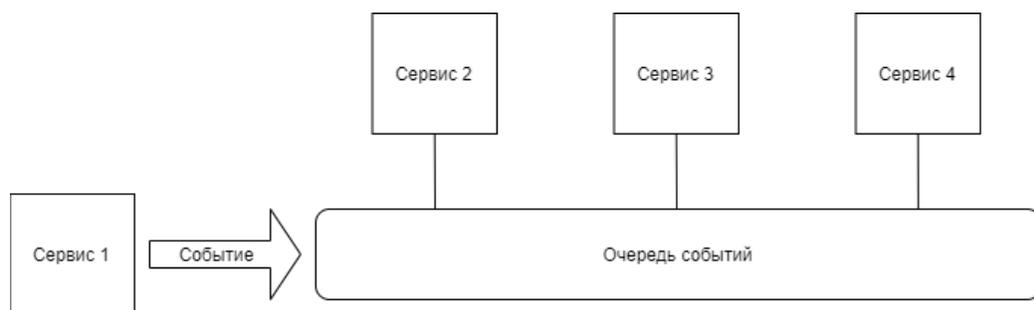
Архитектура приложения



Реализация

- Примерное количество реализованных запросов — 50
- Вся бизнес-логика обработки запросов находится в классах сервисов
- Логика работы с данными находится в классах репозиториев
- Значимые классы микросервиса практик: CourseWorkService, ApplicationService, UniversityService, ReviewService, UserRepository

Взаимодействие сервисов



- Используется очередь сообщений (RabbitMQ)
- Микросервисы используют модель “Издатель - Подписчик”
- Микросервисы подписываются на определенные события при запуске

Подключение авторизации

- Существует реализованный ранее микросервис авторизации
- Требуется поддерживать авторизацию любых пользователей
- Информация о пользователях в сервисе практик и в сервисе авторизации HwProj 2.0 отличается
- Поэтому сервис практик имеет собственные данные о пользователях в базе данных
- Пересекающиеся данные пользователей сервисов синхронизируются с помощью очереди событий

Алгоритм распределения рецензентов

- По результатам биддинга имеем ставки рецензентов на практики
- В качестве основы алгоритма используем венгерский метод для решения задач о назначении
- В качестве матрицы в методе используем ставки
- Добавляем дополнительные переменные к ставкам в матрице, чтобы учитывать количество работ на одного рецензента

Итоги

- Изучены предметная область и требования
- Разработан сервис практик
- Реализовано общение между сервисами
- Подключена авторизация через существующий сервис
- Реализован алгоритм распределения рецензентов