**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой ИУ7

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.В.Рудаков

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

**ЗАДАНИЕ**

**на выполнение курсового проекта**

по дисциплине Распределенные системы обработки информации

Студент группы ИУ7-23М

Горячев Виктор Геннадьевич

(Фамилия, имя, отчество)

Тема курсового проекта Разработка приложения для ведения журнала расходов и заметок

Направленность КП: Учебный

Источник тематики: кафедра

График выполнения проекта: 25% к 4 нед., 50% к 7 нед., 75% к 11 нед., 100% к 14 нед.

***Задание:***

Разработать прототип системы ведения заметок, задач и журнала расходов на базе веб-интерфейса. Система должна состоять из микросервисов, каждый из которых отвечает за свою задачу: сервис пользовательского интерфейса; сервис авторизации и данных пользовательских аккаунтов; сервис журнала расходов; сервис заметок; сервис задач; сервис статистики. Каждый сервис при необходимости может иметь доступ к связанной с ним базе данных, но не должен иметь доступа к базам данных других сервисов. Все запросы между сервисами требуют авторизацию. Запросы пользователей делятся на две категории: запросы, требующие авторизации пользователя, и запросы, доступные для всех пользователей. Все ошибки должны обрабатываться; в случае недоступности некритичного функционала должна осуществляться деградация функциональности. Все действия на сервисах должны логироваться. Все сервисы должны собираться и разворачиваться через CI/CD.

Требования к программной реализации:

1. Создать сервис, выполняющий функцию Identity Provider, реализовать протокол OpenID Connect. Реализовать scope:
   * openid – всегда передается, в ответ возвращается JWT;
   * profile – базовая информация о пользователе;
   * email – email пользователя.
2. Для получения токена на Identity Provider реализовать Authorization Flow. UI рассматривается как 3rd party application и авторизация пользователя так же выполняется через Authorization Flow.
3. Все методы /api/\*\* (кроме /api/v1/authorize и /api/v1/callback) на всех сервисах закрыть token-based авторизацией. 4Так же реализовать ролевую модель:
   * руками создать пользователя с ролью Admin;
   * для зарегистрированных пользователей по-умолчанию роль User.
4. На Identity Provider сделать возможность создания новых пользователей (запрос от пользователя с ролью Admin).
5. Каждый сервис при получении запроса выполняет валидацию [JWT](https://jwt.io/introduction) токена с помощью [JWKs](https://auth0.com/docs/security/tokens/json-web-tokens/json-web-key-sets), которые он получает из Identity Provider (запрос к Identity Provider делать не нужно).
6. JWT токен пробрасывать между сервисами, при получении запроса валидацию токена так же реализовать через JWKs.
7. Убрать заголовок X-User-Name и получать пользователя из JWT-токена.
8. Если авторизация некорректная (отсутствие токена, ошибка валидации JWT токена, закончилось время жизни токена (поле exp в payload)), то отдавать 401 ошибку.
9. Для функционала, реализованного в рамках лабораторных работ, реализовать пользовательский интерфейс как Single Page Application ([React](https://reactjs.org/), [Angular](https://angular.io/), [Vue](https://vuejs.org/)) или мобильный клиент. Использование CSS обязательно.
10. Выделить сервис статистики: в него отправлять через Kafka информацию о выполненных действиях. В зависимости от задания по пришедшим данным строить отчет, доступ к которому должен быть только у пользователя с ролью Admin.
11. Как и в [ЛР #4](https://github.com/bmstu-rsoi/lab4-template) нужно развернуть Managed Kubernetes Cluster и настроить Ingress Controller (для публикации сервисов наружу можно использовать *только* Ingress).
12. Для деплоя использовать [helm charts](https://helm.sh/docs/topics/charts/), они должны быть универсальным для всех сервисов и отличаться лишь набором параметров запуска.

***Оформление курсового проекта:***

Расчётно-пояснительная записка на 20-25 листах формата А4. Расчётно-пояснительная записка должна содержать постановку задачи, введение, аналитическую часть, конструкторскую часть, технологическую часть, заключение, список литературы, приложения.

На защиту проекта должна быть представлена презентация, состоящая из 10-15 слайдов. На слайдах должны быть отражены: постановка задачи, использованные методы и алгоритмы, структура комплекса программ, интерфейс, характеристики разработанного ПО.

Дата выдачи задания « \_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**Руководитель курсового проекта** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Щетинин Г.А.

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Горячев В. Г.

(Подпись, дата) (И.О.Фамилия)