## Доклад по теме: Веб-приложение для управления групповыми мероприятиями с функциями разделения бюджета и отслеживания задач

## Введение

В современном динамичном мире организация групповых мероприятий требует эффективных инструментов координации. Существующие решения часто не обеспечивают единого подхода к управлению мероприятиями, финансами и задачами. Разработанная система EventHub позволяет решить эту проблему, объединив функции планирования мероприятий, распределения бюджета и контроля выполнения задач в одной платформе с микросервисной архитектурой.

## Цель и задачи работы

Целью работы является создание веб-приложения для эффективного управления групповыми мероприятиями. Для достижения этой цели были поставлены следующие задачи:

1. разработать микросервисную архитектуру, обеспечивающую масштабируемость и независимость компонентов;
2. реализовать основные функциональные модули: управление мероприятиями, распределение бюджета, отслеживание задач;
3. обеспечить надежную систему оповещений через email;
4. создать удобный пользовательский интерфейс с современными веб-технологиями.

## Основная часть разработки

### 1. Исследовательский раздел

**Анализ предметной области:**

Проведен анализ существующих решений, таких как Asana, Trello и Splitwise. Выявлено, что каждое из них фокусируется только на одном аспекте организации мероприятий: либо на управлении событиями, либо на задачах, либо на разделении расходов. Ни одна платформа не предлагает комплексного решения.

**Определение требований к системе:**

Сформулированы ключевые требования: создание и управление мероприятиями, приглашение участников, планирование расходов, распределение задач, отслеживание их выполнения, система уведомлений.

### 2. Проектирование и реализация

**Проектирование архитектуры системы:**

Система реализована на основе микросервисной архитектуры, включающей следующие компоненты:

* **Core Service:** центральный микросервис, отвечающий за управление мероприятиями, их создание, редактирование и доступ участников;
* **API Gateway:** обеспечивает единую точку входа для клиентских запросов и маршрутизацию к нужным микросервисам;
* **Communication Service:** управляет взаимодействием между участниками, групповыми чатами и комментариями;
* **Notification Service:** отвечает за отправку уведомлений о важных событиях через email;
* **EMS UI Service:** веб-интерфейс на основе Vite, обеспечивающий взаимодействие пользователей с системой.

**Технологический стек:**

**Frontend:** React с использованием Vite для быстрой сборки и разработки

**Backend:** Golang для микросервисов

**Базы данных:** PostgreSQL для хранения структурированных данных

**Кэширование:** Redis для повышения производительности

**Очереди сообщений:** RabbitMQ для асинхронного взаимодействия между микросервисами

**Уведомления:** MailHog для тестирования email-уведомлений

**Контейнеризация:** Docker и Docker Compose для упаковки и развертывания

### 3. Технологический раздел

**Core Service**

Основной микросервис системы, реализующий:

* управление мероприятиями (создание, редактирование, удаление);
* управление бюджетом мероприятий;
* распределение расходов между участниками;
* отслеживание задач и их статусов;
* управление участниками мероприятий.

**API Gateway**

Обеспечивает:

* единую точку входа для клиентских запросов;
* маршрутизацию запросов к соответствующим микросервисам;
* аутентификацию и авторизацию пользователей;
* кэширование запросов для повышения производительности.

**Communication Service**

Управляет коммуникацией:

* групповые чаты для каждого мероприятия;
* комментирование задач и расходов;

**Notification Service**

Обеспечивает информирование пользователей:

* отправка email-уведомлений о важных событиях;
* напоминания о приближающихся мероприятиях;
* уведомления о новых задачах и изменениях в бюджете;
* интеграция с RabbitMQ для асинхронной обработки уведомлений.

**UI Service**

Веб-интерфейс системы:

* Реактивный интерфейс на основе современных фреймворков
* Адаптивный дизайн для работы на различных устройствах
* Интерактивные элементы для управления задачами
* Визуализация бюджета и расходов
* Интеграция с календарями

**Модуль разделения бюджета:**

В Core Service реализована функциональность управления бюджетом:

* разделение расходов между участниками по разным схемам;
* отслеживание оплат и задолженностей;
* генерация отчетов по расходам.

**Модуль управления задачами:**

Также в Core Service реализована система отслеживания задач:

* создание задач с дедлайнами и приоритетами;
* назначение ответственных и соисполнителей;
* отслеживание статуса выполнения.

**Инфраструктура системы:**

* docker контейнеры для каждого микросервиса;
* независимое масштабирование компонентов;
* health checks для контроля доступности сервисов;
* отказоустойчивость благодаря политикам перезапуска;
* использование общих сетей для коммуникации между сервисами.

**Тестирование системы:**

* модульное тестирование для управления расходами и задачами;
* интеграционное тестирование взаимодействия микросервисов;
* ручное UI-тестирование.

## Заключение

Разработанная система EventHub представляет собой комплексное решение для управления групповыми мероприятиями, объединяющее функции планирования, распределения бюджета и отслеживания задач. Микросервисная архитектура обеспечивает гибкость, масштабируемость и устойчивость системы.

Основные преимущества системы:

* единая платформа для всех аспектов организации мероприятий;
* прозрачное распределение расходов между участниками;
* эффективное отслеживание задач и ответственности;
* оперативное информирование через систему уведомлений;
* современный и интуитивно понятный интерфейс.

Система успешно прошла тестирование и готова к внедрению для использования в реальных условиях организации групповых мероприятий различного масштаба.