

Отчет по 1 этапу курсовой работы

Платформа для организации мероприятий и встреч

Выполнили:

Боринский Игорь Дмитриевич
и Болорболд Аригуун

Преподаватель:

Харитоновна Анастасия Евгеньевна

Санкт-Петербург
2024 год

Содержание

1	Описание предметной области	2
2	Цели информационной системы и задачи	2
2.1	Зачем нужна система	2
2.2	Задачи, которые решает система	2
3	Функциональные требования (Functional Requirements)	2
4	Требования к удобству использования (Usability Requirements)	3
5	Требования к надежности (Reliability Requirements)	4
6	Требования к производительности (Performance Requirements)	4
7	Требования к поддерживаемости (Supportability Requirements)	4
7.1	Требования к логированию	4
8	Дополнительные требования (+)	6
8.1	Архитектурные требования	6
8.2	Требования к реализации	6
8.3	Требования к интерфейсу	6
8.4	Физические требования	6
9	Прецеденты	7
10	Архитектура системы	10
10.1	Общая архитектура системы	10
10.2	Технологии и фреймворки	10
10.3	Deployment Diagram (Диаграмма развертывания)	11

1 Описание предметной области

Платформа представляет собой информационную систему, которая позволяет пользователям создавать и организовывать мероприятия и встречи. Пользователи могут находить интересные события, регистрироваться на них, создавать свои собственные встречи и управлять ими. Организаторы могут добавлять описание, даты, время, место, и управлять участниками.

2 Цели информационной системы и задачи

2.1 Зачем нужна система

- Упрощает процесс организации и проведения мероприятий.
- Помогает пользователям находить интересные события и мероприятия в своем городе или онлайн.
- Предоставляет инструменты для управления участниками, платежами, расписанием и отзывами.

2.2 Задачи, которые решает система

- Организация встреч, семинаров, мастер-классов, воркшопов и других мероприятий.
- Сбор и управление регистрациями участников.
- Поиск и фильтрация мероприятий по категориям, дате и местоположению.
- Управление списком участников и возможность отправки уведомлений.
- Оставление отзывов и оценок после завершения мероприятия.

3 Функциональные требования (Functional Requirements)

- **FR1:** Система должна предоставлять возможность пользователю зарегистрироваться, используя электронную почту и пароль, где длина пароля должна быть не менее 4 символов.
- **FR2:** Система должна позволять пользователю входить в систему, используя зарегистрированный email и пароль.
- **FR3:** Система должна позволять авторизованным пользователям создавать мероприятие, указывая его название (до 100 символов), описание (до 500 символов), дату, время, место проведения, максимальное количество участников (от 1 до 500) и категорию (выбор из списка).
- **FR4:** Система должна предоставлять пользователям возможность просматривать список всех доступных мероприятий, отображая не более 10 мероприятий на одной странице (с пагинацией).

- **FR5:** Система должна позволять пользователям фильтровать мероприятия по дате, категории (конференция, мастер-класс, лекция, семинар, воркшоп) и местоположению (с использованием указания города).
- **FR6:** Система должна позволять пользователям регистрироваться на выбранное мероприятие с подтверждением через электронную почту.
- **FR7:** Система должна позволять организаторам мероприятия редактировать, удалять или просматривать список зарегистрированных участников для своих мероприятий.
- **FR8:** Система должна автоматически отправлять электронное письмо-напоминание всем зарегистрированным участникам за 24 часа до начала мероприятия.
- **FR9:** Система должна позволять зарегистрированным участникам оставить отзыв и оценку от 1 до 5 после завершения мероприятия.
- **FR10:** Система должна позволять организаторам добавлять доступную еду для мероприятий, включая такие параметры, как название, описание, и категорию (вегетарианское, безглютеновое и т.д.).
- **FR11:** Система должна отображать список доступной еды на странице мероприятия для всех зарегистрированных участников. (вегетарианское, безглютеновое и т.д.).
- **FR12:** Система должна позволять организатору редактировать или удалять информацию о еде в любое время до начала мероприятия.

4 Требования к удобству использования (Usability Requirements)

- **U1:** Система поддерживать разрешения экрана:
 - Мобильные устройства: от 360x640 до 414x896 пикселей.
 - Планшеты: от 768x1024 до 1280x800 пикселей.
 - Десктопы: от 1366x768 до 1920x1080 пикселей.
- **U2:** Система должна обеспечивать время загрузки любой страницы системы не должно превышать 3 секунд при стандартной скорости интернет-соединения (40 Мбит/с).
- **U3:** Система должна отображать всплывающие подсказки при наведении на иконки в течение 0.5 секунд, содержащие краткое описание действия.
- **U4:** Система должна обеспечивать отображение информации о еде в удобном и структурированном виде на странице мероприятия (с указанием категорий и краткого описания каждого блюда).

5 Требования к надежности (Reliability Requirements)

- **R1:** Система должна фиксировать все транзакции в журнале событий и хранить их в течение одного месяца. Логи должны быть доступны для просмотра разработчиками и автоматически удаляться по истечении этого срока.
- **R2:** Система должна отправлять оповещение разработчикам о любых сбоях или ошибках в работе в течение 1 минуты после их возникновения.
- **R3:** Система должна обеспечивать хэширование паролей пользователей с использованием алгоритма BCrypt с минимальным количеством итераций (work factor) не менее 10.

6 Требования к производительности (Performance Requirements)

- **P1:** Система должна обрабатывать запросы на просмотр списка мероприятий (до 20 записей) за время не более 2 секунд при одновременном использовании системы не более чем 50 авторизованными пользователями.
- **P2:** Система должна обеспечивать возможность одновременной работы не менее 500 пользователей, при этом среднее время отклика на выполнение любых операций (просмотр списка мероприятий, регистрация на мероприятие, создание мероприятия) не должно превышать 2 секунд при нагрузке 500 пользователей.
- **P3:** Система должна обеспечивать отклик на операции добавления, редактирования и удаления мероприятий в течение 2 секунд.
- **P4:** Система должна быть способна обрабатывать пиковую нагрузку в 150 запросов в секунду без потери производительности.

7 Требования к поддерживаемости (Supportability Requirements)

7.1 Требования к логированию

S1.1 Система должна осуществлять логирование на уровне каждого сервиса с использованием следующих уровней логирования:

- **INFO:** для записи общей информации о работе сервиса.
- **WARN:** для указания потенциальных проблем, которые могут привести к сбоям.
- **ERROR:** для записи ошибок, которые приводят к фактическим сбоям системы.

S1.2 Система должна осуществлять структурированное логирование в формате JSON при записи логов, используя встроенные возможности логирования Spring Boot (Logback).

S1.3 Система должна использовать библиотеку Logbook для логирования входящих HTTP-запросов и ответов, включая следующие детали:

- Путь запроса (URL)
- Метод запроса (GET, POST, PUT, DELETE)
- Время выполнения запроса
- Код ответа

S1.4 Система должна сохранять логи в текстовые файлы на сервере в формате JSON для упрощенного анализа и обеспечивать их хранение не менее 30 дней.

- **S2:** Система должна предоставлять RESTful API с версионностью (/api/v*/), чтобы обеспечить обратную совместимость при будущем обновлении функционала.

8 Дополнительные требования (+)

8.1 Архитектурные требования

- **A1:** Система должна быть разработана с использованием архитектурного паттерна MVC (Model-View-Controller).
- **A2:** Система должна использовать базу данных PostgreSQL версии 14.0 или выше.
- **A3:** Система должна хэшировать все пароли пользователей с использованием алгоритма BCrypt с минимальной длиной соли в 12 символов.
- **A4:** Система должна использовать Spring Boot версии 3.3.4.
- **A5:** Система должна использовать ORM Hibernate для взаимодействия с базой данных, обеспечивая автоматическое маппирование сущностей.

8.2 Требования к реализации

- **I1:** Система должна быть написана на Java версии 17.
- **I2:** Сборка проекта должна осуществляться с использованием Maven версии 3.8 или выше.
- **I3:** Документация API должна быть автоматически сгенерирована с использованием Swagger/OpenAPI и доступна по URL /swagger-ui.html.

8.3 Требования к интерфейсу

- **UI1:** Система должна корректно отображаться в браузерах Google Chrome, Mozilla Firefox, Safari последних двух версий.
- **UI2:** Система должна отображать информацию о бесплатной еде на странице мероприятия в отдельном разделе с возможностью просмотра описания каждого блюда.

8.4 Физические требования

- **PH1:** Система должна быть развернута на сервере **helios**.

9 Прецеденты

Прецедент: Создание мероприятия	
ID:	1
Краткое описание:	Пользователь создает новое мероприятие.
Главные актёры:	Пользователь
Второстепенные актёры:	нет
Предусловия:	Пользователь должен быть авторизован в системе.
Основной поток:	<ol style="list-style-type: none">1. Пользователь выбирает опцию "Создать мероприятие".2. Система предлагает форму для заполнения информации о мероприятии.3. Пользователь вводит название, описание, дату, время и место проведения мероприятия.4. Пользователь указывает максимальное количество участников и категорию мероприятия.5. Пользователь нажимает "Сохранить".6. Система подтверждает создание мероприятия и отображает его в списке мероприятий.
Альтернативные потоки:	ALT1: Если пользователь не заполнил обязательные поля, система показывает сообщение об ошибке и предлагает заполнить недостающую информацию.
Постусловия:	Мероприятие успешно создано и доступно для просмотра другими пользователями.

Прецедент: Регистрация на мероприятие	
ID:	2
Краткое описание:	Пользователь регистрируется на выбранное мероприятие.
Главные актёры:	Пользователь
Второстепенные актёры:	Организатор мероприятия
Предусловия:	<ul style="list-style-type: none"> • Пользователь должен быть авторизован в системе. • Мероприятие должно быть доступно для регистрации.
Основной поток:	<ol style="list-style-type: none"> 1. Пользователь выбирает мероприятие из списка. 2. Пользователь нажимает кнопку "Регистрация". 3. Система проверяет, что количество участников не превышено. 4. Система отображает сообщение о подтверждении регистрации.
Альтернативные потоки:	ALT1: Если количество участников превышено, система показывает сообщение об ошибке "Регистрация закрыта, максимальное количество участников достигнуто."
Постусловия:	Пользователь зарегистрирован на мероприятие, и его имя добавлено в список участников.

Прецедент: Оставление отзыва о мероприятии	
ID:	3
Краткое описание:	Участник мероприятия оставляет отзыв и оценку.
Главные актёры:	Участник мероприятия
Второстепенные актёры:	Организатор мероприятия
Предусловия:	<ul style="list-style-type: none"> • Участник должен быть зарегистрирован на мероприятие. • Мероприятие должно быть завершено.
Основной поток:	<ol style="list-style-type: none"> 1. Участник переходит на страницу завершённого мероприятия. 2. Участник выбирает опцию "Оставить отзыв". 3. Участник вводит текст отзыва и выбирает оценку от 1 до 5. 4. Участник нажимает "Отправить". 5. Система подтверждает добавление отзыва.
Альтернативные потоки:	ALT1: Если участник уже оставил отзыв, система отображает сообщение "Вы уже оставили отзыв для этого мероприятия."
Постусловия:	Отзыв добавлен к мероприятию и доступен для просмотра другими пользователями.

Прецедент: Добавление информации о еде на мероприятии	
ID:	4
Краткое описание:	Организатор добавляет информацию о предоставляемой еде на мероприятии.
Главные актёры:	Организатор
Второстепенные актёры:	нет
Предусловия:	Организатор должен быть авторизован и иметь права на управление мероприятием.
Основной поток:	<ol style="list-style-type: none"> 1. Организатор переходит на страницу управления мероприятием. 2. Организатор выбирает опцию "Добавить еду". 3. Система предлагает форму для ввода названия, описания и категории еды. 4. Организатор заполняет форму и нажимает "Сохранить". 5. Система подтверждает добавление информации о еде и обновляет страницу мероприятия.
Альтернативные потоки:	ALT1: Если организатор не заполнил обязательные поля, система отображает сообщение об ошибке и предлагает заполнить недостающую информацию.
Постусловия:	Информация о еде добавлена к мероприятию и доступна для просмотра зарегистрированным участникам.

10 Архитектура системы

10.1 Общая архитектура системы

Мы будем использовать стандартную трехуровневую архитектуру (Three-Tier Architecture), которая состоит из:

- **Уровень представления (Frontend):** Отвечает за взаимодействие с пользователем и отображение данных.
- **Уровень бизнес-логики (Backend):** Отвечает за обработку бизнес-логики, проверку данных, управление сессиями пользователей и другие бизнес-процессы.
- **Уровень хранения данных (Database):** Отвечает за хранение и управление данными.

10.2 Технологии и фреймворки

- **Frontend:**
 - JavaScript ES6
 - React
 - Fetch API
- **Backend:**

- Spring Boot (Spring MVC)
- Spring Security
- Spring Data JPA
- Hibernate
- Logbook

- Database:

- PostgreSQL

10.3 Deployment Diagram (Диаграмма развертывания)

Ниже представлена базовая схема, иллюстрирующая взаимодействие компонентов системы:

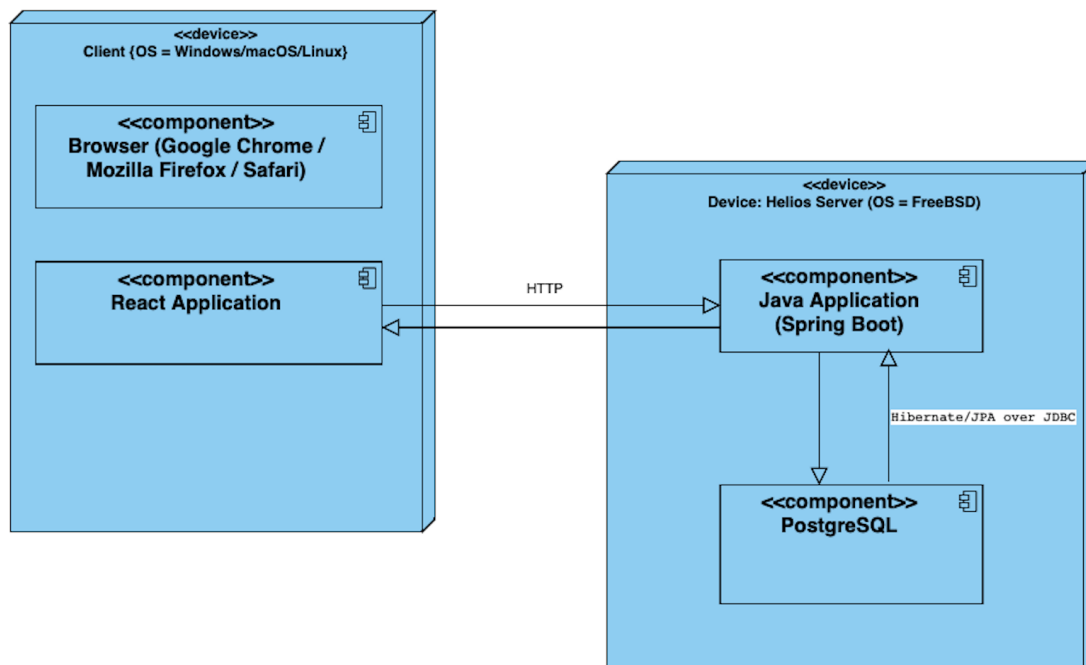


Рис. 1: Deployment Diagram