Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

ОТЧЁТ

по лабораторной работе №1

по дисциплине

ОБЪЕКТНОЕ МОДЕЛИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

| Выполнил | Робилко Т. М. гр. 2217011701 |
| --- | --- |
| Проверил | Зотов Н. В. Н.В. |

**Содержание**

[Этап 1. Текстовое описание объектной модели платформы для создания и управления мероприятиями 3](#_lobjl6il3fsj)

[Этап 2. Задача 1. BPMN-диаграмма бизнес-процесса 5](#_iwpcxg44uhwx)

[Этап 2. BPMN и ER-диаграмма для программной системы 6](#_lz4y0ro5b1ll)

[Этап 3. Use-Case, Sequence диаграммы 8](#_hx1sib5a2fjb)

[Этап 4. Базовая диаграмма классов и диаграммы классов вариантов развития 10](#_k5m0nf78se9s)

[Этап 5. Желание жить 11](#_f9fslcxm1s87)

# Этап 1. Текстовое описание объектной модели платформы для создания и управления мероприятиями

OneEvent - веб-приложение, предназначенное для агрегации информации о проводимых мероприятиях на различных площадках, возможности бронирования мест на эти мероприятия и возможности оставить отзыв о посещенных мероприятиях. Пользователями системы являются администраторы площадок, а также их посетители.

Бронирование билетов обеспечено интеграцией с существующими решениями (платформа ticketpro).

Система реализует авторизацию с использованием ролевой политики в уже готовом решении для авторизации (например, identity server 4 для ASP.NET систем).

Доступ групп пользователей к функционалу системы

В системе предусмотрено три роли: пользователь, администратор и менеджер.

Любой авторизованный человек обладает ролью пользователя. Ролью менеджера обладают представители разработчика. Роль администратора получают через обращение к менеджерам пользователи системы-заказчики. Для использования системы необходимо авторизоваться на главной странице (странице входа).

Режим пользователя предоставляет доступ к следующим подсистемам в виде различных вкладок, которые включают возможности:

1. Локации
2. Просмотр всех публично-доступных локаций в виде плиток. Для каждой локации показано название ближайшего публично-доступное мероприятие в виде гиперссылки на вкладку “Мероприятия”.
3. Фильтрация локаций по названию, геолокации.
4. При нажатии на локацию показывается всплывающее окно с детальной информацией: названием, фотографиями локации, а также ссылка на страницу мероприятий локации. Нижний элемент окна - модуль “список отзывов” для данной локации.
5. Мероприятия
6. Просмотр всех публично-доступных мероприятий в виде плиток. Для каждого мероприятия показаны название мероприятия, название локации (гиперссылка на вкладку “локации”), дата (диапазон дат), время проведения (диапазоны по дням)
7. Фильтрация мероприятий по названию, локации, дате.
8. При нажатии на мероприятие показывается окно с информацией: название, ссылка на страницу локации, дата и время проведения, кнопка перехода на страницу бронирования билетов. Нижний элемент окна - модуль “список отзывов”.
9. Личный кабинет
10. Вкладка “Мой профиль”: просмотр ФИО, даты рождения, электронной почты, и аватара с возможностью редактирования всех этих полей. Адрес электронной почты служит для идентификации аккаунта и не меняется. Внизу страницы кнопка “Выйти из системы”.
11. Вкладка “Мои отзывы”: просмотр в хронологическом порядке оставленных пользователем отзывов (вид отзыва приведен в описании модуля “список отзывов” ниже)

Режим администратора предоставляет доступ ко всему описанному выше функционалу, а также:

1. Локации
2. Галочка “показать только администрируемые мной”.
3. Всплывающее окно с деталями локации имеет кнопку “редактировать”, при нажатии на которую текстовые элементы превращаются в поля для ввода, а около списков фотографий и ссылок на сторонние сервисы появляются кнопки “добавить”, а также кнопки удаления имеющихся фотографий и ссылок. При нажатии на кнопку “редактировать” эта кнопка заменяется на кнопки “сохранить” и “отмена” для сохранения изменений локации и их отмены соответственно.
4. Мероприятия
5. Галочка “показать только администрируемые мной”.
6. Окно деталей мероприятия имеет кнопку “редактировать”, при нажатии на которую становится возможным изменить все его поля, в также появляется кнопка “Удалить мероприятие”
7. Кнопка “Создать мероприятие” в заголовке списка мероприятий. При нажатии на кнопку появляется всплывающее окно с деталями мероприятия, аналогичное описанному пунктом выше. При нажатии на кнопку “сохранить” заполненная информация о мероприятии сохраняется.
8. Модуль “Список отзывов”
9. Для каждого отзыва на локациях и мероприятиях локаций, которые может администрировать данный пользователь, имеется кнопка “удалить отзыв” для осуществления модерации.

Режим менеджера предоставляет доступ ко всему описанному выше функционалу, а также:

1. Локации
2. Окно деталей локации имеет кнопку “изменить администраторов”. При нажатии кнопки показывается всплывающее окно со списком администраторов данной локации. Напротив каждого пользователя имеется кнопка “удалить из администраторов”. В начале списка имеется поле для ввода адреса электронной почты и кнопка “добавить администратора”.
3. Окно деталей локации имеет кнопку “редактировать”. При нажатии на кнопку “редактировать” эта кнопка заменяется на кнопки “сохранить” и “отмена” для сохранения изменений локации и их отмены соответственно. Также при нажатии на кнопку “редактировать” появляется кнопка “Удалить локацию”.
4. Кнопка “Создать локацию”. При нажатии на кнопку появляется окно с деталями локации, чьи поля не заполнены и списки фотографий и ссылок пусты. При нажатии на кнопку “сохранить” указанные данные сохраняются.

Подсистема “Список отзывов”

Модуль представляет из себя список плашек, представляющих отзывы пользователей. Каждая плашка включает в себя: заголовок отзыва, имя и фамилию автора, текст содержания отзыва, а также оценку, данную по пятибалльной системе. В верхней части присутствует редактор для создания отзыва. При заполнении заголовка и содержания становится активной кнопка “Оставить отзыв” для публикации отзыва.

Направления развития

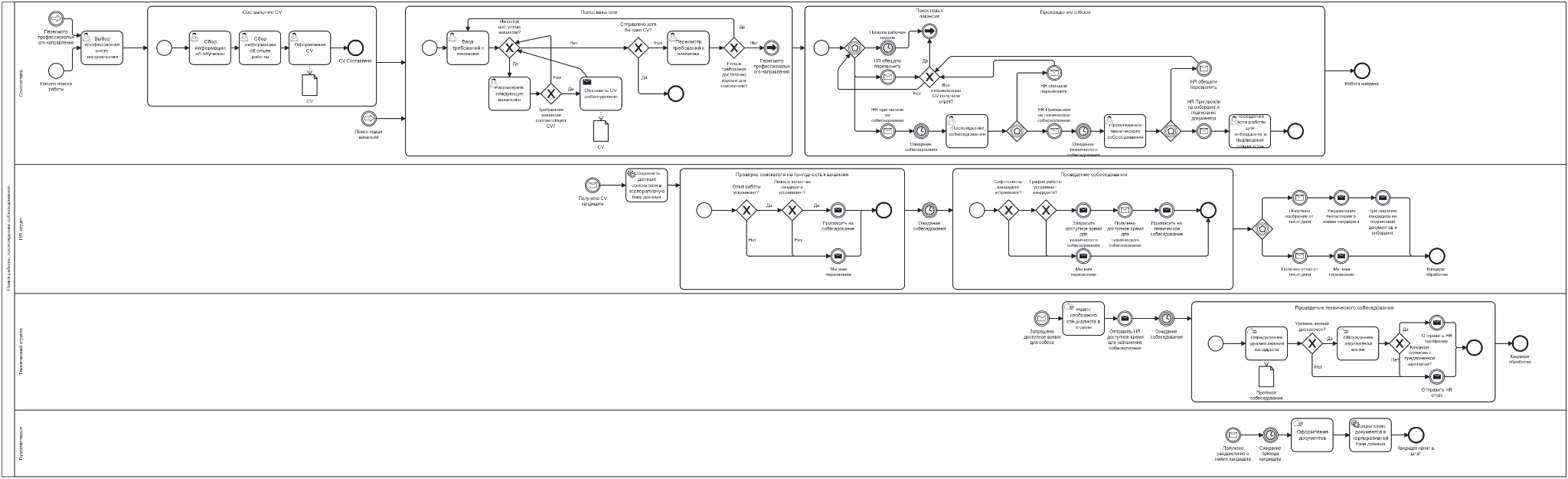
1. Цепочки обсуждений в подсистеме “Список отзывов”

2. Возможность создания локаций с доступом для конкретных групп пользователей (например, клубы по приглашению)

3. “Рецензии” помимо обычных отзывов для предоставления возможности авторитетным критикам оставить свое мнение о мероприятии.

# Этап 2. Задача 1. BPMN-диаграмма бизнес-процесса

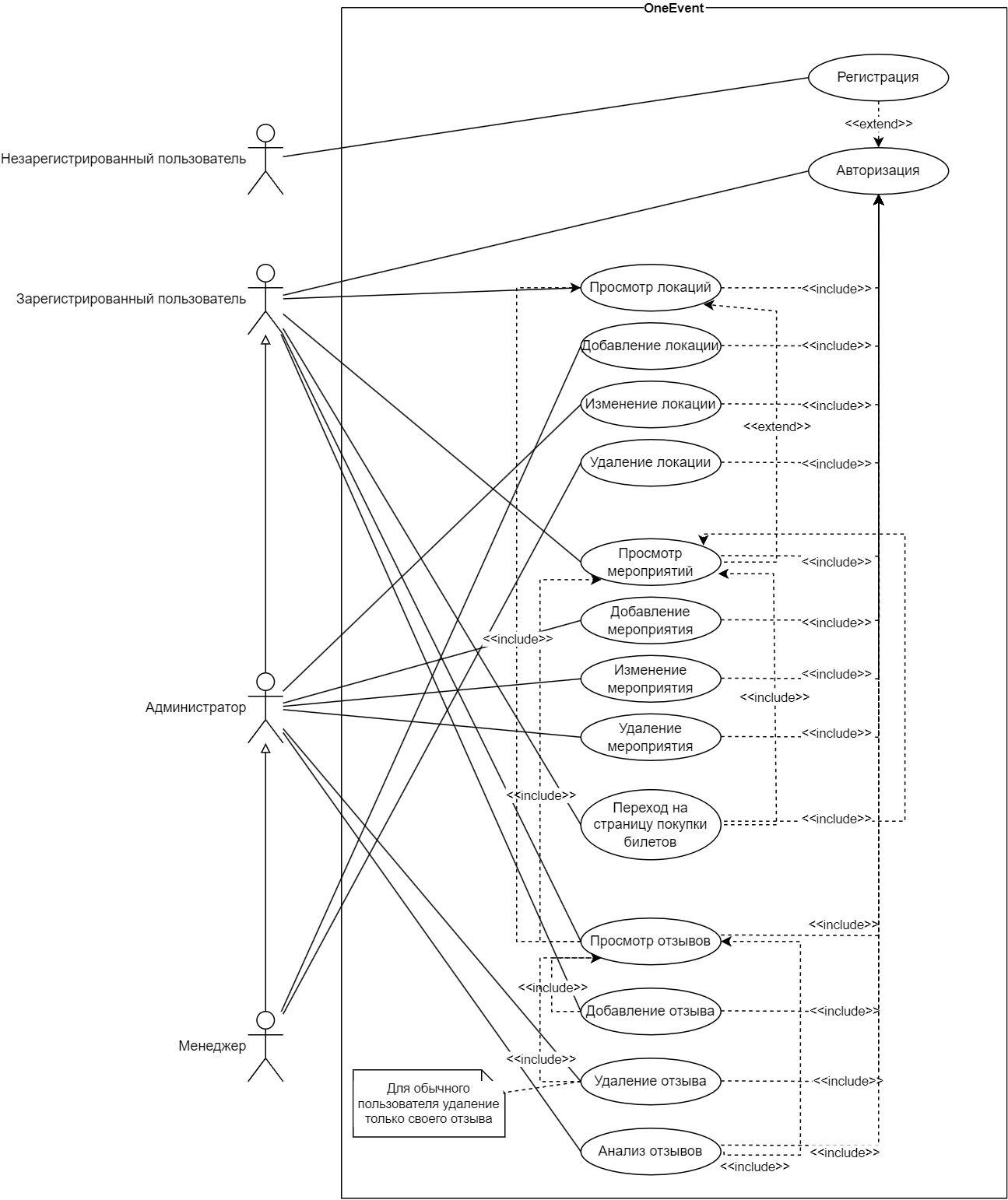
Вариант 18: Поиск работы, прохождение собеседований

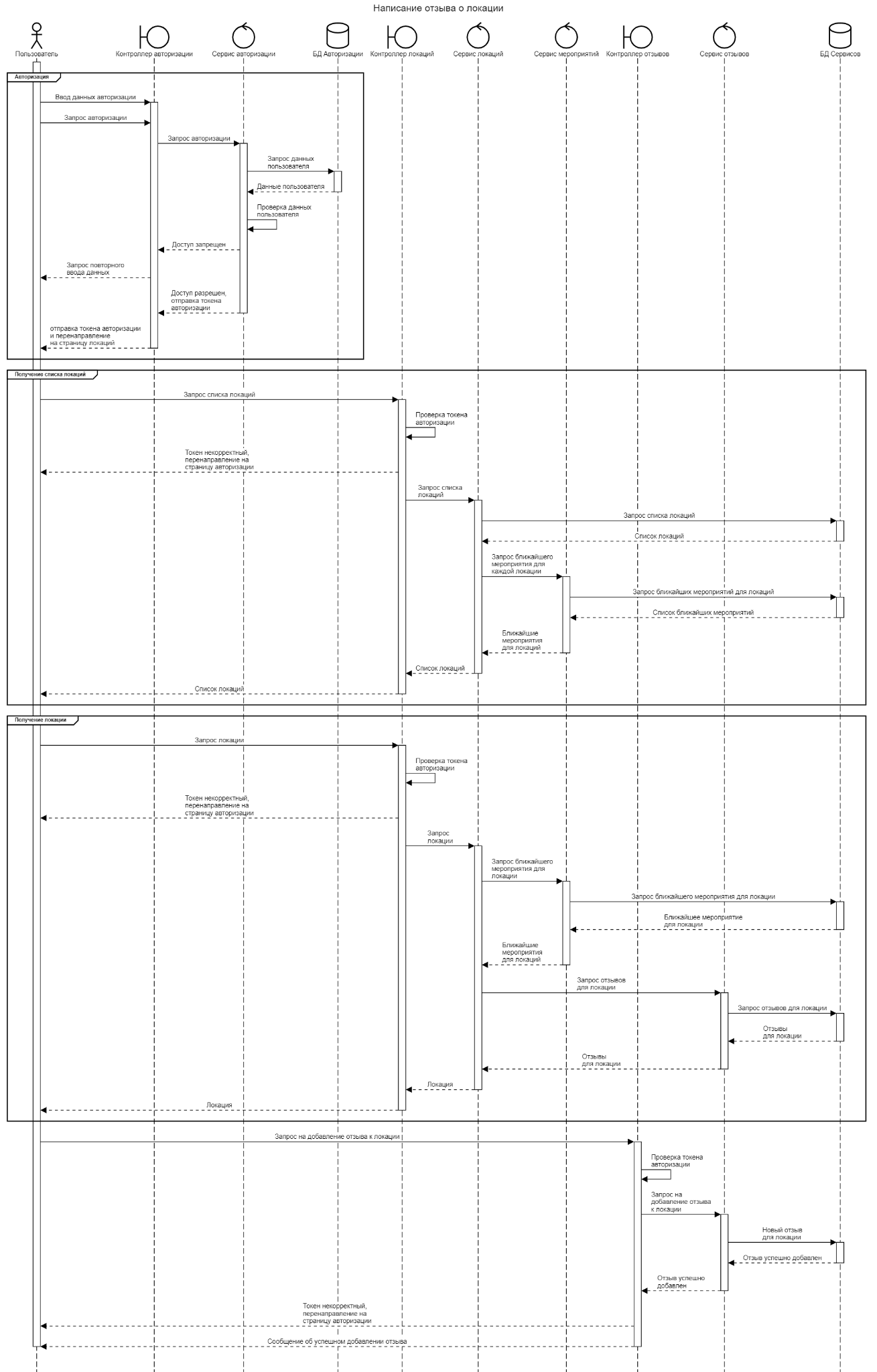


# Этап 2. BPMN и ER-диаграмма для программной системы

# 

# Этап 3. Use-Case, Sequence диаграммы

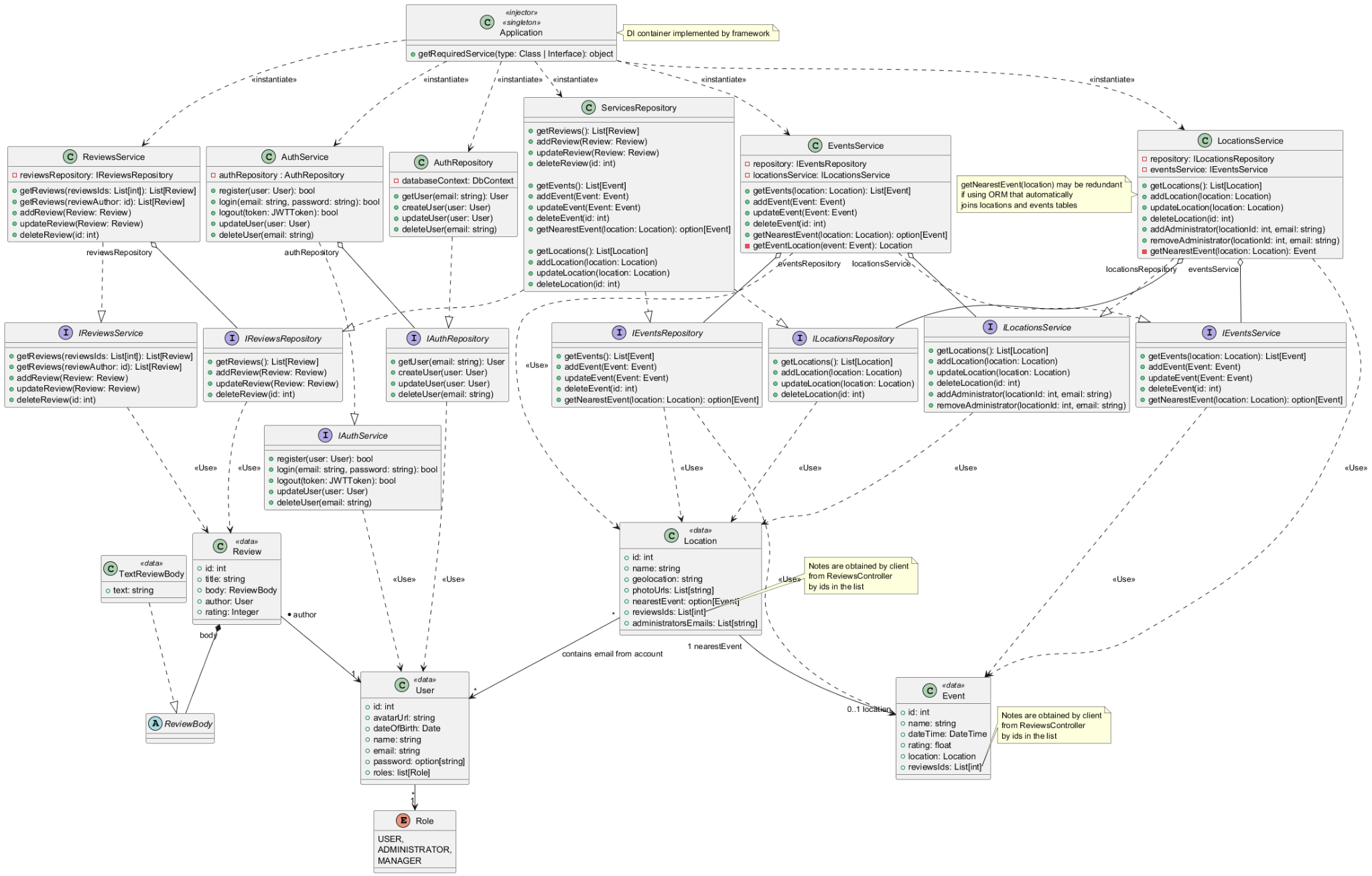




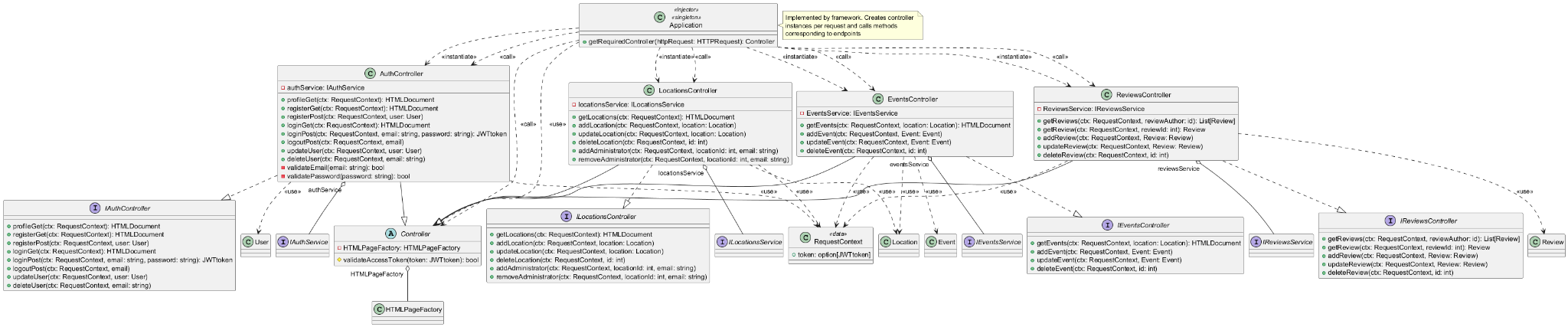
# Этап 4. Базовая диаграмма классов и диаграммы классов вариантов развития

Диаграмма классов системы построены по паттерну MVC и разбиты на соответствующие файлы:

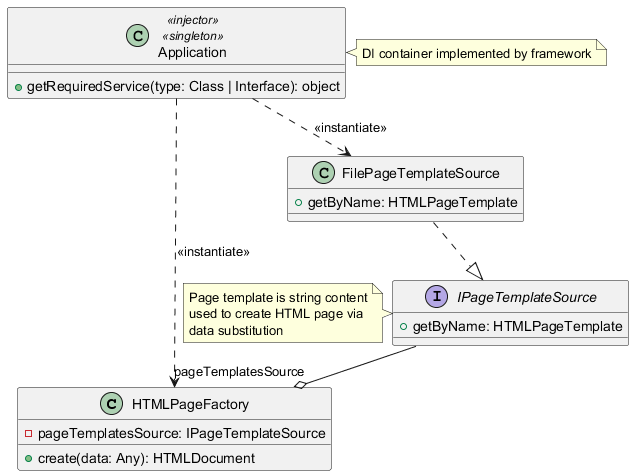
[CD\_model.png](https://drive.google.com/file/d/1nin-eBHQbBmvqHtK1bwSXnaltlCeJpjC/view?usp=drive_link)



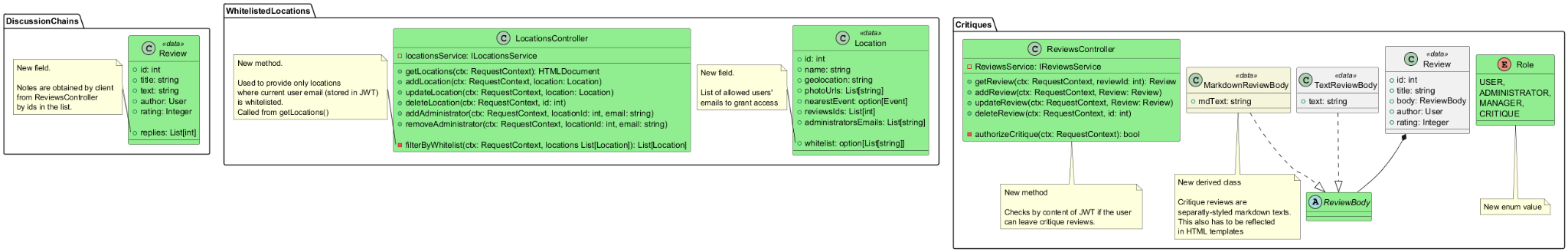
[CD\_controller.png](https://drive.google.com/file/d/112r9iMaYaF2yac7NKHa8JAkYbsOu6F8k/view?usp=drive_link)



[CD\_view.png](https://drive.google.com/file/d/1jd3gKZAZQVziaAYtF7p1G9oUQJXRWr3i/view?usp=drive_link)



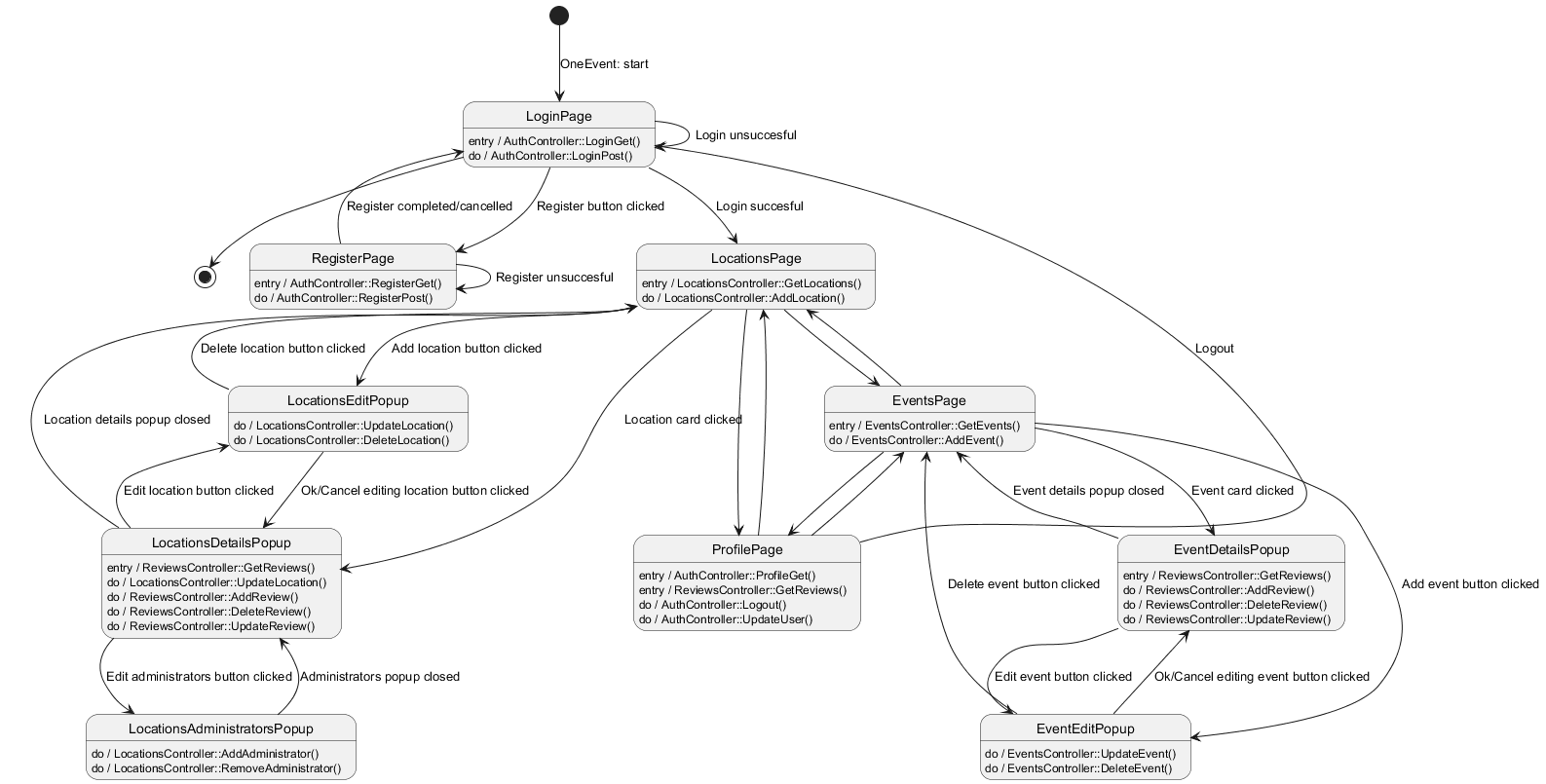
[CD\_development.png](https://drive.google.com/file/d/1oZypS0Ia1did7i6PIUNVneewxhQcGJTC/view?usp=drive_link)

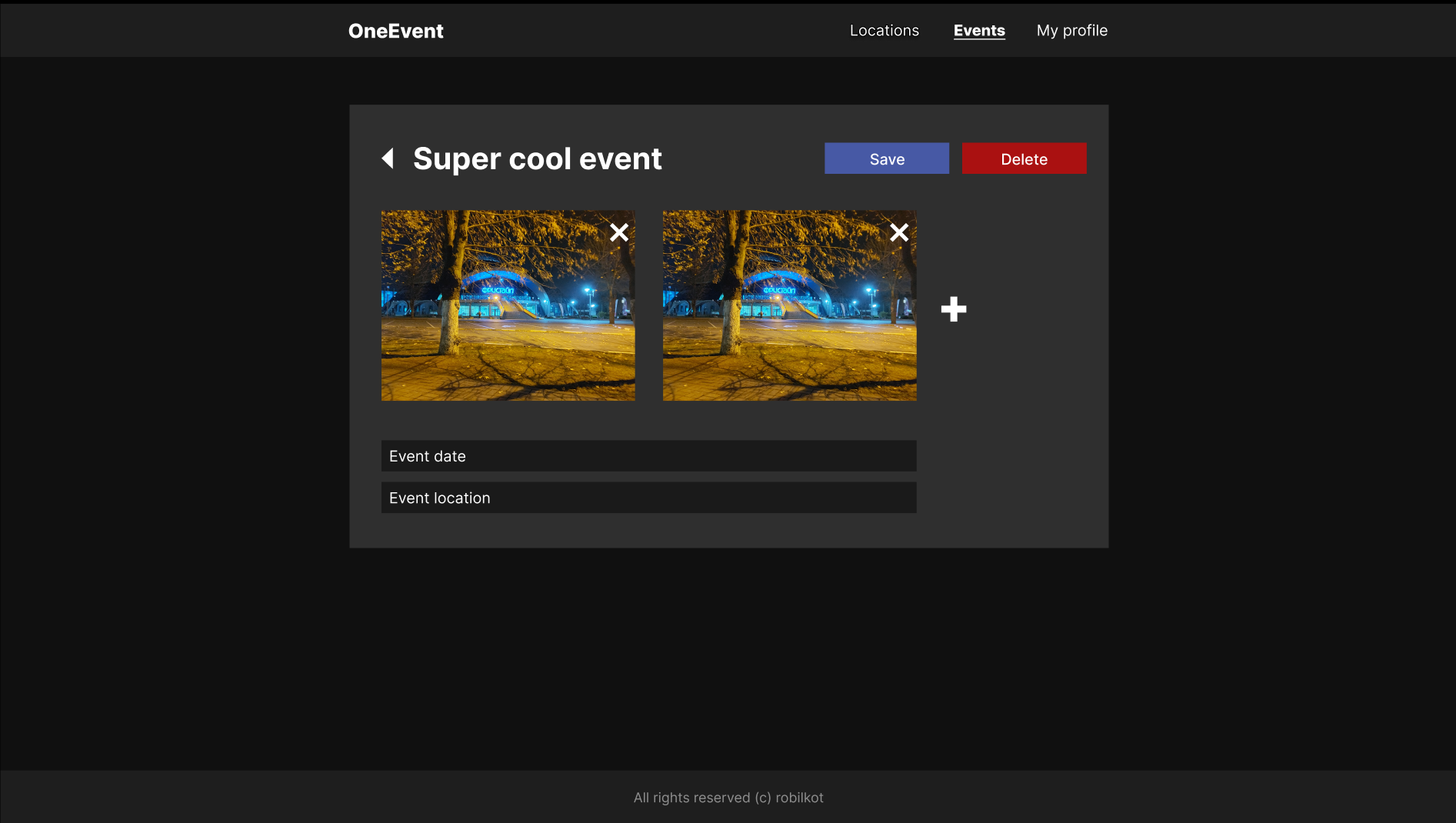
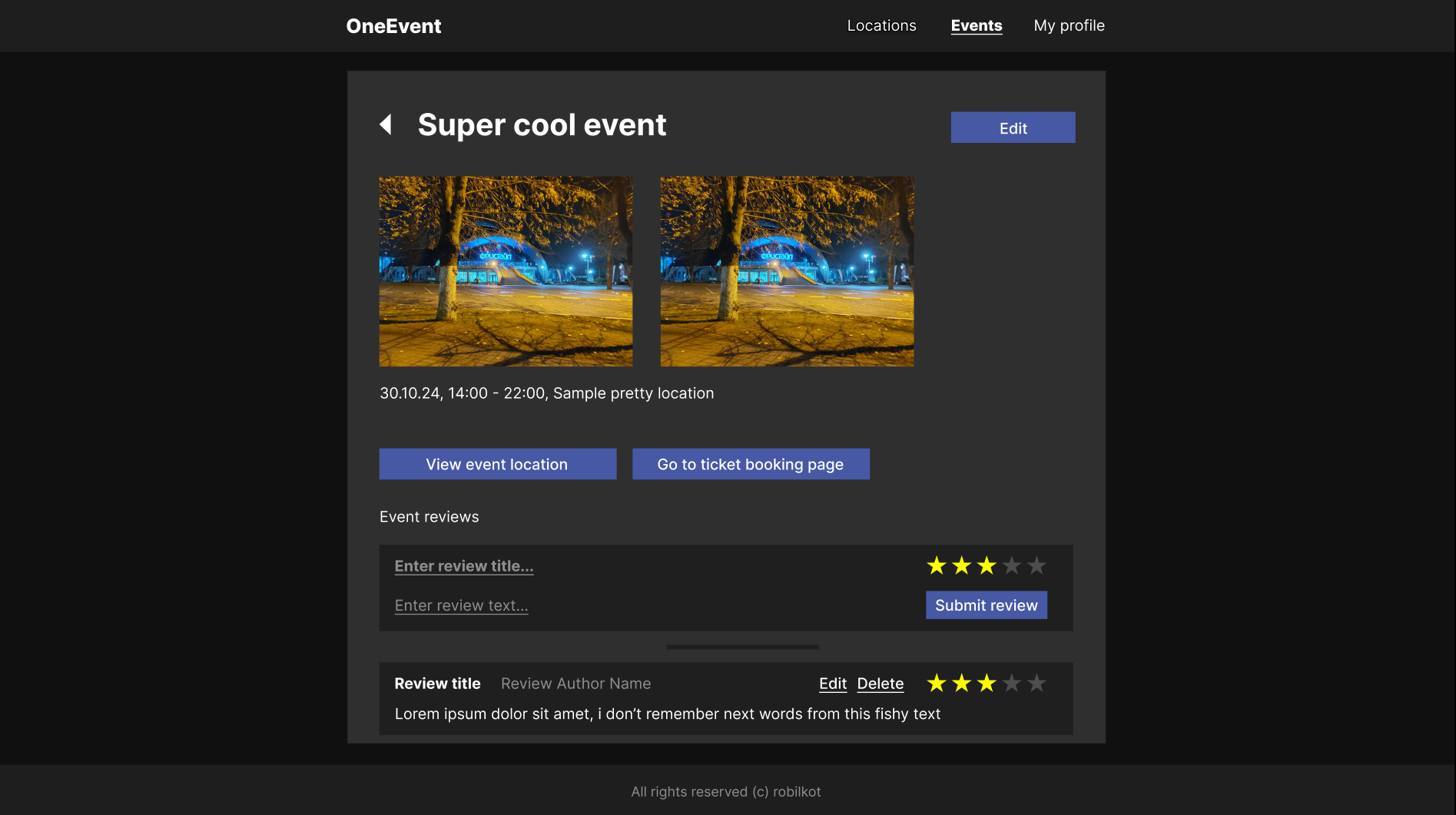
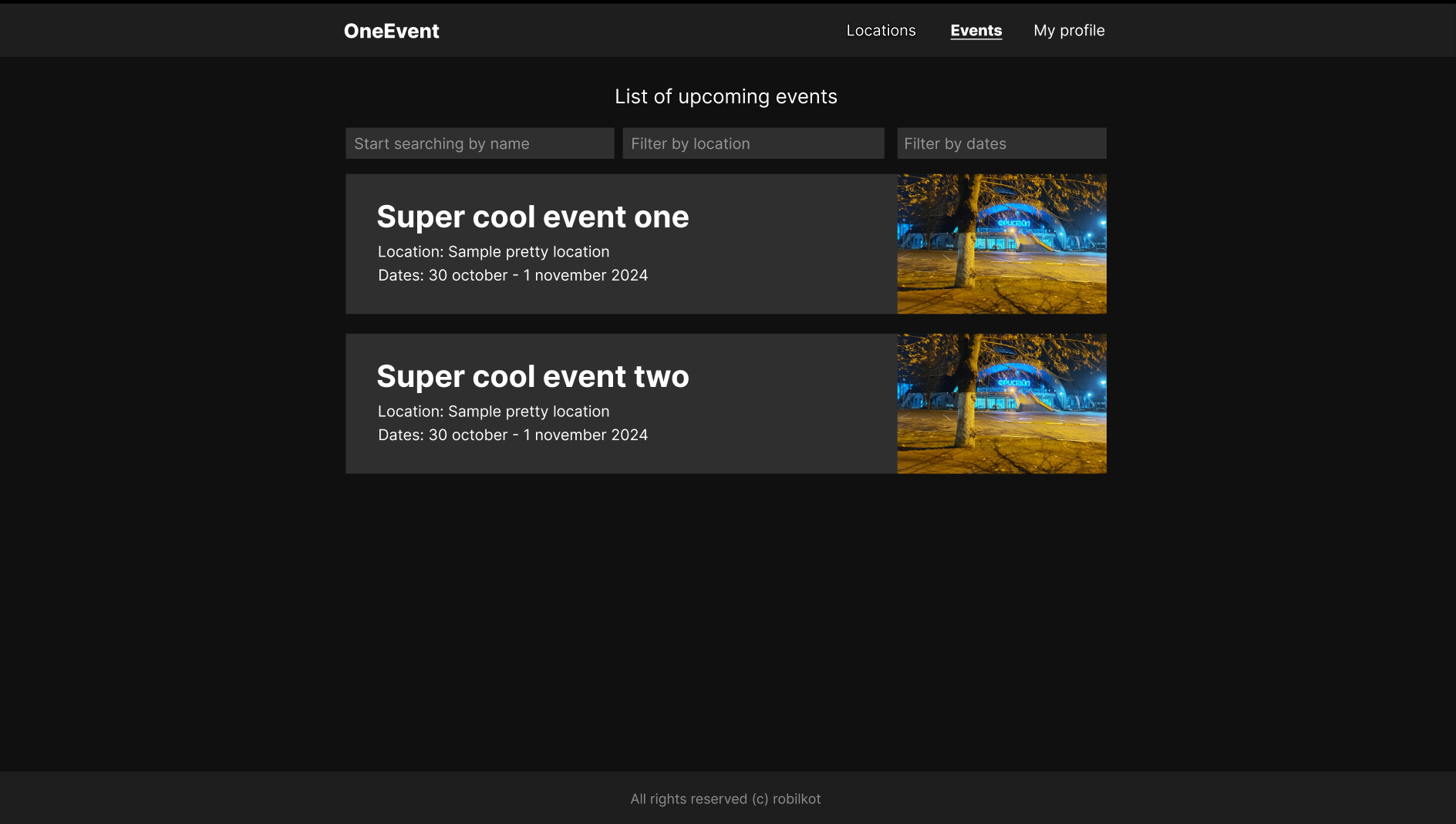
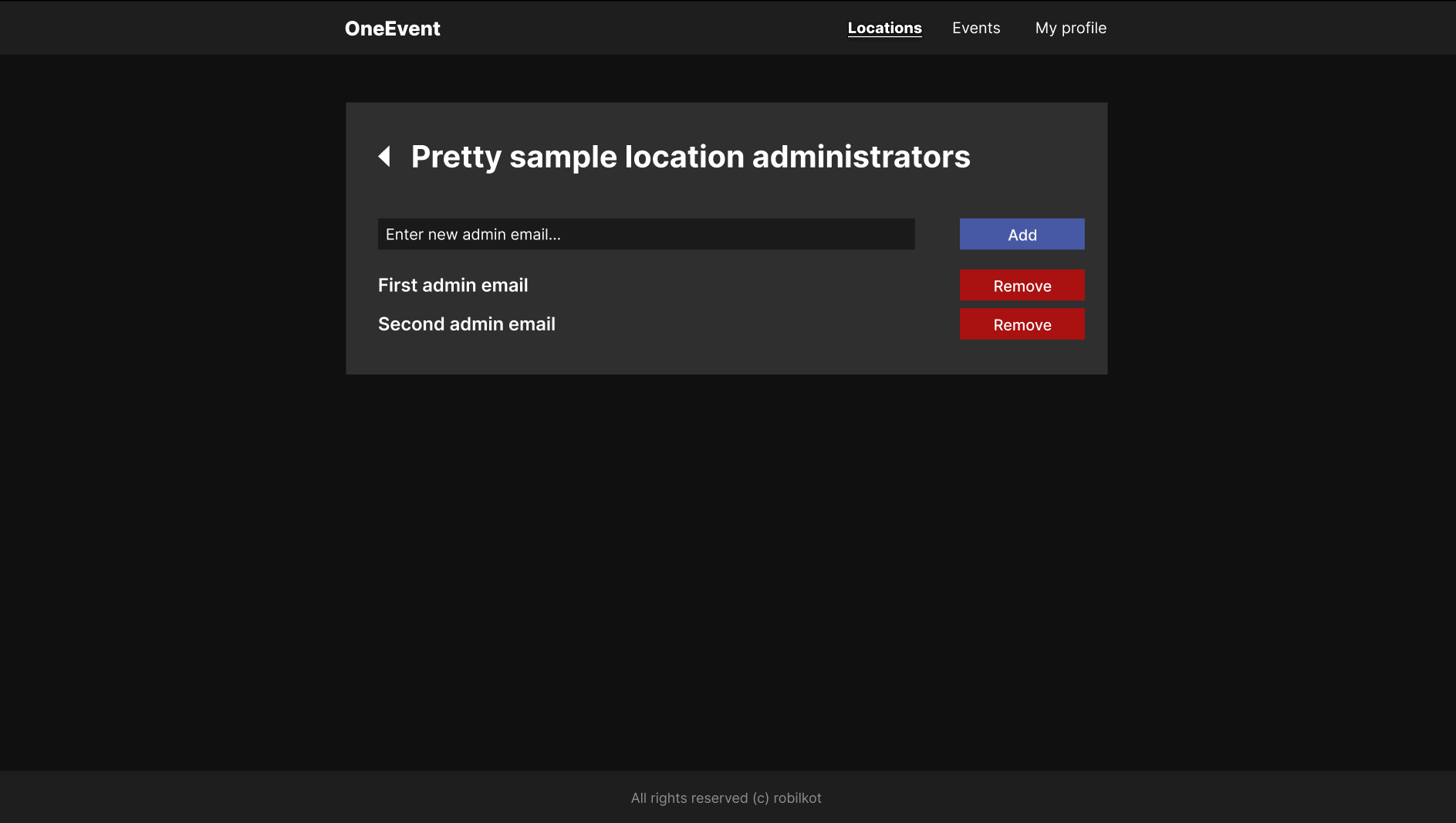
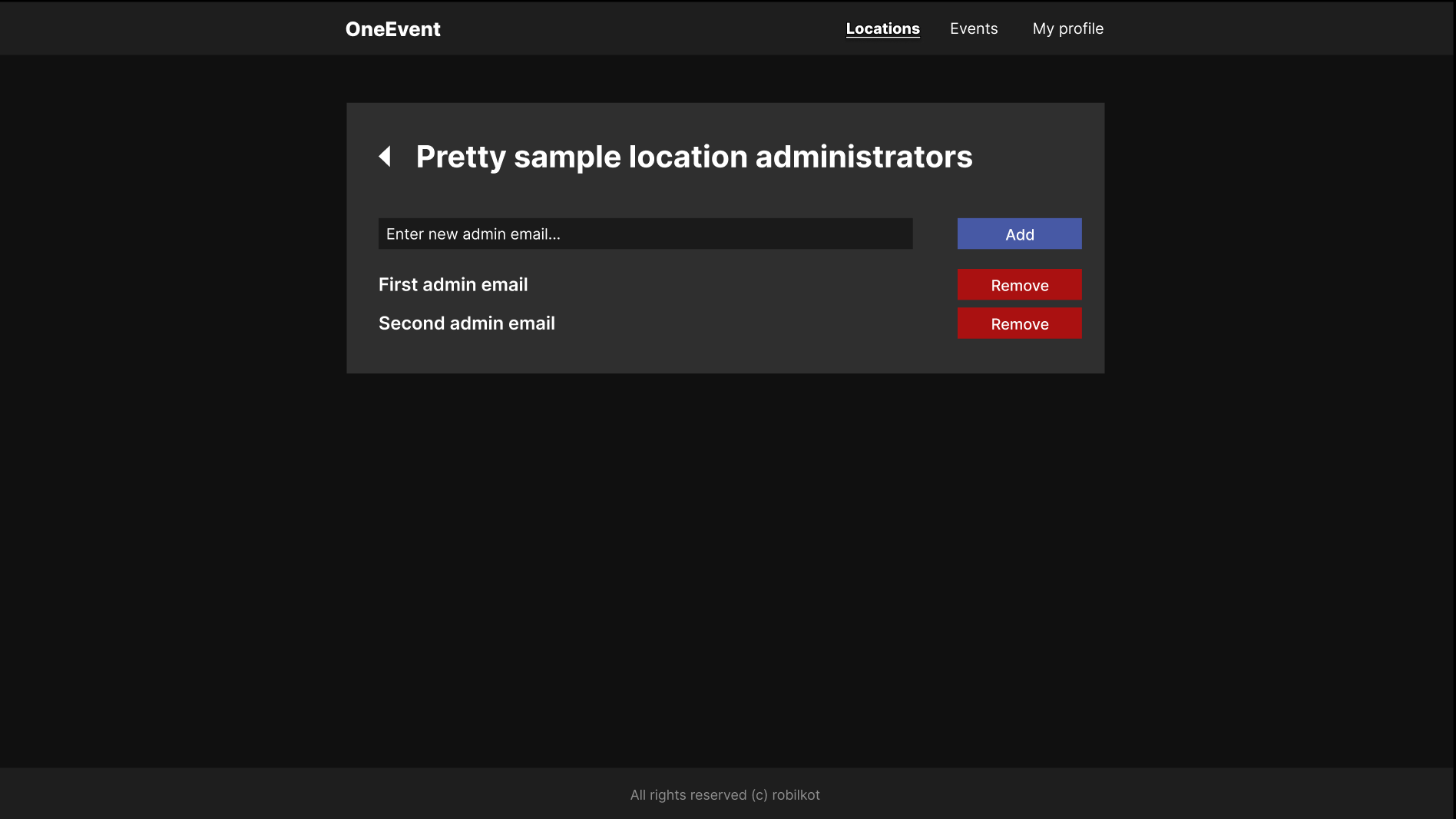
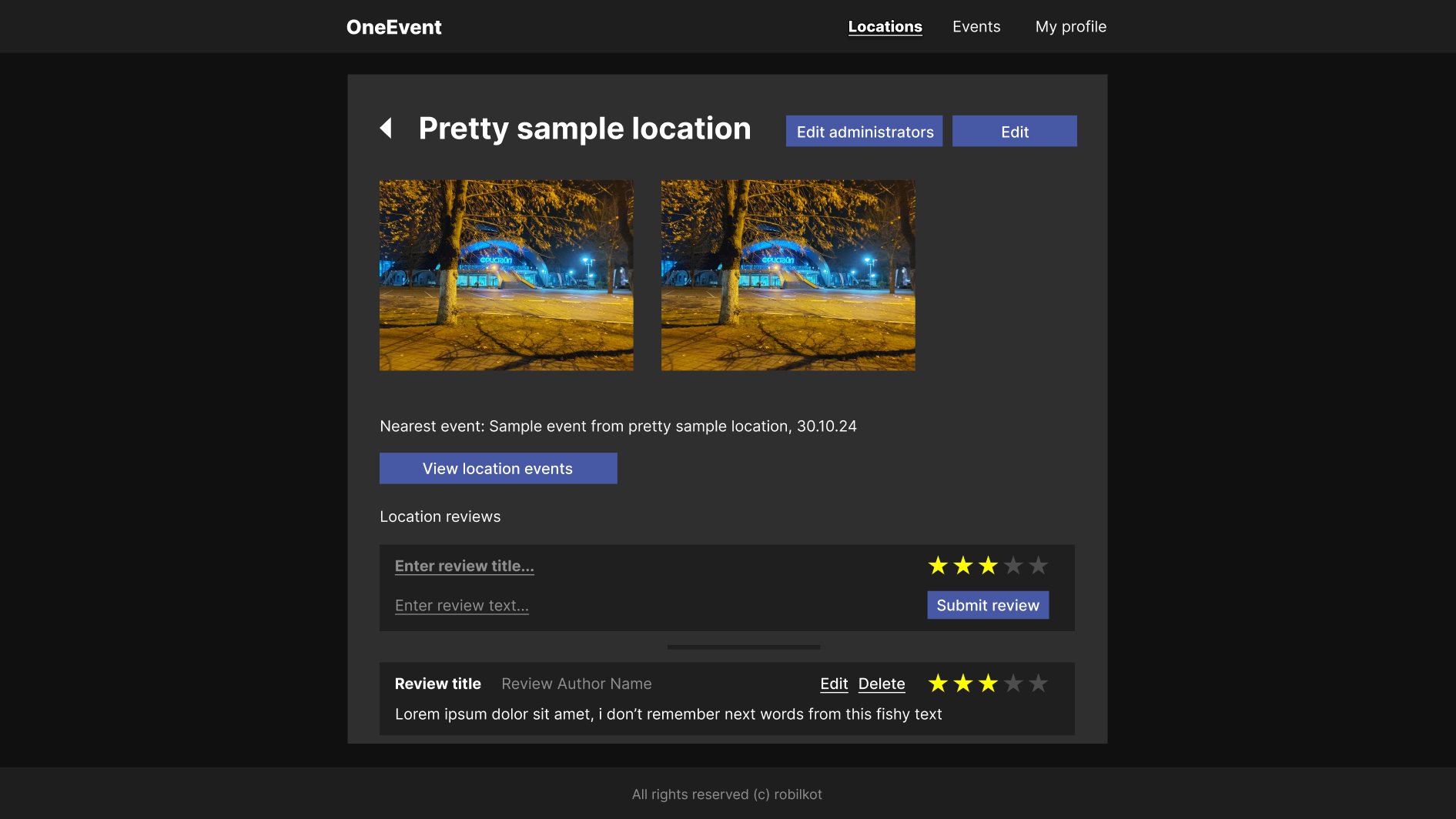
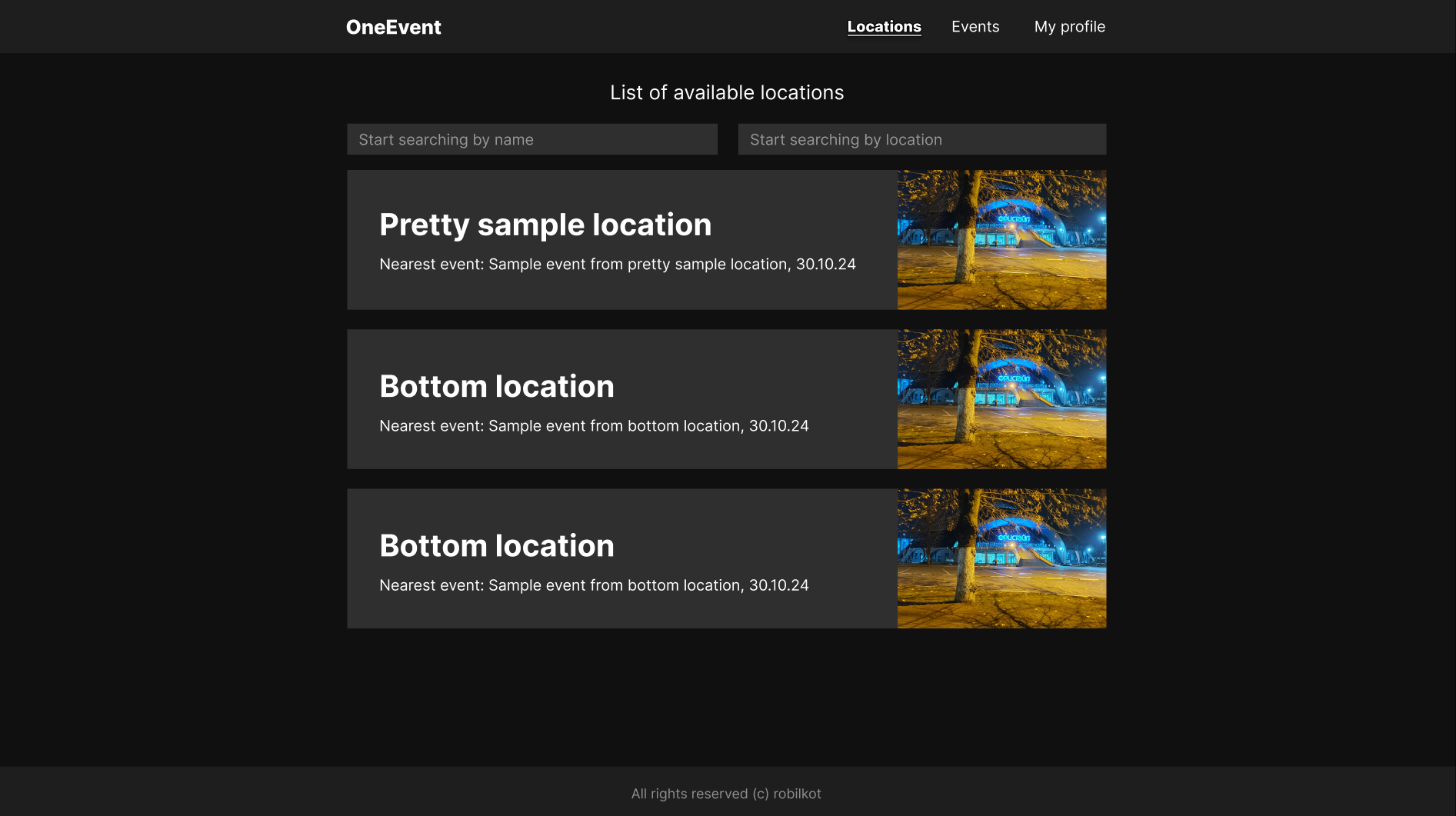
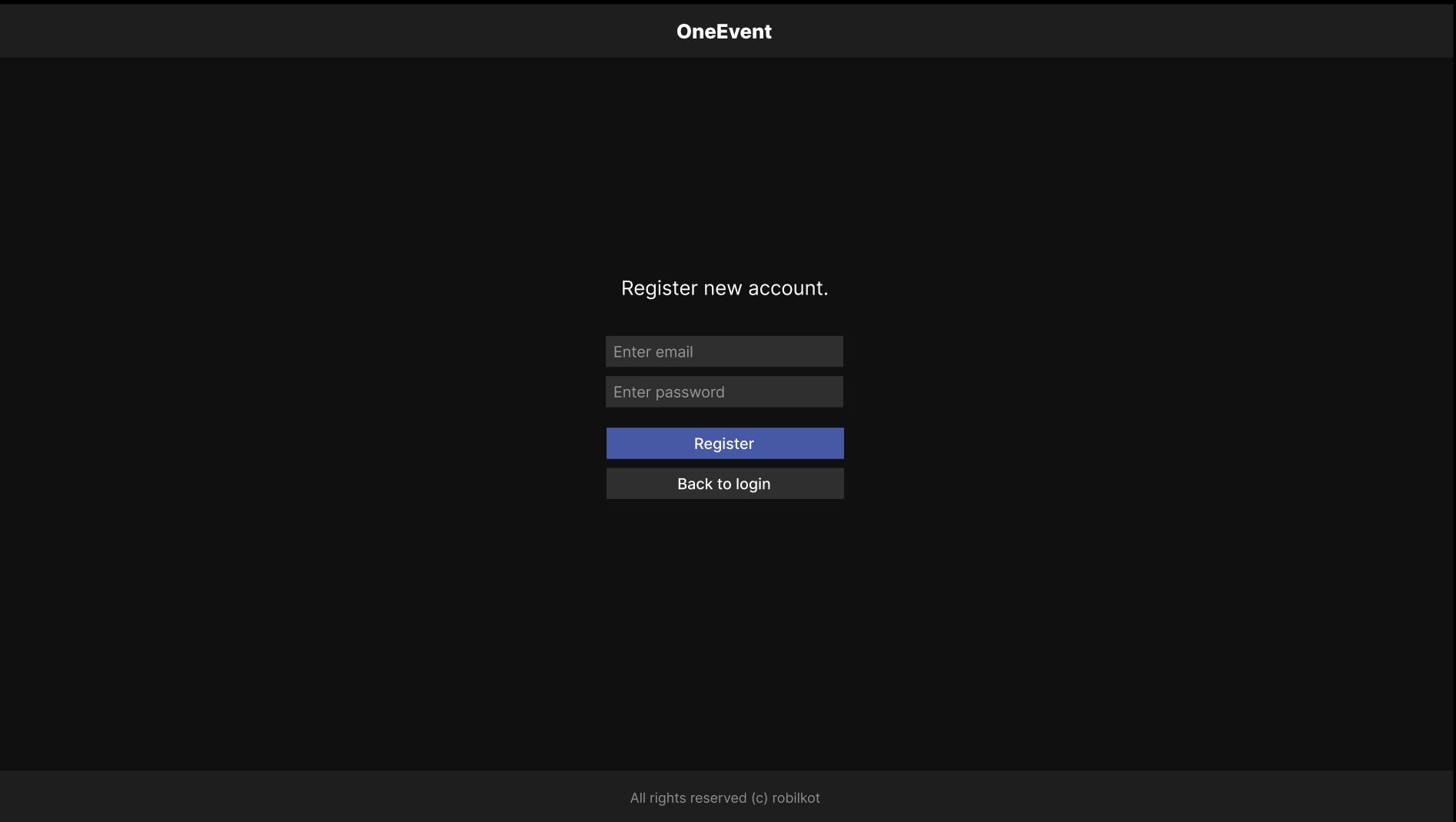
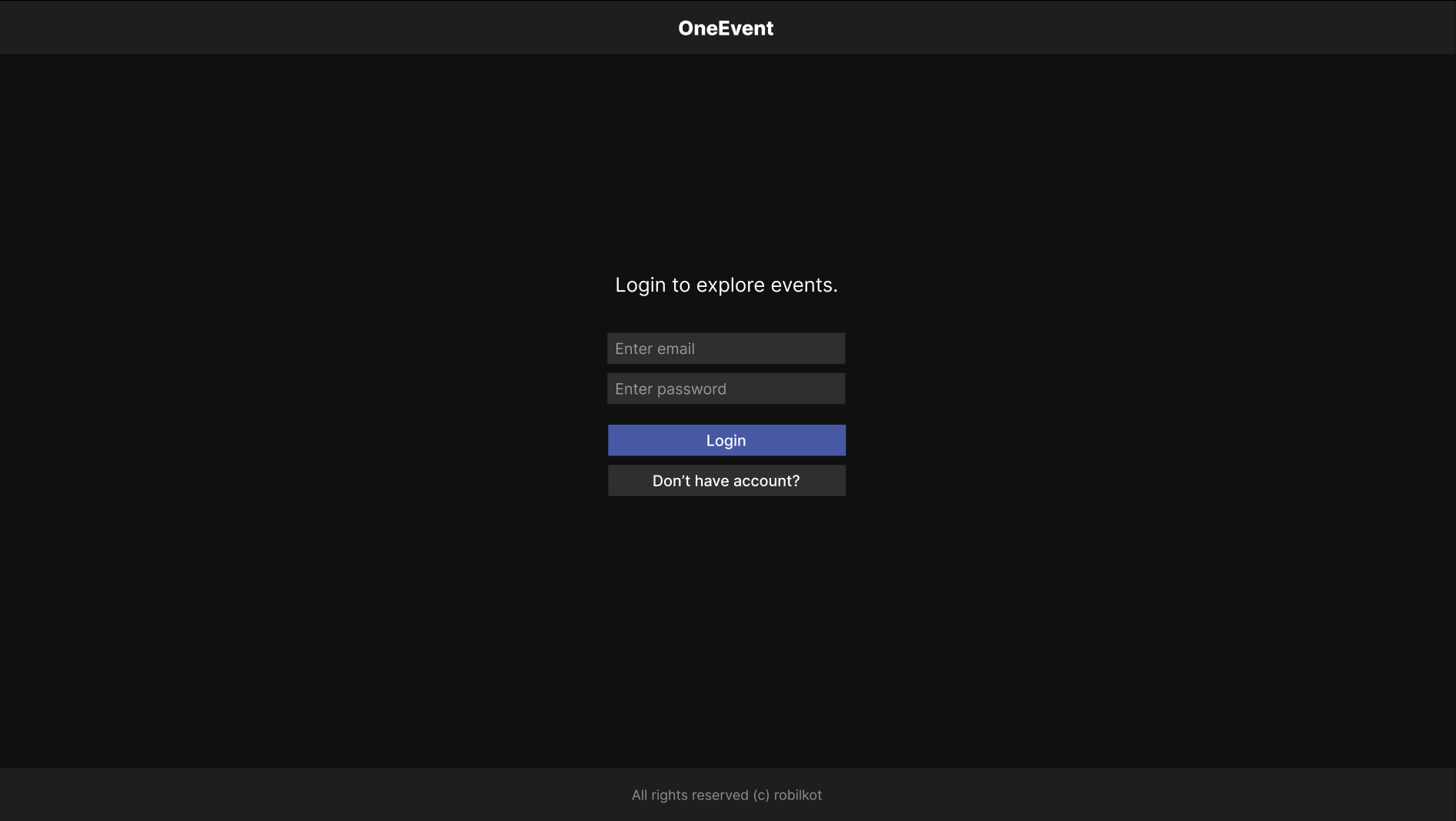


# 

# Этап 5. Диаграммы состояний, макеты интерфейса, рекомендации по реализации

[StateDiagram.png](https://drive.google.com/file/d/1vKNmFlFGZuw627xVstA9Cv3W9AFKm9Bd/view?usp=drive_link)





Для реализации рекомендую использовать язык программирования с готовыми фреймворками для построения webAPI и реализующих Inversion Of Control (Dependency injection pattern) контейнер.

В случае C# рекомендую использовать ASP.NET MVC со встроенным DI контейнером, Identity Server 4 (или любой аналог) для аутентификации, ORM Entity Framework Core для абстрагирования и упрощения доступа к БД. Для аутентификации можно использовать собственное решение, но это будет сложнее. Для отладки рекомендую применять Swagger UI и http-клиенты по типу Postman. Для юнит-тестирования удобно будет применить xUnit, NUnit, Moq и другие фреймворки.

В случае Python рекомендую использовать fastapi либо django из-за поддержки html шаблонов в соответствии с диаграммой классов, а также реализованного DI контейнера. Решение будет чем-то аналогично ЛР4 по ППОИС.

Возможно реализация и на других языках, к примеру Go.

# Лабораторная работа 2

Реализация системы

В рамках работы была реализована программная система для управления парковочными местами.

[Ссылка на проект](https://github.com/robilkot/BSUIR/tree/main/OMIS/Sem5/LW2_ParkingLots)

Для запуска проекта нужна платформа .NET 8.0 +

Проект можно открыть в Visual Studio. Понадобятся пакеты для языка C# и фреймворк Blazor.

*Примечание*: из-за структуры проекта Blazor реализация контроллеров находится в одних .razor файлах с представлением. Ввиду отсутствия реального функционала в проекте некоторые контроллеры в коде опущены.

Ошибки, допущенные при проектировании

* Были использованы enum, не позволяющие гибко расширять функционал
* Логическая ошибка в списке аргументов методов updateParkingZone
* Логическая ошибка в сигнатуре методов deleteParkingLot (должны принимать айди)
* Репозитории нарушают SRP и ISP
* Слой сервисов не несет смысловой нагрузки (слой дублирует методы репозиториев)
* Слой контроллеров не несёт смысловой нагрузки (отсутствуют функции валидации либо писание прочего функционала)
* LoginRepository предусматривает реализацию бизнес логики, хотя должен отвечать за доступ к данным
* ProfileRepository логические ошибки в сигнатурах методов; не ясно, по каким айди получать данные
* RegisterRepository и LoginRepository можно было объединить в один AuthRepo, так как очень близкая зона ответственности
* Метод GetUserDetails не несет смысловой нагрузик, данные пользователя уже получаются в других методах
* PaymentRepository должен быть сервисом
* Модель платежа не предусмотрена вовсе
* Модель бронирования не предусмотрена, из-за чего модель паркоместа нарушает SRP
* В истории бронирований требуется получать адрес паркинга, но взаимодействия с ZoneService не предусмотрены диаграммой классов
* Пагинация не предусмотрена(хотя в отзывах есть её подобие)
* Автора отзыва получить можно только через обращение к сервису профилей, что не предусмотрено диаграммой классов
* Пользователь может оставить только один отзыв, что противоречит UI
* Стоимость брони и цена за один час не различимы по модели. Расчёт финансов это не ответственность фронтенда
* На UI не показаны оценки в отзывах
* Элементы UI применены неуместно (например, чекбоксы вместо радиобаттонов, а также flyout с сомнительной логикой использования)
* Многие макеты можно объединить в один, например выбор парковочного места.
* Ошибка в диаграмме состояний: процесс создания парковочного места не описан полностью и не соответствует макетам
* Нельзя выйти из профиля
* Макеты восстановление пароля не соответствуют диаграмме состояний
* Окна отмены бронирования нет на макетах

# Лабораторная работа 3

## Анализ и проектирование:

Ошибки при проектировании:

* Отсутствует Use-Case диаграмма
* Отсутствует Sequence диаграмма
* Нету рекомендаций по реализации

Ошибки в диаграммах:

* Диаграмма классов местами нечитаема; есть перекрытия, часть данных обрезана или закрыта
* Разбиение контроллера отзывов на несколько контроллеров добавления, удаления и т.д. излишне. Аналогично с другими контроллерами - Over-SOLID.
* Сигнатуры контроллеров нелогичные. Состав пользователей не должен присутствовать в их аргументах с учётом выбранной CSR архитектуры.

Прочие ошибки были описаны в отчёте о реализации проекта.

## Реализация:

Лабораторная работа 2 была реализована в соответствии с диаграммой состояний, а также текстовым описанием. Стоит отметить, что есть некоторые различия с данной информацией о системе, но они обусловлены несоответствиями и ошибками при проектировании.

Присутствуют несоответствия с диаграммой классов модели:

* Нет классов состава пользователей
* Нету классов клиента, админа. Есть только пользователь
* На диаграмме классов есть интерфейсы по типу “отзыв”, в реализации их нет
* На диаграмме есть 12 конкретных контроллеров, в реализации их 19
* Не реализованы классы менеджеров. Ближайший их аналог - классы сервисов

Преимуществом реализации можно отметить исправление некоторых несоответствий текстового описания диаграмме классов и диаграмме состояний, а также реализацию внутренней логики и грамотную реализацию интерфейса.