МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ИС

КУРСОВАЯ РАБОТА

по дисциплине «Архитектура информационных систем»

Тема: проектирование сервиса каршеринга

Студенты гр. 2375 _____ Усачев Л.Е.
Шитов Б.А.

Преподаватель _____ Водяхо А.И.

Санкт-Петербург
2024

ЗАДАНИЕ
НА КУРСОВУЮ РАБОТУ

Студенты: Усачев Л.Е., Шитов Б.А.

Группа: 2375

Тема работы: проектирование сервиса каршеринга

Исходные данные: средствами ПО Enterprise Architect спроектировать сервис каршеринга. Сформировать технические требования, архитектурное описание и список тестов для проекта. Богдан Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан

Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан

Богдан

Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан
Богдан

Содержание пояснительной записки: введение, требования, архитектурное описание, архитектурное обоснование, модели, UML описание, use case, классы, активности, развертка, тестирование, заключение, список литературы.

Предполагаемый объем пояснительной записки: не менее 25 страниц

Дата выдачи задания: 5.02.2024

Дата сдачи отчёта:

Студенты гр. 2375 _____ Усачев Л.Е.
Шитов Б.А.

Преподаватель _____ Водяхо А.И.

АННОТАЦИЯ

Содержание курсовой работы заключается в проектировании сервиса каршеринга с помощью ПО Enterprise Architect. Были сформированы технические требования, архитектурное описание, диаграммы и список тестов для проекта.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель работы: проектирование сервиса каршеринга.

Форма выполнения работы: курсовая работа.

Задание: Разработка базы данных для сервиса каршеринга, в которой хранится информация о автомобилях, их местоположении, состоянии, доступности, а также о клиентах и их арендных операциях. Для каждого автомобиля фиксируется: марка, модель, год выпуска, расход топлива, тип кузова, цвет, текущее местоположение, статус доступности. По каждому клиенту сохраняется следующая информация: ФИО, контактные данные, водительское удостоверение, история аренд. Аренда автомобилей планируется по дням и включает в себя список доступных автомобилей и продолжительность аренды каждого автомобиля. В сервисе каршеринга предусмотрена одна парковка. Поступление новых автомобилей на парковку отслеживается через акты приема-передачи. Каждый акт приема-передачи содержит следующие атрибуты: номер документа, дата документа, парковка,

на которую были доставлены автомобили, поставщик, марка и модель автомобиля, количество каждой модели, стоимость аренды за единицу времени и общая сумма за все автомобили.

1.Требования

1.1. Глоссарий

База данных - Набор постоянно хранимых данных, используемых прикладными программными системами.

Система управления базами данных (СУБД) - Программный комплекс, обеспечивающий централизированное хранения данных и предоставляющий приложениям услуги по обработке данных.

Фреймворк - Программное обеспечение, облегчающее разработку и объединение разных модулей программного проекта.

Use case - Описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды.

Автомобиль - Транспортное средство, доступное для аренды в сервисе каршеринга. Атрибуты автомобиля включают марку, модель, год выпуска, расход топлива, тип кузова, цвет, текущее местоположение и статус доступности.

Клиент - Человек, использующий услуги каршеринга. Данные о клиенте содержат ФИО, контактные данные, водительское удостоверение и историю аренд.

Аренда - Действие или период времени, в течение которого клиент использует автомобиль. Включает в себя продолжительность аренды и список выбранных автомобилей.

Парковка - Место, где хранятся автомобили сервиса каршеринга, когда они не используются. В данной системе предусмотрена одна парковка.

Акт приема-передачи - Документ, фиксирующий поступление новых автомобилей на парковку. Содержит номер документа, дату документа, информацию о парковке, поставщике, марке и модели автомобиля, количество каждой модели, стоимость аренды за единицу времени и общую сумму за все автомобили.

Местоположение автомобиля - Текущее расположение автомобиля, которое может быть использовано для определения его доступности для клиентов.

Статус доступности - Индикатор, показывающий, доступен ли автомобиль для аренды в данный момент.
История аренд - Записи о всех прошлых арендах, совершенных клиентом. Включает информацию о дате аренды, продолжительности и типе арендуемого автомобиля.

Расход топлива - Количество топлива, которое автомобиль потребляет на определенное расстояние (обычно измеряется в литрах на 100 км).

Тип кузова - Категория или стиль кузова автомобиля, например, седан, хэтчбек или внедорожник.

Водительское удостоверение - Официальный документ, подтверждающий право клиента управлять автомобилем.

Поставщик - Компания или индивидуум, предоставляющий автомобили для сервиса каршеринга.

Стоимость аренды за единицу времени - Цена использования автомобиля за определенный период времени (час, день и т.д.).

1.2 Бизнес-требования

1. Система должна обеспечивать управление поставками автомобилей на парковку.
2. Система должна позволять отслеживать наличие автомобилей на парковке.

3. Система должна предоставлять возможность рассчитывать стоимость аренды автомобиля в зависимости от времени использования и километража.
4. Система должна автоматически предлагать доступные автомобили для аренды, исходя из текущих запросов пользователей.

1.3 Пользовательские требования

1. Интерфейс сервиса должен быть интуитивно понятным и удобным в использовании.
2. Доступ к системе разрешен только после прохождения авторизации.
3. Доступ к системе должен быть возможен через интернет, чтобы обеспечить доступность с любого устройства.
4. Пользователи должны иметь возможность изменять и удалять данные своих бронирований.

1.4 Системные требования

1. Система должна поддерживать функции создания, редактирования и удаления данных о бронированиях и автомобилях.
2. База данных должна быть доступна для работы через интернет.
3. Доступ к базе данных должен быть строго регламентирован с разделением прав пользователей.
4. Необходим выделенный сервер для хранения и обработки данных.
5. Система должна гарантировать сохранность данных в случае технических сбоев.

1.5 Функциональные требования

1. Учет базы данных автомобилей, клиентов и парковок.
2. Автоматическое предложение доступных автомобилей для аренды в зависимости от запросов пользователей.
3. Расчет стоимости аренды автомобиля на основе времени и пройденного расстояния.
4. Ввод данных о новых поставках автомобилей на парковку.
5. Мониторинг наличия автомобилей на парковке и оповещение о необходимости пополнения парка.
6. Возможность авторизации администратора для управления системой.
7. Ведение базы данных поставщиков автомобилей.
8. Создание актов приема-передачи при поставке новых автомобилей на парковку.
9. Регистрация новых пользователей в системе.

2. Архитектурное описание

2.1 Архитектурное обоснование

Для разработки нашего веб-приложения мы решили использовать Java и Spring. Java - это высокоуровневый язык программирования, который обладает простым и интуитивно понятным синтаксисом, а также множеством библиотек и фреймворков, что делает его очень удобным для разработки приложений. Spring - это фреймворк для разработки приложений на языке Java, предоставляющий инструменты и библиотеки для упрощения разработки, улучшения производительности и обеспечения безопасности.

Для хранения и обработки информации мы выбрали реляционную систему управления базами данных PostgreSQL. Она обладает высокой производительностью и масштабируемостью, а также имеет множество инструментов для обработки и анализа данных.

Для обеспечения безопасности и надежности при проведении платежей мы выбрали платежный шлюз системы Robokassa. Это позволит нам обеспечить высокий уровень защиты данных и обеспечить безопасность при проведении платежей. Платежный шлюз Robokassa поддерживает множество

способов оплаты, что позволит нашим пользователям выбирать наиболее удобный для них способ оплаты.

Наше веб-приложение будет создано с использованием клиент-серверной архитектуры. Клиентская часть приложения будет работать на стороне пользователя и обеспечивать интерфейс для взаимодействия с приложением. Серверная часть приложения будет обрабатывать запросы от клиентской части и взаимодействовать с базой данных и платежным шлюзом. Для взаимодействия между клиентской и серверной частями приложения мы будем использовать протокол HTTPS.

В целом, наше веб-приложение будет обладать высокой производительностью, масштабируемостью и безопасностью, что позволит нам обеспечить нашим пользователям удобный и безопасный опыт использования приложения.

2.2 Модели

В таблице 1. представлены данные и методы работы с ними.

Таблица 1. Данные и методы

Объект	**Методы**	**Свойства**
User	Регистрация, аренда автомобиля, возврат автомобиля, просмотр информации о доступных автомобилях, просмотр истории поездок	ID, ФИО, номер телефона, логин и пароль, статус аренды, история поездок, документы
Car	Отслеживание местоположения автомобиля, бронирование автомобиля, завершение аренды	ID автомобиля, марка и модель автомобиля, географические координаты, статус доступности
RentalHistory	Просмотр истории аренды автомобилей, расчет общей стоимости аренды	ID аренды, ID пользователя, ID автомобиля, координаты начала и окончания

		поездки, дата начала и окончания аренды, стоимость каждого дня аренды

2.3 UML описание

2.3.1 Use case

Пользователи делятся на два вида:

Администратор. Возможности: проверка документов новых пользователей, редактирование базы данных и состояний пользователей и доступных машин, круглосуточная поддержка пользователей.

Обычный пользователь. Возможности: бронирование доступных для поездки автомобилей, оплата поездок, загрузка документов, возможность связи с технической поддержкой.

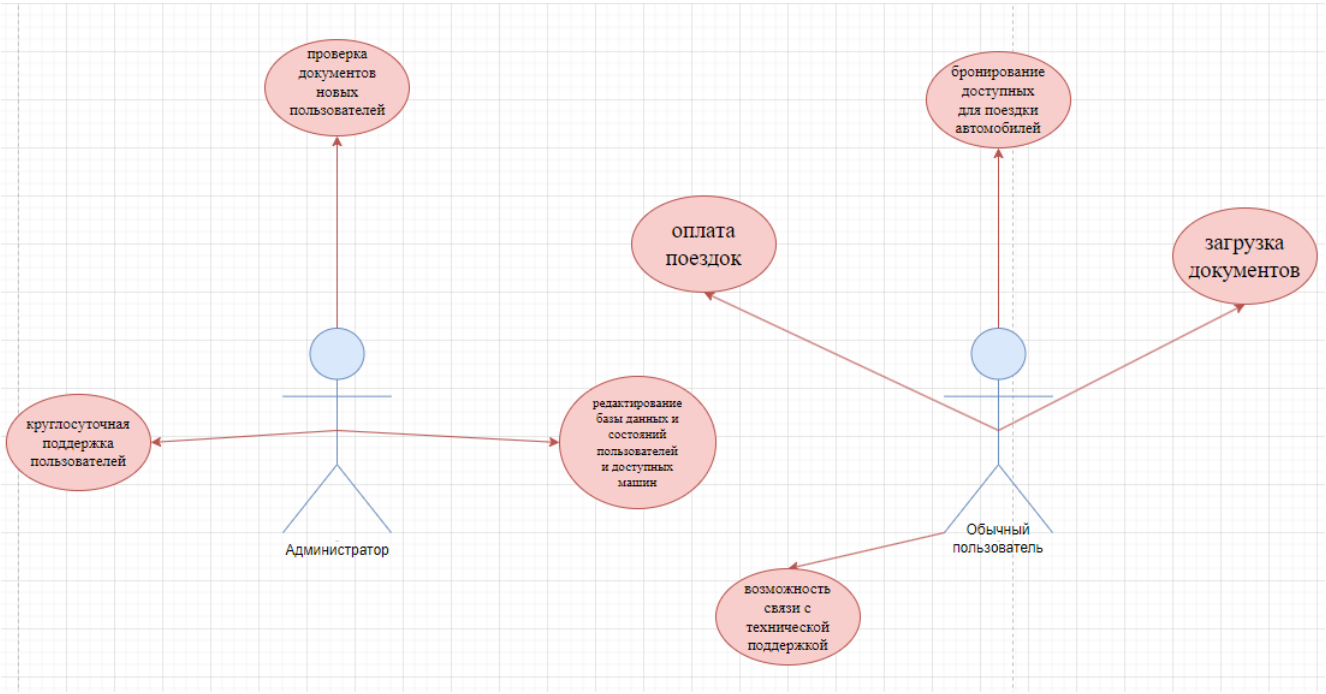

Рисунок 1. Use case диаграмма проекта