МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**Факультет программной инженерии и компьютерной техники**

**Лабораторная работа No1**

по дисциплине

“Архитектура программных систем”

Студент:

Гуменник Петр Олегович

Группа P3333

Преподаватель:

Перл Иван Андреевич

Санкт-Петербург, 2025

## Задание:

Выбрать любую реально существующую систему и описать её в терминах UML. Желательно, чтобы система была не полностью информационной, но опиралась на информационную систему как показано в примере на лекции (Point of sale). Необходимо описать границы системы на разных уровнях, а также описать сценарии использования для нескольких Акторов.

## Описание рассматриваемой системы с требованиями к ней:

Рассматриваемая система - это **автоматизированная парковочная система**, которая сочетает в себе физическое оборудование (шлагбаум, терминалы въезда/выезда) и программно-аппаратный комплекс для идентификации, учёта оплаты и управления пропуском автомобилей на парковку.

### Основные элементы системы

1. **Въездной терминал** - оснащён сканером/валидатором билетов и кнопкой для выдачи нового парковочного билета.
2. **Выездной терминал** - оснащён сканером/валидатором билетов и информирует об остатке времени или необходимости оплаты.
3. **Шлагбаум** - физическое устройство, блокирующее проезд, открывается только после получения сигнала от системы при наличии действующего билета или абонемента.
4. **Платёжные терминалы** - устройства для оплаты парковки (оплата наличными или картой).
5. **Информационная система** - серверное ПО, обрабатывающее данные о времени заезда/выезда, состоянии оплат, ведущее журнал событий и предоставляющее интерфейс для администратора.

### Требования к системе

### Функциональные требования

1. **Регистрация въезда**
   * Система должна выдавать посетителю парковочный билет при въезде (или считывать абонемент).
   * Записывать в базе данных время въезда и идентификатор билета.
2. **Оплата парковки**
   * Система должна предоставлять функциональность оплаты через платёжный терминал (наличные, банковская карта).
   * Система должна обеспечивать связь между платёжным терминалом и базой данных для обновления статуса оплаты.
3. **Регистрация выезда**
   * Система должна считывать (валидировать) билет при выезде.
   * Система должна проверять статус оплаты (оплачен ли билет и действителен ли абонемент).
   * При успешной проверке отправлять команду на открытие шлагбаума.
4. **Администрирование**
   * Возможность добавлять и редактировать тарифы в базе данных.
   * Предоставлять отчёты по времени пользования парковкой, количеству автомобилей и суммам оплат.
5. **Обслуживание и диагностика**
   * Предусмотреть отдельный интерфейс или функции для сервис-инженера, позволяющие тестировать работу шлагбаума и терминалов.
   * Уведомлять администратора о сбоях в работе системы (логирование ошибок, событий).

### Нефункциональные требования

1. **Надёжность**
   * Система должна продолжать работу в случае отказа одного из терминалов (избыточность, возможность ручного пропуска при сбое).
2. **Безопасность**
   * Доступ к административным функциям должен быть защищён (аутентификация и авторизация).
   * Система должна шифровать критические данные (например, данные платежей).
3. **Производительность**
   * Время ответа системы при считывании билета не должно превышать 2–3 секунд.
   * Обновление статуса оплаты в базе данных должно происходить моментально (максимум 2 секунды).
4. **Удобство использования**
   * Интерфейс платёжного терминала должен быть интуитивно понятен и прост в использовании.
   * Физическое расположение терминалов на парковке должно обеспечивать удобный подъезд и считывание билета.

## Формальное описание системы с необходимым количеством UML диаграмм

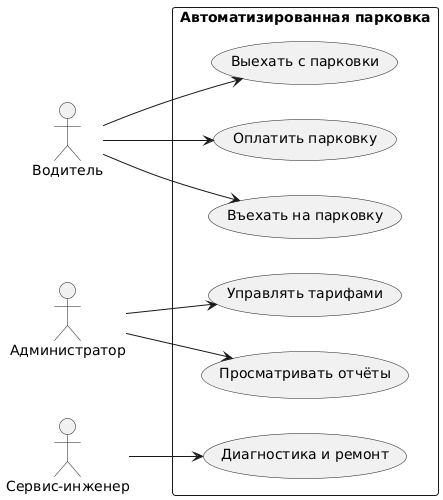


Рисунок 1. Общий обзор системы

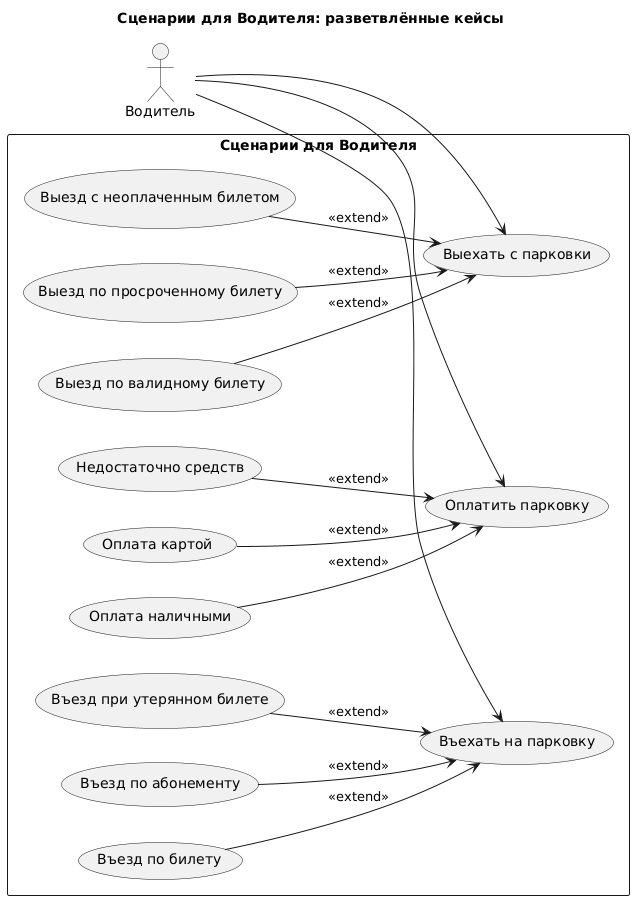


Рисунок 2. Use-case водитель

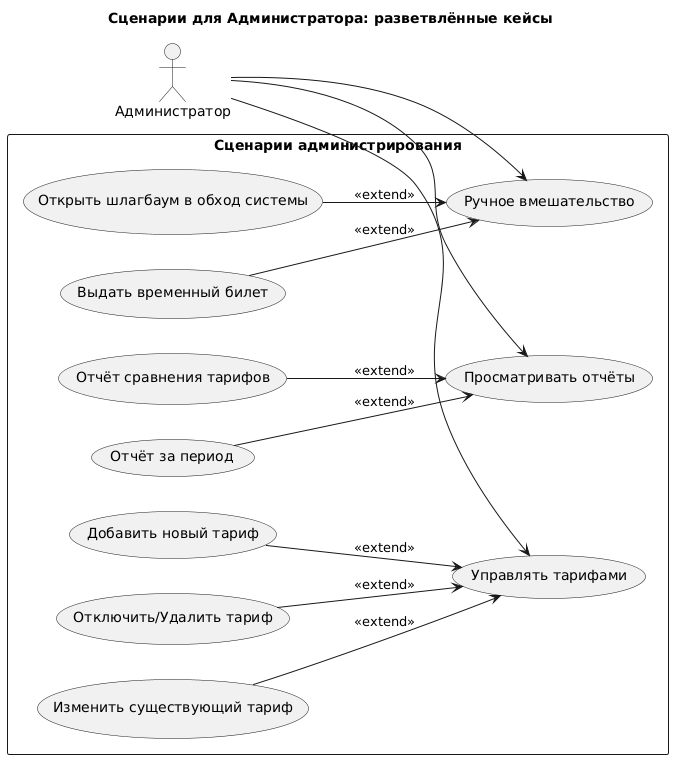


Рисунок 3. Use-Case администратор

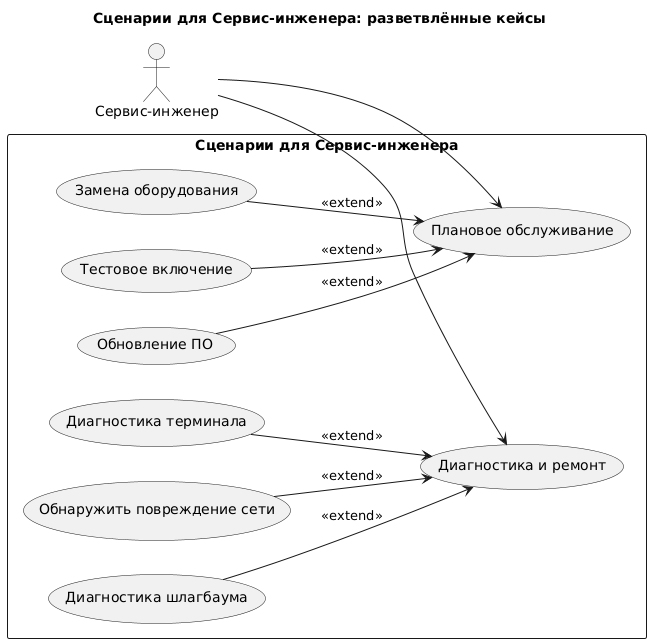


Рисунок 4. Use-Case Сервис-инженер

### 

Рисунок 5. Диаграмма классов

## Словесное описание сценариев использования

### Сценарии использования для ****Водителя****

1. **«Въезд на парковку по билету»**
   1. Водитель подъезжает к въездному терминалу.
   2. Нажимает кнопку «Получить билет» → терминал вызывает метод issueTicket().
   3. Терминал регистрирует новый билет в **ИнформационнойСистеме** (метод зарегистрироватьСеанс(сеанс)), указывая время въезда и привязку к билету.
   4. Система выдаёт сигнал на Шлагбаум → открыть().
   5. Водитель забирает билет, шлагбаум поднимается, въезжает на парковку.
2. **«Оплата парковки»**
   1. Перед выездом водитель находит **ПлатёжныйТерминал**.
   2. Вставляет/сканирует билет (метод scanTicket()), терминал получает информацию о времени стоянки из ИнформационнойСистемы.
   3. Водитель выбирает способ оплаты: **наличными** (acceptCash()) или **картой** (acceptCard()).
   4. Если оплата прошла успешно, система помечает билет как «оплаченный».
   5. Терминал распечатывает чек или просто подтверждает оплату.
3. **«Выезд с парковки (валидный билет)»**
   1. Водитель подъезжает к **ВыездномуТерминалу**.
   2. Сканирует билет (validateTicket()), терминал отправляет запрос в систему: проверитьОплату(билет).
   3. Если билет оплачен, система даёт команду командаОткрыть(шлагбаум).
   4. Шлагбаум открывается, водитель покидает парковку.
4. **«Выезд с парковки (неоплаченный или просроченный билет)»**
   1. Водитель сканирует билет.
   2. Система видит, что плата не внесена или билет просрочен. Выдаётся сообщение: «Оплатите, пожалуйста!»
   3. Шлагбаум **не** открывается, водитель вынужден вернуться к платёжному терминалу для погашения долга.

### Сценарии использования для ****Администратора****

1. **«Управление тарифами»**
   1. Администратор авторизуется в интерфейсе **ИнформационнойСистемы**.
   2. Открывает раздел «Тарифы».
   3. Добавляет новый тариф (метод добавить объект Тариф в базаТарифов), либо редактирует существующий.
   4. Сохраняет изменения → система начинает применять новый тариф ко всем новым сеансам.
2. **«Просмотр отчётов»**
   1. Администратор выбирает период (дата/время).
   2. Система агрегирует данные (из базаСеансов, базаБилетов) и формирует отчёт: количество заездов, общая сумма оплат, пиковые часы загрузки.
   3. Администратор просматривает отчёт в виде таблицы или графика.
3. **«Ручное вмешательство (аварийное открытие шлагбаума)»**
   1. Администратор обнаруживает, что, например, терминал не работает.
   2. В интерфейсе вручную отправляет командаОткрыть(шлагбаум) без проверки билета.
   3. Система записывает факт аварийного вмешательства в журнал.

### Сценарии использования для ****Сервис-инженера****

1. **«Диагностика терминала»**
   1. Сервис-инженер подключается к сервисному порту терминала.
   2. Запрашивает журнал ошибок (лог неисправностей) — проверяет корректность датчиков/сканера/приёмника денег.
   3. Если найдена неисправность, инженер заменяет деталь (сканер, платёжный модуль и т.д.).
   4. Проводится тестовый запуск, терминал снова доступен для пользователей.
2. **«Плановое обслуживание шлагбаума»**
   1. Сервис-инженер вручную блокирует шлагбаум (чтобы никто не проезжал).
   2. Проверяет датчики, механизмы, двигатель. Смазывает движущиеся части.
   3. Обновляет прошивку (при необходимости).
   4. Снимает блокировку, проверяет правильность срабатывания («открыть/закрыть») несколько раз подряд.
   5. При нормальном функционировании инженер завершает обслуживание, отмечает в журнале работы.
3. **«Обновление ПО системы»**
   1. Сервис-инженер заходит на сервер **ИнформационнойСистемы**.
   2. Устанавливает новую версию ПО, переносит данные (сеансы, тарифы, билеты).
   3. Перезапускает службу системы.
   4. Проверяет, что терминалы на парковке успешно соединяются и обмениваются данными с сервером.