

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики»**



Факультет Программной Инженерии и Компьютерной Техники

## **Лабораторная работа №1**

по дисциплине

**«Архитектура программных систем»:**

Выполнили:  
Ястребов-Амирханов Алекси  
Группа: Р3332  
ПРЕПОДАВАТЕЛЬ: Перл И.А.

Санкт-Петербург,

2025

## Оглавление

Задание .....	3
1 Рассматриваемая система.....	4
2 Спецификация требований.....	5
3 UML-диаграммы .....	7
4 Сценарии использования.....	12
Вывод .....	15

## 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.

Отчёт по работе должен содержать:

1. Титульный лист с указанием автора и номера группы
2. Само задание
3. Описание рассматриваемой системы с требованиями к ней
4. Формальное описание системы с необходимым количеством UML диаграмм
5. Словесное описание сценариев использования для рассматриваемых акторов

# 1 Рассматриваемая система

**Система:** Яндекс.Еда — сервис онлайн-заказа и доставки готовой еды. (<https://eda.yandex/>)

## Общее описание:

Яндекс.Еда — платформа, позволяющая пользователям просматривать меню ресторанов и кафе, оформлять заказы на доставку или самовывоз, оплачивать их онлайн и отслеживать статус выполнения в реальном времени. Система объединяет клиентские мобильные и веб-приложения, панель для ресторанов-партнёров, мобильное приложение для курьеров и диспетчерскую часть, обеспечивающую маршрутизацию и распределение заказов.

## Назначение:

Обеспечить удобный и надёжный процесс заказа еды: от выбора блюда в меню до передачи доставленного заказа клиенту, при этом минимизировать время ожидания и максимально автоматизировать взаимодействие между клиентом, рестораном и курьером.

## Основные акторы:

- **Гость:** просматривает доступные рестораны и меню, изучает акции и предложения без необходимости авторизации.
- **Авторизованный пользователь:** оформляет заказы на доставку еды, оплачивает их, отслеживает статус выполнения заказа и оставляет отзывы.
- **Ресторан:** публикует и обновляет меню, принимает и обрабатывает заказы, готовит блюда и передаёт готовый заказ курьеру.
- **Курьер:** принимает заказы на доставку, забирает готовые блюда из ресторана и доставляет их клиенту, обновляя статус заказа в системе.
- **Администратор** — управляет каталогом ресторанов, промо-акциями, поддержкой и модерацией.

## Внешние системы и интеграции:

**Картографические сервисы** (прокладка маршрутов, расчёт времени доставки) — чаще всего Яндекс.Карты.

**Платёжные сервисы** (банковские карты, Apple/Google Pay, эквайринг, платёжные агрегаторы).

**Сервисы уведомлений** (push, SMS, e-mail).

**POS/ERP ресторанов** — интеграция с внутренними системами партнёров для синхронизации статусов и меню.

**Сервисы учёта и аналитики** (включая внутренние сервисы Яндекса).

## Операционная среда:

Платформа работает через клиентские приложения для мобильных устройств (iOS, Android) и веб-интерфейс (браузеры Chrome, Firefox, Safari, Edge). Серверная часть развёрнута в облачной/дата-центральной инфраструктуре с API-эндпоинтами для мобильных приложений, партнёров и курьеров. Для работы требуется доступ в Интернет; геолокация клиента/курьера используется для расчёта маршрутов и ETA.

## 2 Спецификация требований

### 2.1 Функциональные требования

#### Для пользователя (клиента)

1. **Регистрация и вход:** регистрация/авторизация по e-mail/телефону, вход через СМС-код, сохранение профиля с адресами доставки.
2. **Просмотр ресторанов и меню:** фильтрация (по кухне, рейтингу, времени доставки), сортировка (по скорости, цене, популярности), просмотр карточки блюда с составом и ценой.
3. **Оформление заказа:** сбор корзины, выбор адреса и способа доставки, выбор времени (сейчас/по расписанию), выбор способа оплаты (онлайн/наличные).
4. **Оплата:** интеграция с платёжными шлюзами (банковские карты, СБП, электронные кошельки) и подтверждение оплаты.
5. **Отслеживание статуса заказа:** отображение шагов (принят рестораном → готовится → курьер в пути → доставлен), в т.ч. реальное местоположение курьера на карте.
6. **Уведомления:** push и SMS-уведомления о смене статуса заказа, про подтверждение/отмену, о времени прибытия курьера.
7. **Оценки и отзывы:** возможность оценить ресторан/курьера и оставить отзыв после доставки.
8. **Избранное и история:** сохранение любимых ресторанов/блюд и история предыдущих заказов.
9. **Промокоды и акции:** применение скидок и бонусов в процессе оформления заказа.
10. **Поддержка:** встроенный чат / контакт-центр для вопросов по заказу и возвратам.

#### Для ресторана (партнёра)

11. **Панель ресторана:** приём заказов, подтверждение времени приготовления, отметка готовности, просмотр статистики (количество заказов, рейтинг).
12. **Менеджмент меню:** добавление/удаление блюд, управление ценами, наличие позиций, изображения и описания.
13. **Управление расписанием:** рабочие часы, время подготовки, отключение при высокой загрузке.

#### Для курьера

14. **Панель/приложение курьера:** приём задания, навигация к ресторану и к клиенту, подтверждение получения/доставки, смена статуса (в пути, забрал, доставил).
15. **Маршрутизация:** оптимальные маршруты, расчёт ETA, поддержка нескольких заказов (пул).

#### Для администратора (владельца сервиса)

16. **Админ-панель:** управление пользователями/ресторанными аккаунтами, мониторинг заказов, возвратов, аналитика, модерация отзывов.
17. **Интеграции:** подключение/настройка платёжных провайдеров, картографических сервисов и услуг уведомлений.
18. **Отчётность:** выгрузки по продажам, комиссиям, KPI партнёров.

## 2.2 Нефункциональные требования

### 2.2.1 Требования к удобству использования (Usability)

1. Интерфейс должен быть интуитивно понятным: пользователь должен выполнить оформление заказа не более чем в **3 клика/экрана** (не считая ввода данных).
2. Поддержка адаптивного дизайна: корректное отображение на мобильных телефонах (типичные разрешения 360×780, 414×896) и на десктопах (1366×768, 1920×1080).
3. Быстрый доступ к избранному и истории заказов из главного меню.
4. Поддержка локализации интерфейса (русский; при необходимости — английский).
5. Доступность основных функций в офлайн-режиме ограничена (например, просмотр сохранённых меню) с явным уведомлением о недоступности опций требующих сеть.

### 2.2.2 Требования к надёжности (Reliability / Availability)

1. **Доступность сервиса:**  $\geq 99.9\%$  времени (не более ~8 часов простоя в год).
2. Резервное копирование транзакционных данных и заказов — не реже **1 раза в сутки**; критические журналы — репликация в реальном времени.
3. Время восстановления после критической ошибки (RTO) — не более **1 часа**; среднее время на устранение инцидента —  $\leq 30$  минут для критических инцидентов.
4. Транзакции оплаты должны быть атомарны: либо платёж успешно обработан и заказ создан, либо операция откатывается без частичных состояний.

### 2.2.3 Требования к производительности (Performance / Scalability)

1. Время загрузки главной страницы приложения при стандартном соединении (10 Мбит/с) — **< 2 секунд**.
2. Время ответа API для ключевых операций (создание заказа, проверка статуса) — **< 300 мс** при среднемесячной нагрузке.
3. Поддержка одновременной активности **не менее 100 000** пользователей (пиковые нагрузки) с возможностью автошардинга/горизонтального масштабирования.
4. Обновление координат курьера и ETA — не реже чем раз в **10–15 секунд** в реальном времени.
5. Хранение и трансляция изображений/медиа — использование форматов сжатия (WebP для изображений, HLS для видео/преформатов), CDN для статики.
6. Лимит времени отклика для страниц оформления заказа — **< 1.5 сек**.

### 2.2.4 Требования к безопасности (Security)

1. Передача всех пользовательских данных только по **HTTPS** (TLS 1.2+).
2. Соответствие стандартам при обработке платежей (соответствие PCI DSS для обработки карт) — при обработке карт через платёжные шлюзы.
3. Защита от типичных уязвимостей: SQL-инъекции, XSS, CSRF, инъекции команд.
4. Механизмы защиты от DDoS (WAF, rate-limiting, CDN).
5. Шифрование персональных данных пользователя в покое (at-rest) — имена/адреса/телефоны хранятся зашифрованными.
6. Многоуровневая система прав доступа: роль-базированный доступ (пользователь, ресторанный менеджер, курьер, администратор).
7. Логи безопасности и аудит действий администратора/сотрудников; хранение логов минимум 90 дней.
8. Механизм обнаружения мошенничества (антифрод) при подозрительных платежах/возвратах.

### 3 UML-диаграммы

Use-Case диаграмма системы

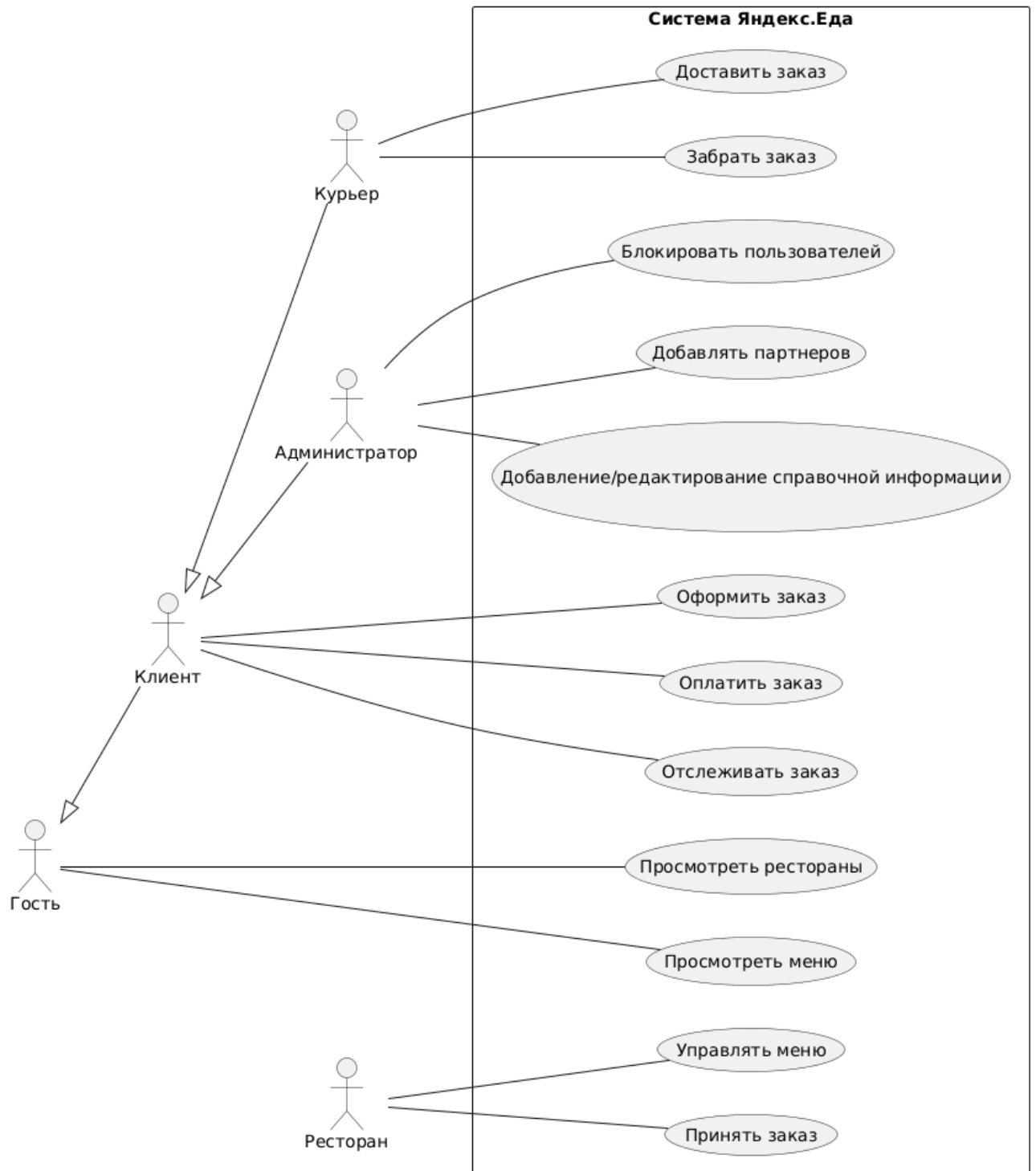


Диаграмма классов

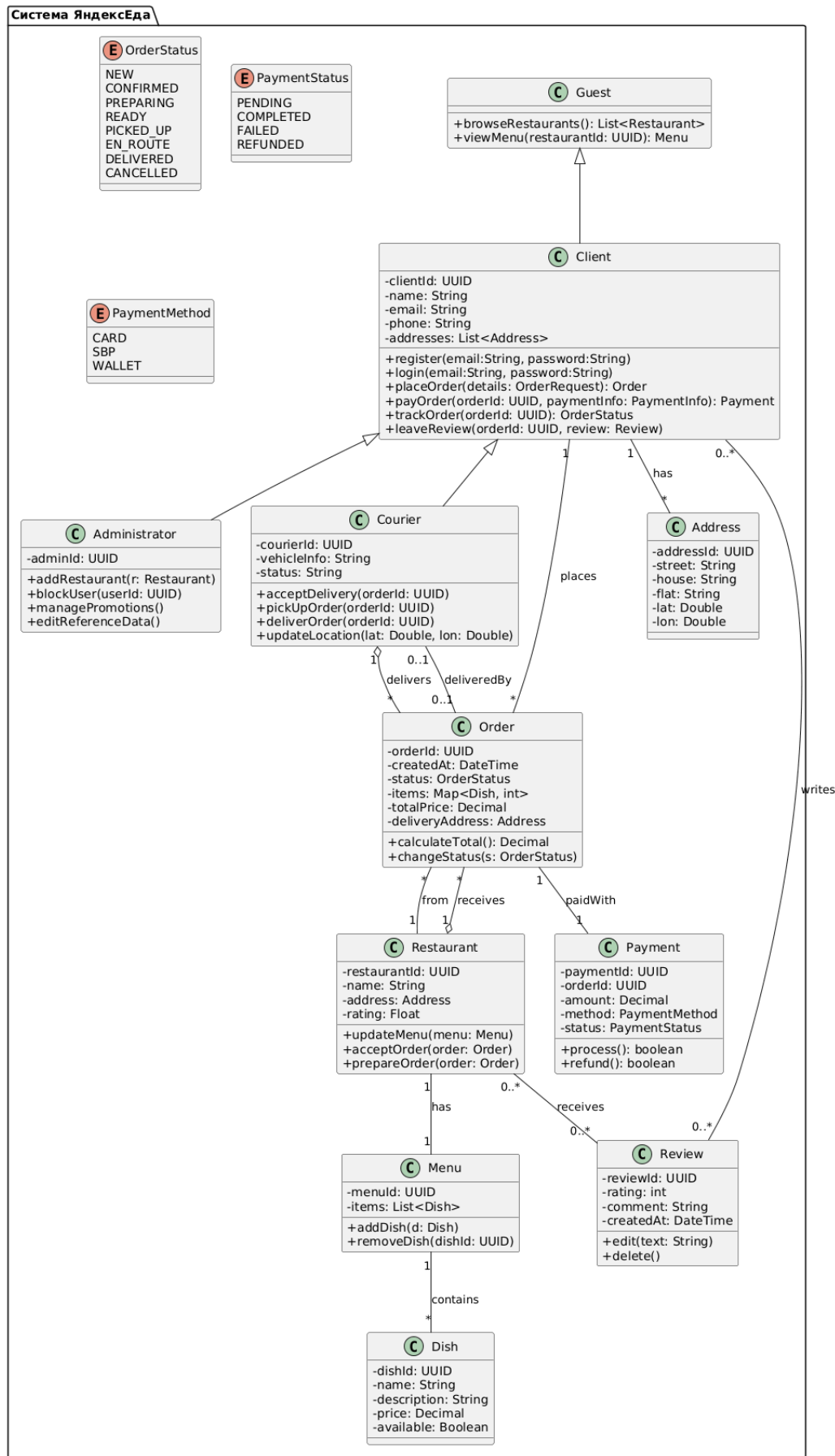
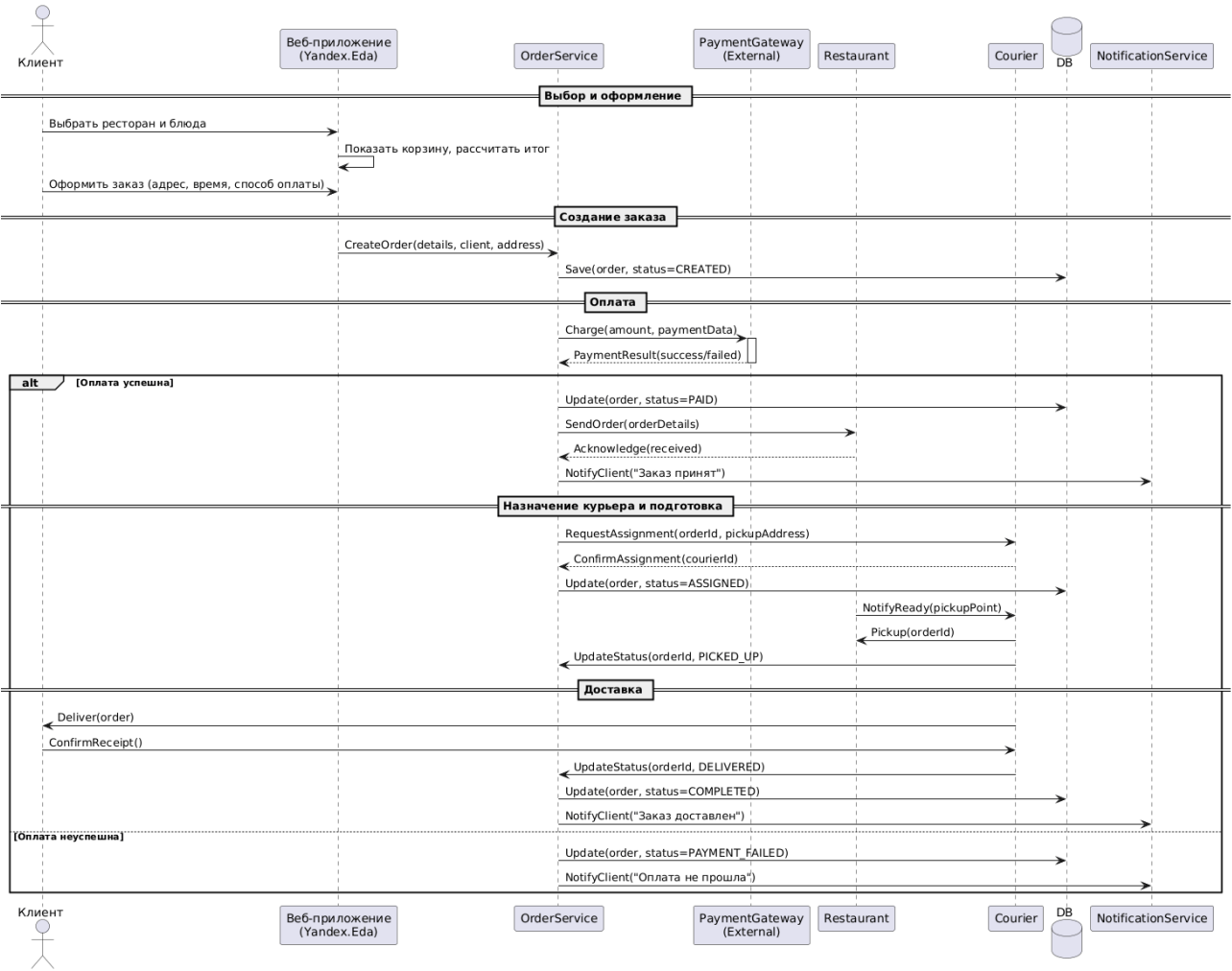




Диаграмма последовательности (Sequence Diagram)



Activity Diagram — «Регистрация пользователя»

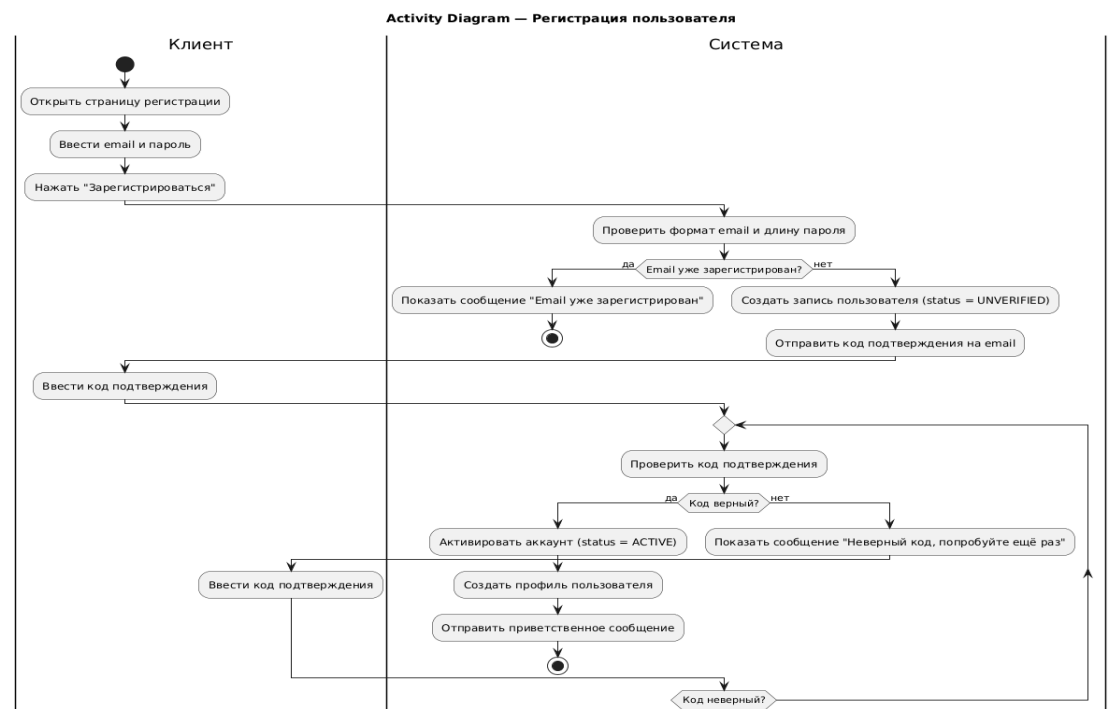
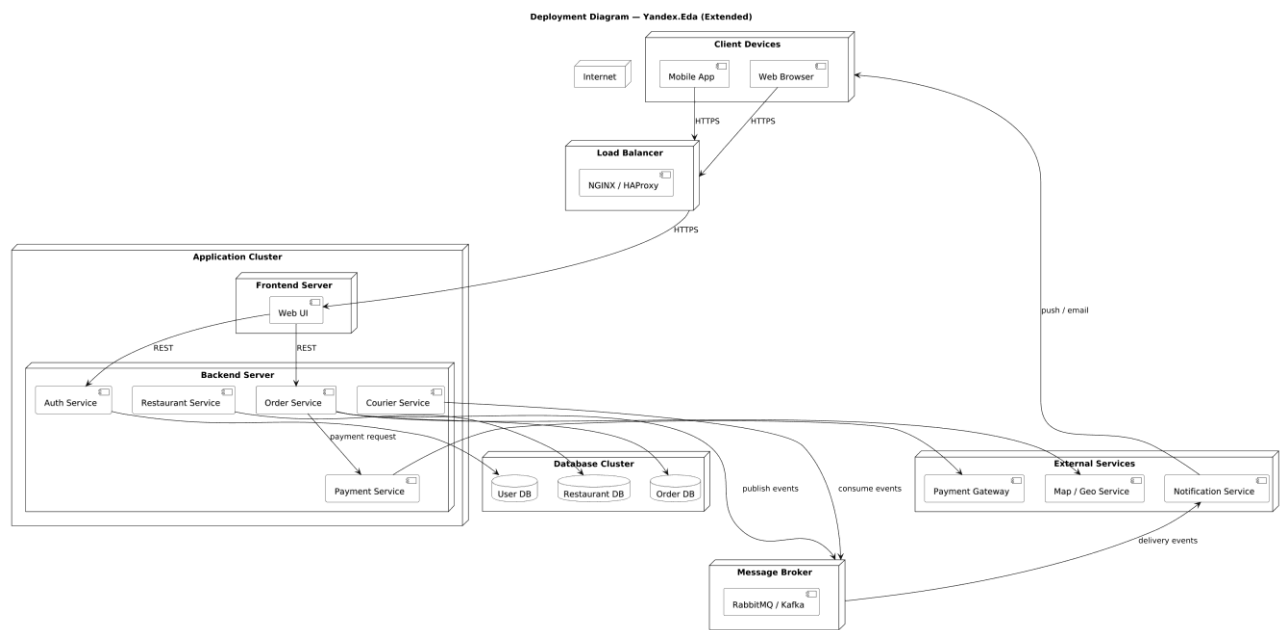
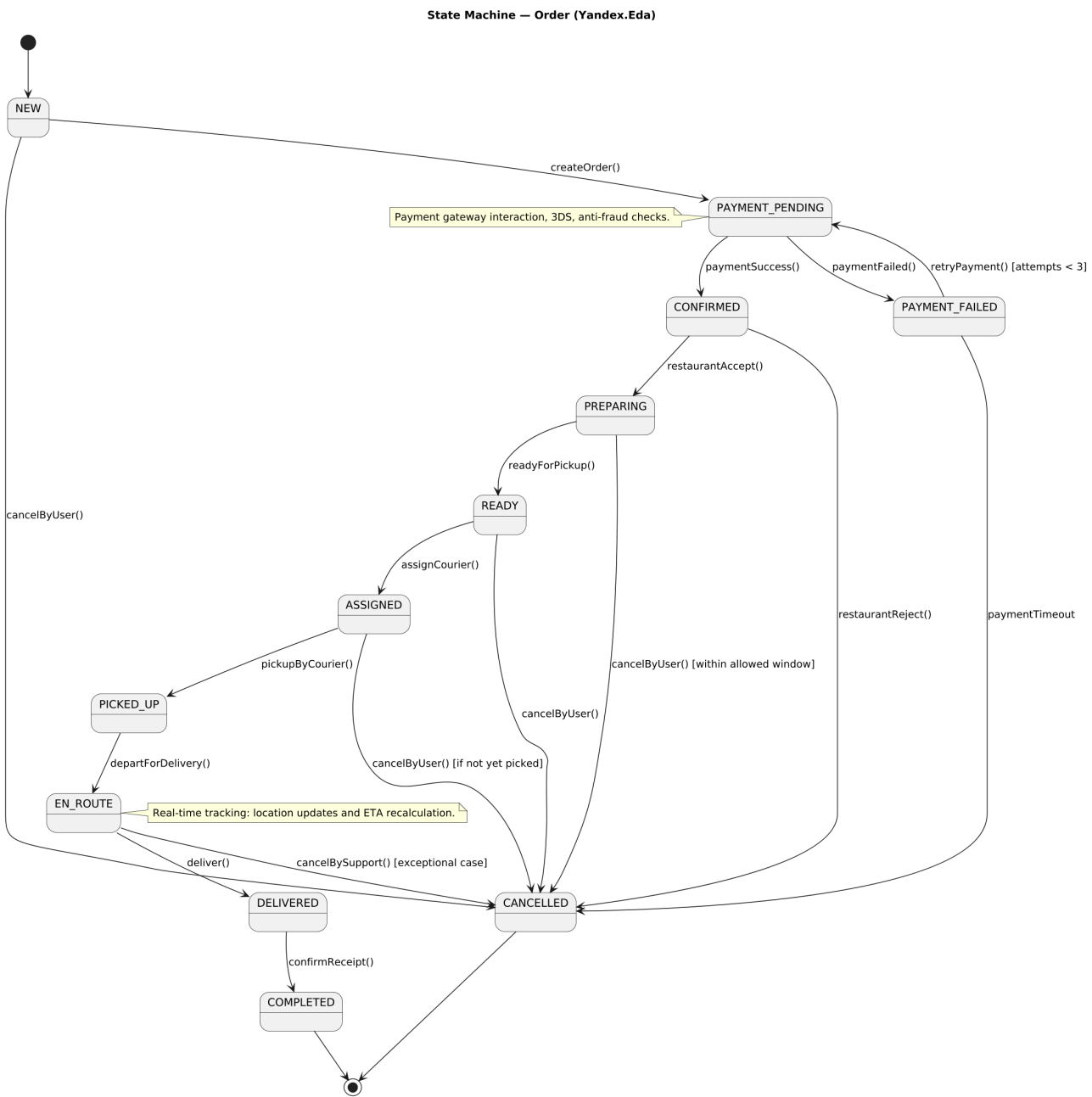


Диаграмма развертывания



State diagram — жизненный цикл заказа



## 4 Сценарии использования

<b>ID</b>	1
<b>Краткое описание</b>	Просмотр списка доступных ресторанов, применение фильтров и открытие карточки ресторана.
<b>Главный актер</b>	Гость (также доступно Авторизованному пользователю)
<b>Предусловие</b>	В системе имеется каталог ресторанов с метаданными (кухня, рейтинг, время доставки).
<b>Основной поток</b>	<ol style="list-style-type: none"><li>1. Пользователь открывает список ресторанов или выполняет поиск по адресу/радиусу.</li><li>2. Система показывает список ресторанов с базовой информацией (рейтинг, среднее время доставки, минимальная сумма).</li><li>3. Пользователь применяет фильтры/сортировку (по кухне, рейтингу, цене, времени).</li><li>4. Пользователь выбирает ресторан и открывает его карточку с меню и подробной информацией.</li></ol>
<b>Альтернативный поток</b>	<ol style="list-style-type: none"><li>1. Если по заданным фильтрам нет результатов, система предлагает расширить фильтры или показать рекомендуемые рестораны.</li><li>2. Пользователь выбирает рекомендованный ресторан либо меняет фильтры и открывает карточку.</li></ol>
<b>Постусловия</b>	Просмотренный ресторан фиксируется в истории; счётчик просмотров увеличен.

<b>ID</b>	2
<b>Краткое описание</b>	Регистрация нового пользователя или вход в существующий аккаунт (телефон, e-mail, соцсеть, SMS-код).
<b>Главный актер</b>	Клиент (Авторизованный пользователь)
<b>Предусловие</b>	Доступна служба аутентификации и подтверждения (SMS/email/OAuth).
<b>Основной поток</b>	<ol style="list-style-type: none"><li>1. Пользователь выбирает способ регистрации/входа.</li><li>2. Пользователь вводит данные (телефон/email) и подтверждает действие (получает код или проходит OAuth).</li><li>3. Система проверяет код/данные и создаёт аккаунт или выполняет вход.</li><li>4. Система направляет пользователя в личный кабинет.</li></ol>
<b>Альтернативный поток</b>	<ol style="list-style-type: none"><li>1. Если введён некорректный код, система показывает ошибку и предлагает повторную отправку кода (с ограничением по попыткам).</li><li>2. Если email/телефон уже зарегистрирован, система предлагает восстановление доступа или вход.</li></ol>
<b>Постусловия</b>	Аккаунт активирован/пользователь вошёл; создан профиль с базовыми настройками.

<b>ID</b>	3
<b>Краткое описание</b>	Формирование корзины, выбор адреса/способа доставки и подтверждение заказа.
<b>Главный актер</b>	Клиент

<b>Предусловие</b>	Выбран ресторан; доступны позиции меню; указан адрес доставки или пользователь вводит его при оформлении.
<b>Основной поток</b>	<ol style="list-style-type: none"> <li>1. Клиент добавляет блюда в корзину и проверяет состав заказа.</li> <li>2. Клиент выбирает адрес доставки или самовывоз, указывает время (сейчас/по расписанию).</li> <li>3. Клиент применяет промокод (при наличии) и сверяет итоговую сумму.</li> <li>4. Клиент подтверждает заказ — система создаёт заказ со статусом CREATED / PAYMENT_PENDING.</li> <li>5. Система уведомляет ресторан о новом заказе и показывает ожидаемый статус клиенту.</li> </ol>
<b>Альтернативный поток</b>	<ol style="list-style-type: none"> <li>1. Если в корзине есть недоступные блюда, система уведомляет клиента и предлагает удалить или заменить позицию.</li> <li>2. Если ресторан закрыт или не может принять заказ, система предлагает изменить время или выбрать другой ресторан.</li> </ol>
<b>Постусловия</b>	Заказ создан; клиенту показан номер заказа и текущий статус; инициирована оплата при необходимости.

<b>ID</b>	4
<b>Краткое описание</b>	Проведение онлайн-платежа по заказу через платёжный шлюз и обработка результатов транзакции.
<b>Главный актер</b>	Клиент
<b>Предусловие</b>	Заказ существует со статусом PAYMENT_PENDING; выбран способ оплаты.
<b>Основной поток</b>	<ol style="list-style-type: none"> <li>1. Клиент выбирает способ оплаты (карта, SBP, кошелёк).</li> <li>2. Система инициирует платёж в платёжный шлюз (включая 3DS при необходимости).</li> <li>3. Платёжный шлюз возвращает результат; при успехе система обновляет статус заказа на PAID.</li> <li>4. Система уведомляет ресторан/курьера о подтверждении оплаты.</li> </ol>
<b>Альтернативный поток</b>	<ol style="list-style-type: none"> <li>1. При ошибке оплаты система сообщает об ошибке и предлагает повторить платёж или выбрать другой способ.</li> <li>2. При повторных неудачных попытках заказ переводится в PAYMENT_FAILED и клиенту предлагается отмена или помощь поддержки.</li> </ol>
<b>Постусловия</b>	При успехе — заказ помечен PAID; при неудаче — PAYMENT_FAILED и сохранён лог транзакции.

<b>ID</b>	5
<b>Краткое описание</b>	Отслеживание статуса заказа и местоположения курьера в реальном времени; получение уведомлений о смене статуса.
<b>Главный актер</b>	Клиент
<b>Предусловие</b>	Заказ создан и находится в статусе, начиная с ASSIGNED / PICKED_UP / EN_ROUTE; у курьера включена передача геопозиции.
<b>Основной поток</b>	1. Клиент открывает страницу отслеживания или получает push-уведомление.

	<p>2. Система отображает текущий статус заказа и карту с местоположением курьера и ETA.</p> <p>3. Система отправляет push/SMS-уведомления при смене ключевых статусов.</p>
<b>Альтернативный поток</b>	<p>1. Если курьер ещё не назначен, система показывает 'Ожидание курьера' и предлагает связаться с поддержкой при задержке.</p> <p>2. В случае отмены заказчику показывается причина отмены и инструкции по возврату средств.</p>
<b>Постусловия</b>	Клиент информирован о текущем статусе; история статусов сохраняется в системе.

<b>ID</b>	6
<b>Краткое описание</b>	Добавление и редактирование позиций меню партнёром (рестораном) через партнёрскую панель.
<b>Главный актер</b>	Ресторан (Партнёр)
<b>Предусловие</b>	У ресторана есть активный партнёрский аккаунт и доступ в панель управления.
<b>Основной поток</b>	<p>1. Представитель ресторана входит в партнёрскую панель.</p> <p>2. Переходит в раздел 'Меню' и выбирает 'Добавить блюдо' или 'Редактировать позицию'.</p> <p>3. Вводит название, описание, цену, время приготовления, категорию и загружает фото.</p> <p>4. Сохраняет изменения — система валидирует данные и публикует обновление.</p>
<b>Альтернативный поток</b>	<p>1. При некорректных данных или слишком большом изображении система показывает сообщение об ошибке.</p> <p>2. Ресторан исправляет данные и повторно сохраняет изменения.</p>
<b>Постусловия</b>	Меню обновлено; изменения отражаются в клиентских приложениях с учётом кэширования.

<b>ID</b>	7
<b>Краткое описание</b>	Курьер принимает задание, забирает заказ из ресторана и доставляет клиенту с обновлением статусов.
<b>Главный актер</b>	Курьер
<b>Предусловие</b>	Курьер авторизован в курьерском приложении; заказ назначен или доступен в пуле заданий.
<b>Основной поток</b>	<p>1. Курьер получает уведомление о новом задании и просматривает детали.</p> <p>2. Курьер принимает задание (назначение подтверждается системой).</p> <p>3. Курьер приезжает в ресторан, подтверждает 'Прибыл' и получает заказ (скан/фото/подпись).</p> <p>4. Курьер помечает заказ как PICKED_UP и начинает движение к клиенту (передаёт геопозицию).</p> <p>5. Курьер передаёт заказ клиенту; клиент подтверждает получение; курьер отмечает DELIVERED.</p>
<b>Альтернативный поток</b>	<p>1. Если курьер не может принять задачу, он отклоняет её — заказ перепубликуется в пул другим курьерам.</p> <p>2. Если клиент недоступен, курьер связывается с</p>

	клиентом/диспетчером; в зависимости от политики назначается повторная попытка или возврат.
<b>Постусловия</b>	Заказ доставлен и закрыт или зафиксирован инцидент с отметкой причины и дальнейшими действиями.

## Вывод

В ходе выполнения лабораторной работы была спроектирована и проанализирована информационная система онлайн-сервиса доставки еды на примере платформы «Яндекс.Еда». В рамках работы были определены основные акторы системы (гость, клиент, ресторан, курьер, администратор) и их роли во взаимодействии с системой.

Были разработаны и описаны ключевые сценарии использования, отражающие основные бизнес-процессы сервиса: просмотр ресторанов, регистрация и аутентификация пользователей, оформление и оплата заказа, отслеживание доставки, управление меню со стороны ресторана, а также процесс доставки заказа курьером. Каждый сценарий был формализован с указанием предусловий, основного и альтернативных потоков, а также постусловий, что позволило детально описать поведение системы в различных ситуациях.

Дополнительно была построена диаграмма состояний жизненного цикла заказа, которая наглядно демонстрирует переходы между статусами заказа от момента его создания до завершения или отмены. Использование UML-диаграмм и текстовых сценариев обеспечило целостное представление архитектуры и логики работы системы.

В результате выполненной работы были получены практические навыки моделирования требований к программной системе, анализа взаимодействия акторов с системой и формализации бизнес-процессов с использованием UML.