Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра «Информационные системы и технологии»

Направление подготовки/специальность: 09.03.02 Информационные системы и технологии

ОТЧЕТ

по проектной практике

Студент: Исмаилзаде Иса Намиг оглы\_\_\_Группа: 241-332

Место прохождения практики: Московский Политех, Кафедра информационных технологий

Отчет принят с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики: Худайбердиева Гулшат.

Москва 2025

**Оглавление**

[**ВВЕДЕНИЕ** 2](#_Toc198730893)

[Общая информация о проекте 2](#_Toc198730894)

[Общая характеристика заказчика проекта 3](#_Toc198730895)

[Описание задания по проектной практике 4](#_Toc198730896)

[Описание достигнутых результатов по проектной практике 7](#_Toc198730897)

[**ЗАКЛЮЧЕНИЕ** 11](#_Toc198730898)

# **ВВЕДЕНИЕ**

Сегодня одним из ключевых критериев при найме сотрудников со стороны работодателей является практический опыт. Данный проект предоставляет отличную возможность развить навыки в области разработки игр и получить ценный практический опыт, который будет востребован в профессиональной сфере.

## Общая информация о проекте

**Название проекта:**

Разработка игры с онлайн-прокачкой и оффлайн-встречей

**Цели и задачи проекта:**

Разработать концепцию и архитектуру программного обеспечения для игры, в которой игроки сначала прокачивают своих персонажей онлайн, а затем используют эти прокачанные персонажи в оффлайн-встречах

## Общая характеристика заказчика проекта

**Наименование заказчика:** ООО "Корпорация роботов"

**Организационная структура:** ООО "Корпорация роботов" делится на

* Робостанция. Постоянно действующий образовательно-просветительский центр, основной миссией которого является популяризация робототехники, инженерии и инновационных наук.
* Кулибин Про. Сеть инженерных интерактивных центров с открытыми мастерскими в формате «сделай сам» для детей и взрослых.
* Робоагенство. Компания по организации мероприятий с роботами, а также других интерактивных взаимодействий с новыми технологиями.
* Конструкторское бюро «Корпорации роботов». Инновационный инженерно-исследовательский центр, создающий передовые робототехнические решения для мероприятий и различных отраслей промышленности.
* Бал роботов / Робофорум. Интерактивная выставка роботов со всего мира. Впервые в истории современной России в одном месте собраны уникальные экспонаты из разных стран.
* Бронебот. Первые бои роботов БРОНЕБОТ. Были созданы по мотивам культовых мировых шоу 90-х — Robot Wars

**Описание деятельности:** Современная инновационная компания, объединяющая в себе образовательно-просветительскую, выставочно-развлекательную и инженерно-техническую деятельность. Является флагманом и лидером в сфере выставочной робототехники в России.

## Описание задания по проектной практике

Задание: Разработка игры с онлайн-прокачкой и оффлайн-встречей

Цель: Разработать концепцию и архитектуру программного обеспечения для игры, в которой игроки сначала прокачивают своих персонажей онлайн, а затем используют эти прокачанные персонажи в оффлайн-встречах

I. Концепция игры:

Выберите жанр игры и опишите его ключевые особенности. Примеры жанров:

• Коллекционная карточная игра (ККИ): Игроки собирают и улучшают карты персонажей онлайн, затем используют эти карты в физической ККИ.

• Ролевая игра (РПГ): Игроки создают и развивают персонажей онлайн, а затем используют их в оффлайн-сессиях под руководством ведущего.

• Спортивная игра: Игроки тренируют виртуальных спортсменов онлайн, а затем соревнуются с другими игроками в физической игре (например, гонки на дронах, робототехнические бои).

• Игра с элементами дополненной реальности (AR): Игроки прокачивают персонажей онлайн, затем сражаются с другими игроками в реальном мире, используя приложение дополненной реальности.

II. Архитектура ПО:

Предложите архитектуру программного обеспечения, включающую:

• Серверная часть:

Технологии (например, Node.js, Python с Django/Flask, Java с Spring).

База данных (например, PostgreSQL, MongoDB, MySQL).

API для взаимодействия с клиентской частью.

Системы аутентификации и авторизации пользователей.

Система управления игрой (гейм-дизайн, баланс, прогрессия).

Система хранения данных игроков (профили, персонажи, инвентарь).

Система интеграции с оффлайн-компонентом игры (если таковой есть).

• Клиентская часть (онлайн):

Технологии (например, React, Angular, Vue.js, Unity, Unreal Engine).

Интерфейс пользователя (UI/UX дизайн).

Взаимодействие с сервером через API.

Система отображения игрового процесса.

Система внутриигровой покупки (опционально).

• Оффлайн-компонент (при наличии):

Физические компоненты игры (карты, фигурки, игровое поле).

Приложение дополненной реальности (AR) (если применимо).

Возможность синхронизации данных с онлайн-частью игры.

III. Функциональные возможности:

Опишите функциональные возможности игры, включая:

• Онлайн-режим:

Создание и настройка профиля игрока.

Создание и прокачка персонажей.

Участие в онлайн-событиях и заданиях.

Взаимодействие с другими игроками (чат, торговля).

Система рейтингов и достижений.

• Оффлайн-режим:

Возможность использования прокачанных персонажей в оффлайн-играх.

Правила оффлайн-игры.

Интеграция с оффлайн-компонентами (если есть).

Возможность синхронизации данных с онлайн-частью игры после оффлайн-сессии (результаты, изменения персонажа).

IV. Технические требования и ограничения:

• Опишите технические требования к серверному и клиентскому оборудованию.

• Укажите ограничения по количеству игроков онлайн и оффлайн.

• Опишите предполагаемый масштаб проекта.

V. Презентация решения:

Представьте своё решение в виде документа, содержащего:

• Подробное описание выбранной концепции игры.

• Диаграмму архитектуры ПО.

• Описание основных функциональных возможностей.

• Описание технических требований и ограничений.

• План разработки (опционально).

Это задание позволит вам разработать полную концепцию игры, включая техническую сторону и гейм-дизайн. Не бойтесь экспериментировать с разными идеями и подходам

## Описание достигнутых результатов по проектной практике

Жанр игры: Коллекционная карточная игра (ККИ) с элементами ролевой игры (РПГ)

**Серверная часть:**

Java с Spring Boot: Для реализации серверной логики, REST API и управления бизнес-процессами.  
PostgreSQL: Основная реляционная база данных для хранения профилей игроков, карт, инвентаря и истории матчей.  
Redis: Для кэширования часто запрашиваемых данных (например, рейтингов игроков, активных сессий) и управления очередями сообщений.  
JWT (JSON Web Tokens): Для аутентификации и авторизации пользователей.  
WebSocket: Для real-time взаимодействия, такого как чат и уведомления о событиях.  
Docker: Для контейнеризации и упрощения деплоя.  
Kubernetes: Для оркестрации контейнеров и масштабирования.

**База данных:**

* **PostgreSQL**:
  + Таблицы:
    - users (профили игроков: id, username, email, password\_hash, created\_at).
    - cards (информация о картах: id, name, type, stats, owner\_id).
    - decks (колоды игроков: id, user\_id, card\_ids).
    - matches (история матчей: id, player1\_id, player2\_id, result, timestamp).
    - inventory (инвентарь: user\_id, resource\_type, amount).
  + Используется индексация для оптимизации запросов (например, по user\_id).
* **Redis**:
  + Кэширование текущих сессий, рейтингов игроков, активных событий.
  + Очереди для обработки асинхронных задач (например, обновление рейтингов после матчей).

**API:**

* RESTful API на Spring Boot:
  + /api/auth/register — регистрация пользователя.
  + /api/auth/login — аутентификация с выдачей JWT.
  + /api/cards — управление картами (создание, улучшение, просмотр).
  + /api/decks — управление колодами.
  + /api/matches — запись результатов матчей (онлайн и оффлайн).
  + /api/trade — торговля картами между игроками.
  + /api/events — участие в глобальных событиях.
* WebSocket для чата и уведомлений.

**Аутентификация и авторизация:**

* JWT-токены для безопасного доступа к API.
* Spring Security для управления ролями (игрок, администратор).
* Redis хранит активные токены для проверки сессий.

**Система управления игрой:**

* Балансировка характеристик карт через конфигурационные файлы (JSON/YAML).
* Алгоритмы подбора противников (matchmaking) на основе рейтинга.
* Система прогрессии: уровни персонажей, ресурсы, награды за задания.

**Интеграция с оффлайн-компонентом:**

* QR-коды на физических картах связывают их с цифровыми аналогами.
* API для синхронизации результатов оффлайн-матчей (через сканирование QR-кода в мобильном приложении).

**Клиентская часть:**

* **Технологии:**
  + **React**: Для веб-интерфейса с использованием TypeScript.
  + **Tailwind CSS**: Для адаптивного и современного дизайна.
  + **WebSocket**: Для real-time взаимодействия (чат, события).
  + **Axios**: Для взаимодействия с REST API.
* **UI/UX дизайн:**
  + Главное меню: профиль, коллекция карт, колоды, события, чат.
  + Интерактивный редактор колод с drag-and-drop.
  + Визуализация матчей с анимацией действий карт.
  + Адаптивный дизайн для мобильных и десктопных устройств.
* **Взаимодействие с сервером:**
  + Запросы к REST API для управления профилем, картами, колодами.
  + WebSocket для чата и уведомлений о событиях.
* **Внутриигровые покупки (опционально):**
  + Покупка косметических предметов (скины для карт, эффекты).
  + Ускорители прогресса (дополнительные ресурсы).
  + Интеграция с платёжными системами (например, Stripe).

##### Оффлайн-компонент

* **Физические компоненты:**
  + Физические карты с уникальными QR-кодами, генерируемыми на основе цифровых карт.
  + Игровое поле с зонами для карт (поле боя, рука, сброс).
* **Мобильное приложение (AR, опционально):**
  + Технологии: React Native + ARKit/ARCore.
  + Функции: сканирование QR-кодов для синхронизации, визуализация битв в AR.
* **Синхронизация:**
  + После оффлайн-матча игроки сканируют QR-коды карт через приложение, чтобы обновить результаты в онлайн-профиле.

**Функциональные возможности**

##### Онлайн-режим

* **Создание и настройка профиля:**
  + Регистрация/вход через email и пароль.
  + Настройка аватара, имени, предпочтений.
* **Создание и прокачка персонажей:**
  + Игроки получают стартовую колоду карт.
  + Прокачка карт через ресурсы (золото, кристаллы), получаемые за задания.
  + Разблокировка новых карт через достижения или покупки.
* **Онлайн-события и задания:**
  + Ежедневные задания (например, "Выиграть 3 матча").
  + Глобальные события (например, "Турнир гильдий").
* **Взаимодействие с игроками:**
  + Чат (глобальный, гильдийский, приватный).
  + Торговля картами через безопасную систему предложений.
* **Рейтинги и достижения:**
  + Глобальный рейтинг игроков.
  + Достижения за выполнение заданий и победы.

##### Оффлайн-режим

* **Использование персонажей:**
  + Игроки используют физические карты, созданные на основе цифровых.
  + Правила: дуэли 1v1, пошаговые сражения, основанные на характеристиках карт.
* **Интеграция:**
  + QR-коды на картах связывают их с онлайн-профилем.
  + Мобильное приложение сканирует результаты матчей и отправляет их на сервер.
* **Синхронизация:**
  + Обновление рейтинга и инвентаря после оффлайн-матчей.
  + Возможность разблокировки наград за оффлайн-игры.

**Технические требования для сервера**

**Серверное оборудование:**

* Минимальные требования: 4 vCPU, 8 GB RAM, 100 GB SSD.
* Рекомендуемые: 8 vCPU, 16 GB RAM, 500 GB SSD для масштабирования.
* Облачная платформа (AWS/GCP) для высокой доступности.

**Клиентское оборудование:**

* Веб: Современный браузер (Chrome, Firefox, Edge).
* Мобильное приложение: iOS 15+ / Android 10+.

**Оффлайн-компонент:**

* Физические карты: стандартный размер (63.5x88 мм).
* Мобильное приложение: камера для сканирования QR-кодов.

**Ограничения:**

* Онлайн: до 10,000 одновременных игроков на старте (масштабируется через Kubernetes).
* Оффлайн: до 100 игроков на одном турнире (ограничение по организации).

**Масштаб проекта:**

* Средний: команда из 5-10 разработчиков, 6-12 месяцев на MVP.

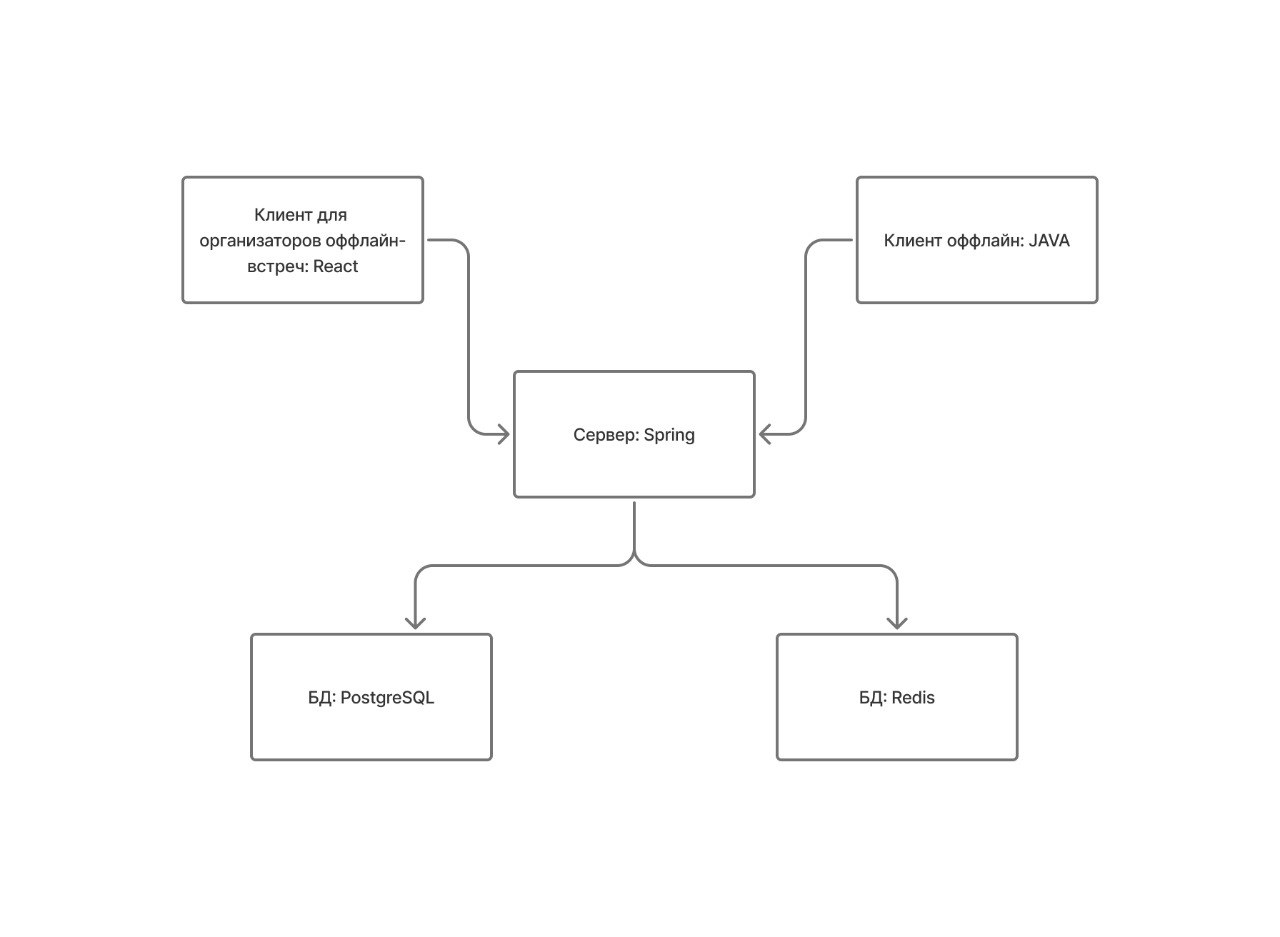


Рисунок 1 (Архитектура приложения)

# **ЗАКЛЮЧЕНИЕ**

Во время проделанной работы были получены опыт и навыки разработки концепции игр с уникальными механиками (онлайн- и оффлайн-взаимодействие), что будет полезно в дальнейшей профессиональной деятельности. Заказчик получил четкое представление о конечном продукте и может приступить к его разработке на основе предложенного плана.