

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра информационных технологий управления

Курсовой проект

по дисциплине

Технологии программирования

09.03.02 Информационные системы и технологии

Информационные системы и технологии управления предприятием

Зав. кафедрой	_____	М. Г. Матвеев, д.т.н., профессор
Обучающийся	_____	О. А. Ревенко, ст. 3 курса оч. отд.
Обучающийся	_____	К. Н. Щербаков, ст. 3 курса оч. отд.
Обучающийся	_____	Л. А. Челяпин, ст. 3 курса оч. отд.
Руководитель	_____	А. В. Москаленко, преподаватель
Руководитель	_____	В.С. Тарасов, ст. преподаватель _____.20__

Воронеж 2025

## Содержание

Содержание.....	2
Введение.....	4
1 Постановка задач.....	6
1.1 Требования к разрабатываемой системе.....	6
1.1.1 Функциональные требования .....	6
1.1.1.1 Для неавторизованных пользователей.....	7
1.1.1.2 Для авторизованных пользователей.....	7
1.1.2 Нефункциональные требования .....	8
1.1.3 Технические требования .....	8
1.2 Требования к интерфейсу .....	9
1.3 Задачи, решаемые в процессе разработки .....	10
2 Анализ предметной области .....	11
2.1 Терминология (гlossарий) предметной области .....	11
2.2 Обзор аналогов .....	14
2.2.1 Финансовый тренажёр (CashGo).....	14
2.2.2 Симулятор инвестора (BuildYourStax) .....	14
2.2.3 Симулятор инвестора (от medoborodyj) .....	15
2.3 Моделирование системы .....	17
2.3.1 Диаграмма прецедентов .....	17
2.3.2 Диаграмма последовательности.....	18
2.3.3 Диаграмма состояний .....	21
2.3.4 Диаграмма активностей .....	21
2.3.5 Диаграмма классов .....	23
2.3.6 ER-диаграмма .....	23
2.3.7 Диаграмма развёртывания .....	25
2.3.8 Диаграмма в стиле методологии IDEF0 .....	26
2.3.9 Диаграмма объектов .....	28
2.4 Аналитика веб-приложения .....	29
2.5 Границы проекта.....	30
3 Реализация .....	31

3.1 Средства реализации .....	31
3.2 Реализация серверной части веб-приложения .....	31
3.3 Реализация клиентской части веб-приложения .....	33
3.4 Навигация по сайту .....	35
3.4.2 Для авторизованного .....	37
4 Тестирование .....	42
4.1 Ручное UI–тестирование.....	42
Заключение .....	46
Список используемых источников.....	47

## **Введение**

В настоящее время цифровые технологии играют ключевую роль в образовании и профессиональном развитии. Одним из эффективных способов обучения являются интерактивные игры, которые позволяют пользователям приобретать теоретические и практические навыки в развлекательном формате, который людям легче воспринимать. В частности, это подходит для такой сложной и важной сферы, как инвестирование в стартапы, где важно уметь анализировать риски, принимать стратегические решения, которые могут понести за собой значительные последствия, и адаптироваться к стремительно изменяющимся условиям рынка.

Инвестирование достаточно сложная сфера, требующая значительного изначального опыта и капитала, из-за чего традиционные методы обучения, такие как лекции или книги, не всегда в полной мере позволяют освоить практические аспекты этой деятельности. Следовательно, интерактивные игры становятся мощным инструментом, который помогает приобрести общее представление о сфере и на начальном этапе развить базовые навыки.

В данной курсовой работе рассматривается разработка веб-приложения TechTrackInvest – интерактивной игры, сочетающей элементы стратегического планирования и случайных событий. Основная цель проекта – создать систему, которая позволит пользователям в игровой форме освоить принципы инвестирования в стартапы, научиться управлять ресурсами, оценивать доступные решения, продумывать влияние каждого из них и осознанно делать выборы. Игра также включает динамические события, такие как кризисы, что позволит развить умение адаптироваться к стремительно меняющимся ситуациям.

В работе будет детально рассмотрен процесс создания веб-приложения - от проектирования архитектуры системы и разработки игровой механики до реализации функционала для пользователей. Особое внимание уделено аспектам безопасности данных и масштабируемости решения, что позволит

системе стабильно работать при увеличении числа участников. Проект сочетает в себе образовательную ценность с элементами развлечения.

## **1 Постановка задач**

Целью данного проекта является разработка веб-приложения для интерактивной игры "TechTrackInvest", которая моделирует процесс инвестирования в стартапы с использованием механик случайных событий. Основные задачи проекта включают реализацию функционала для анализа инвестиционных возможностей, управления ресурсами игрока, взаимодействия с динамическими событиями (кризисами) и генерации контента с помощью ИИ.

Для достижения поставленных целей необходимо:

- разработать UML-диаграммы, отражающие архитектуру системы, включая диаграммы Use-Case, состояний, активностей и последовательностей, а также создать дизайн веб-приложения, соответствующий требованиям пользователей;
- реализовать базы данных для хранения информации о пользователях, стартапах, игровых сессиях и других данных, обеспечив их безопасность и масштабируемость;
- обеспечить интеграцию с ML-моделью для генерации разнообразного контента, такого как описания стартапов и кризисных ситуаций, что позволит создать гораздо больше интересных игровых ситуаций;
- провести тестирование системы, включая проверку функциональности, безопасности и производительности, а также выполнить аналитику для оценки соответствия проекта поставленным целям.

### **1.1 Требования к разрабатываемой системе**

#### **1.1.1 Функциональные требования**

К разрабатываемому приложению выдвинуты следующие требования:

- разделение пользователей на: неавторизованных и авторизованных;

- обеспечение возможности регистрации пользователей и авторизации пользователей;
- обеспечение возможности восстановления пароля;
- обеспечение возможности изменения данных аккаунты;
- обеспечение возможности сохранения и загрузки игрового процесса;
- обеспечение возможности взаимодействия с игровыми механиками: инвестирование в стартапы, участие в мероприятиях, решение кризисных ситуаций.

#### **1.1.1.1 Для неавторизованных пользователей**

- обеспечение возможности просматривать описание игры;
- обеспечение возможности регистрироваться в системе;
- обеспечение возможности авторизоваться в системе;
- обеспечение возможности восстановления пароля.

#### **1.1.1.2 Для авторизованных пользователей**

- обеспечение возможности посмотреть данные аккаунта в личном кабинете;
- обеспечение возможности сменить почту в личном кабинете;
- обеспечение возможности сменить логин в личном кабинете;
- обеспечение возможности сменить пароль в личном кабинете;
- обеспечение возможности начать новую игровую сессию;
- обеспечение возможности загрузить сохранённую игровую сессию;
- обеспечение возможности посмотреть список доступных для покупки стартапов;
- обеспечение возможности просмотреть условия сделки по покупке стартапа;
- обеспечение возможности предложить свои условия сделки по покупке стартапа;

- обеспечение возможности заключить сделку;
- обеспечение возможности просмотреть список купленных стартапов;
- обеспечение возможности просмотреть статистику изменений и состояние купленного стартапа;
- обеспечение возможности продать купленный стартап;
- обеспечение возможности просмотреть список доступных для посещения мероприятий;
- обеспечение возможности посетить мероприятие;
- обеспечение возможности просмотреть свои показатели экспертности в каждой из областей;
- обеспечение возможности просмотреть свои финансовые показатели;
- обеспечение возможности просмотреть свои показатели репутации;
- обеспечение возможности просмотреть оставшееся количество очков действий на текущий ход;
- обеспечение возможности просмотреть текущий месяц;
- обеспечение возможности перейти на следующий ход;
- обеспечение возможности просмотреть кризис;
- обеспечение возможности выбрать решение кризиса;
- обеспечение возможности завершить игру с сохранением;
- обеспечение возможности завершить игру без сохранения;

### **1.1.2 Нефункциональные требования**

К проекту выдвинуты следующие нефункциональные требования:

- защита от SQL-инъекций и XSS-атак;
- аутентификация пользователя через сессии;
- поддержка вертикальной и горизонтальной масштабируемости.

### **1.1.3 Технические требования**



### Список требований к архитектуре:

- сайт должен быть построен с использованием протоколов HTTPS, поддерживать REST API для взаимодействия клиентской и серверной частей;
- для хранения информации должны использоваться как реляционные базы данных (PostgreSQL), так и документоориентированные (MongoDB), что обеспечит гибкость в управлении структурированными и неструктурированными данными;
- клиентская часть сайта должна быть написана с использованием технологий frontend-разработки: HTML, CSS и TypeScript с фреймворком React и UI Kit Material UI;
- серверная часть сайта должна быть разработана с использованием языка программирования Java и фреймворка Spring;
- в системе предусмотрена интеграция с Python-библиотеками для реализации функционала искусственного интеллекта с целью генерации внутриигрового контента;
- веб-приложение должно быть разделено на backend и frontend части, взаимодействующие между собой через API, а также обеспечивать возможность масштабирования;
- разработка должна вестись с применением архитектурного паттерна MVC (Model-View-Controller), что позволит упростить поддержку и расширение системы;
- все данные пользователей должны защищаться с учетом современных требований безопасности.

### 1.2 Требования к интерфейсу

Интерфейс приложения должен быть выполнен в единой цветовой гамме и стилевом решении для всех экранов, обеспечивая целостность восприятия. Все элементы управления должны быть оформлены в едином стиле, с соблюдением одинаковых размеров и визуальных характеристик.

Интерфейс должен содержать только необходимую для пользователя информацию. Информация должна размещаться строго в тех разделах интерфейса, где она наиболее актуальна для пользователя, избегая избыточности и перегруженности.

### **1.3 Задачи, решаемые в процессе разработки**

В процессе создания веб-приложения TechTrackInvest были поставлены и последовательно решались следующие задачи:

- анализ предметной области;
- анализ аналогичных продуктов и образовательных игр;
- формализация требований и составление технического задания;
- разработка структуры базы данных и создание ER-диаграмм;
- разработка функциональных возможностей сайта;
- создание макета дизайна и его реализация;
- разработка клиентской части приложения;
- разработка серверной части приложения;
- реализация REST API для связи между frontend и backend;
- тестирование, отладка и обеспечение корректной работы функционала;
- подготовка технической и курсовой документации.

## **2 Анализ предметной области**

### **2.1 Терминология (гlossарий) предметной области**

Проект, система – разрабатываемое интерактивное веб-приложение TechTrackInvest, моделирующее процесс инвестирования в стартапы с использованием механик случайных событий.

Игрок (пользователь) – авторизованный пользователь системы, принимающий участие в симуляции инвестиционного процесса, взаимодействующий с игровыми механиками, принимающий стратегические решения.

Главный экран – основная страница игрового процесса, содержащая информацию об очках действий, состоянии ресурсов и навигацию по функциональным разделам игры (стартапы, мероприятия, кризисы).

Стартап – внутриигровой инвестиционный объект, доступный для покупки, анализа и продажи. Может приносить прибыль или убытки в зависимости от решений игрока и внешних событий.

Мероприятие – игровое событие, участие в котором улучшает характеристики игрока (например, экспертность, репутацию), и требует траты очков действий и финансов.

Кризис – непредвиденное событие, требующее от игрока принятия решений. Решения влияют на состояние стартапов и ресурсы игрока.

Очки действий – внутриигровой ресурс, ограничивающий количество действий игрока за один игровой ход (месяц).

Экспертиза – функционал, позволяющий получить дополнительную информацию о выбранном стартапе перед инвестированием.

Договор (сделка) – результат взаимодействия игрока со стартапом, оформляющий процесс инвестирования. Включает выбор условий и результат, определяемый механикой броска кубика.

Ход (месяц) – единица времени в игре. Каждый ход игрок может выполнять определенное количество действий, затем переходит к следующему периоду.

Игра – весь цикл взаимодействия игрока с системой: от регистрации и прохождения сессии до победы или поражения.

Backend (серверная часть) – программно-аппаратная часть, реализующая бизнес-логику приложения, взаимодействие с базой данных, реализацию правил игры и предоставление API.

Frontend (клиентская часть) – пользовательский интерфейс приложения, разработанный на React с использованием HTML, CSS и TypeScript, отвечающий за отображение данных и взаимодействие с игроком.

Сервер – компьютер или облачный сервис, обеспечивающий выполнение кода backend-части и обмен данными с клиентом.

REST API – архитектурный стиль для построения взаимодействия между front-end и back-end через HTTP-запросы.

GitHub – система хранения и совместной разработки исходного кода проекта.

Фреймворк – программная платформа, обеспечивающая структуру приложения.

Java – язык программирования, используемый для реализации серверной части приложения.

Spring – Java-фреймворк для создания REST API и реализации бизнес-логики.

Python – язык программирования, применяемый для работы с ИИ-алгоритмами в проекте.

PostgreSQL – реляционная СУБД, используемая для хранения информации о пользователях, сессиях, сделках и игровых событиях.

MongoDB – документоориентированная база данных, применяемая для хранения динамического игрового контента и ресурсов.

HTML, CSS, TypeScript – основные языки frontend-разработки, применяемые для построения клиентского интерфейса.

React – JavaScript-библиотека, используемая для построения компонентов пользовательского интерфейса.

UI Kit Material UI – библиотека компонентов интерфейса, применяемая для стилизации и стандартизации внешнего вида приложения.

Тестирование UI – процесс проверки корректности отображения и функционирования элементов интерфейса.

Юзабилити-тестирование – оценка удобства интерфейса и взаимодействия с ним с точки зрения игрока.

## 2.2 Обзор аналогов

### 2.2.1 Финансовый тренажёр (CashGo)

CashGo[1] – игра, направленная на обучение базовым принципам инвестирования. Пользователю предлагается вкладывать виртуальные средства в различные активы, однако функционал ограничен. Интерфейс сайта представлен в соответствии с рисунком 1.

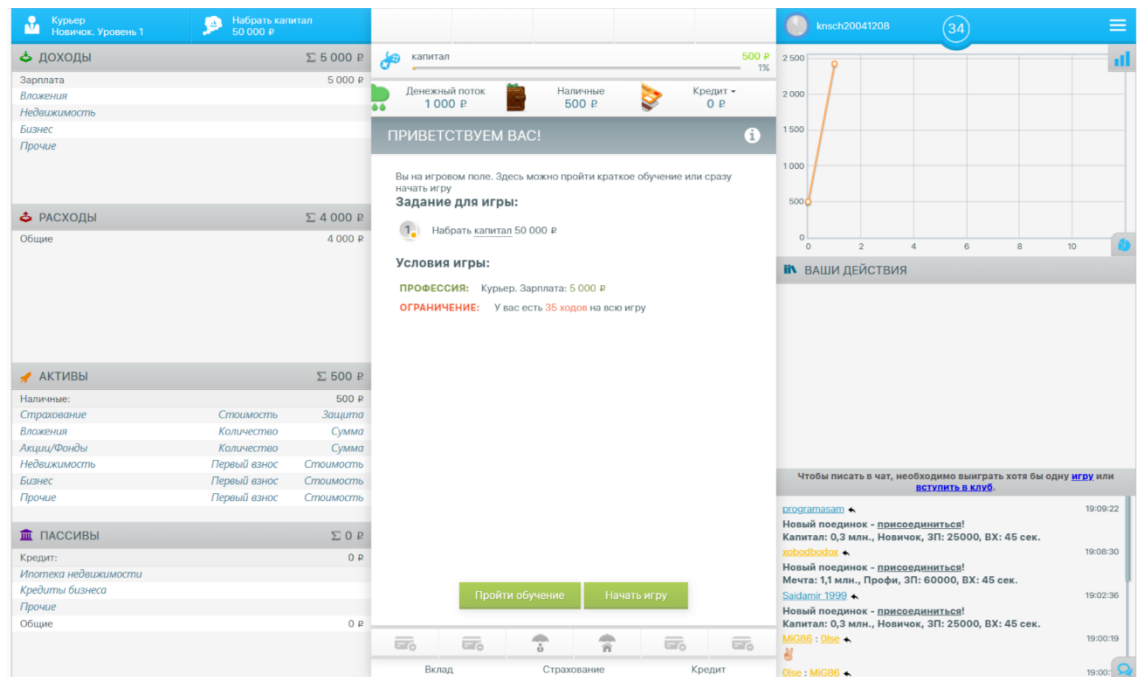


Рисунок 1 - Интерфейс сайта CashGo

Недостатками сайта CashGo являются:

- перегруженный интерфейс;
- однообразные варианты для инвестирования;
- активы предоставляются без подробной аналитики;
- отсутствует развитие личных характеристик игрока;
- отсутствует сохранение игровой сессии.

### 2.2.2 Симулятор инвестора (BuildYourStax)

BuildYourStax[2] – упрощённый симулятор, где игрок управляет абстрактным портфелем активов. Интерфейс сайта представлен в соответствии с рисунком 2.

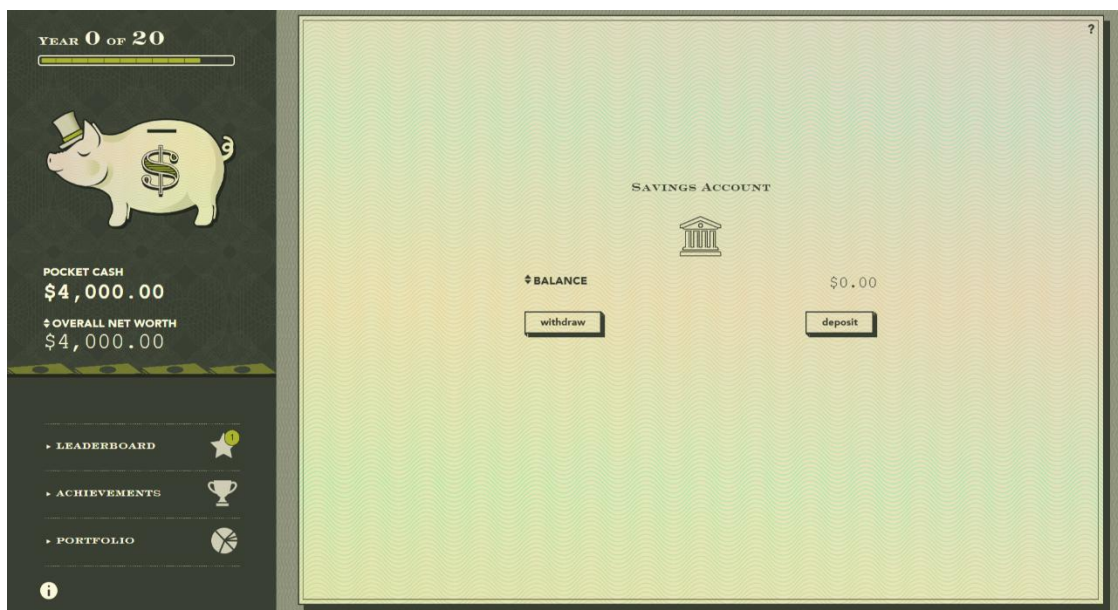


Рисунок 2 - Интерфейс сайта BuildYourStax

Из плюсов стоит отметить:

- простой и понятный интерфейс;
- реализованная система кризисов.

Недостатками сайта BuildYourStax являются:

- отсутствие информации об активах, что снижает образовательную ценность;
- отсутствие развития персональных навыков игрока;
- однообразный игровой процесс.

### 2.2.3 Симулятор инвестора (от medoborodyj)

Симулятор инвестора (от medoborodyj)[3] – это симулятор, который концентрируется на базовых механиках инвестирования без глубокой проработки контента. Интерфейс сайта представлен в соответствии с рисунком 3.

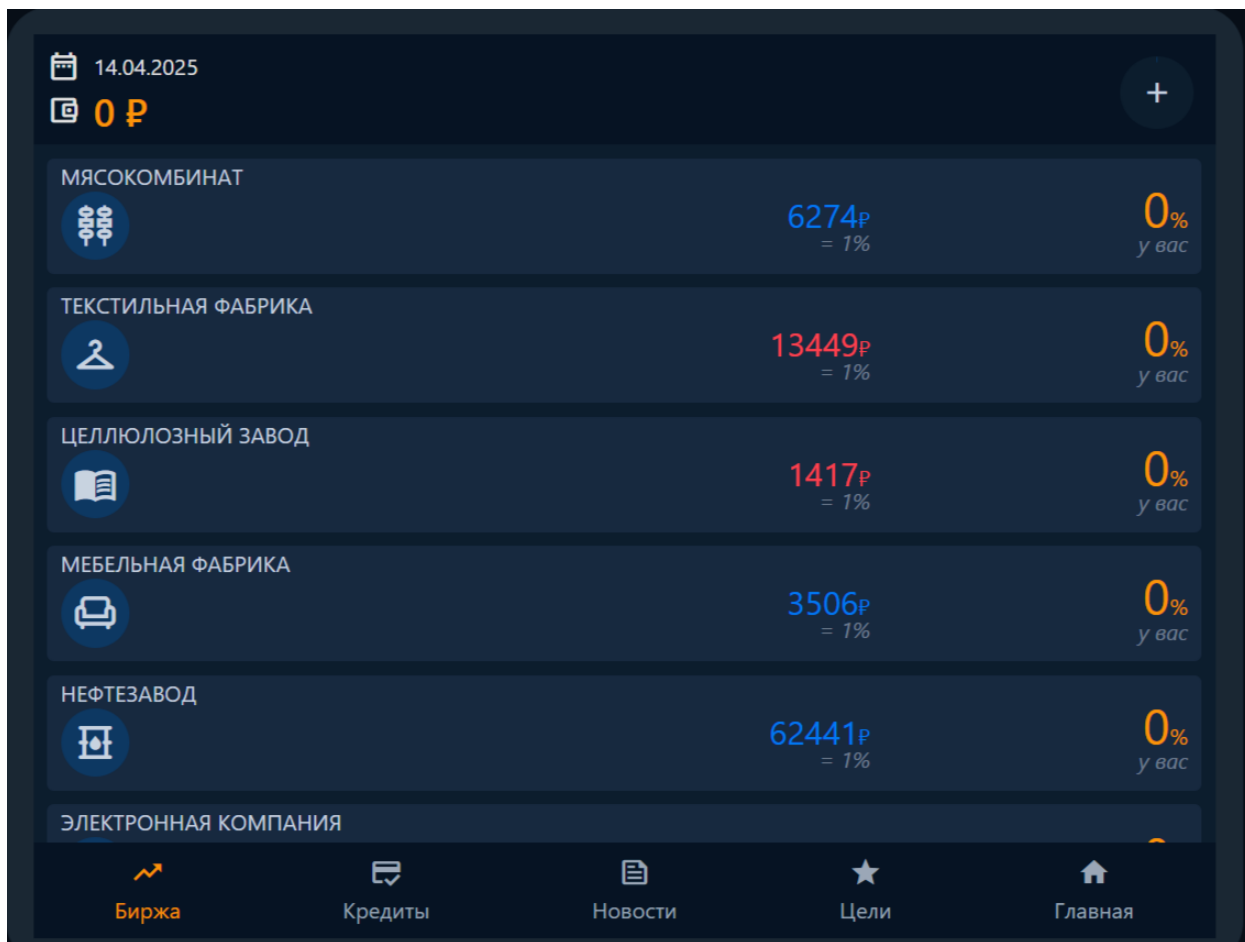


Рисунок 3 - Интерфейс симулятора инвестора (от medoboroduj)

Минусы симулятора инвестора (от medoboroduj):

- однообразные сценарии инвестирования;
- отсутствие системы развития характеристик;
- отсутствие сохранения прогресса.



## 2.3 Моделирование системы

### 2.3.1 Диаграмма прецедентов

Неавторизованный пользователь (Рисунок 4) может:

- просмотреть приветственное окно;
- создать новый аккаунт, введя логин, email и пароль;
- войти в существующий аккаунт, используя почту и пароль.

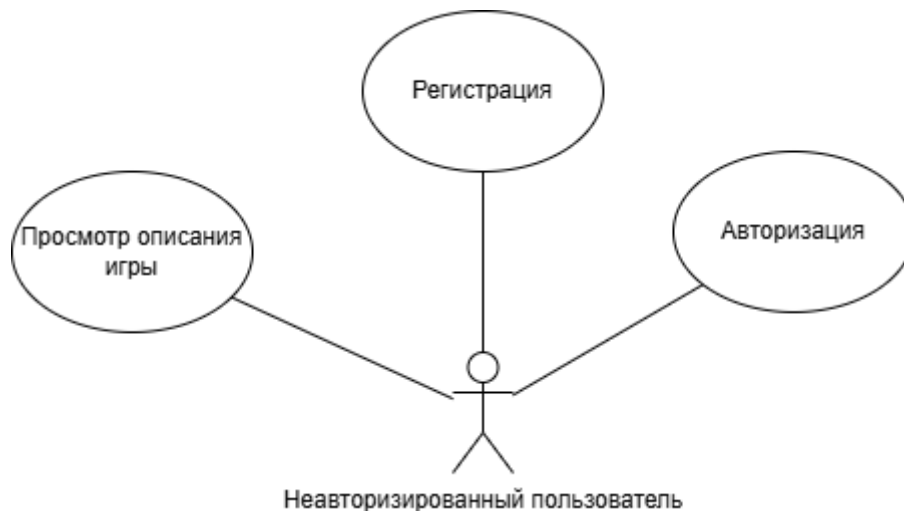


Рисунок 4 - Диаграмма прецедентов для неавторизованного пользователя

Авторизованный пользователь (Рисунок 5) может:

- начать новую игровую сессию: создать новую игру с начальными параметрами;
- загрузить сохранённую сессию: продолжить игру с ранее сохранённого момента;
- просматривать список стартапов: анализировать доступные для инвестирования стартапы, их отрасли и описания;
- заключить сделку: инвестировать в выбранный стартап, предлагая свои условия;
- заказать экспертизу: получить дополнительную информацию о стартапе перед инвестированием;
- продать стартап: избавиться от купленного стартапа для возврата части средств;

- посещать мероприятия: улучшать свои характеристики (экспертность, репутацию) через участие в мероприятиях;
- решать кризисы: выбирать стратегии для преодоления случайных кризисных событий;
- просматривать статистику: анализировать динамику изменений своих финансов, репутации и экспертизы;
- завершить ход: перейти к следующему игровому месяцу;
- завершить игру: сохранить или отменить текущую сессию.

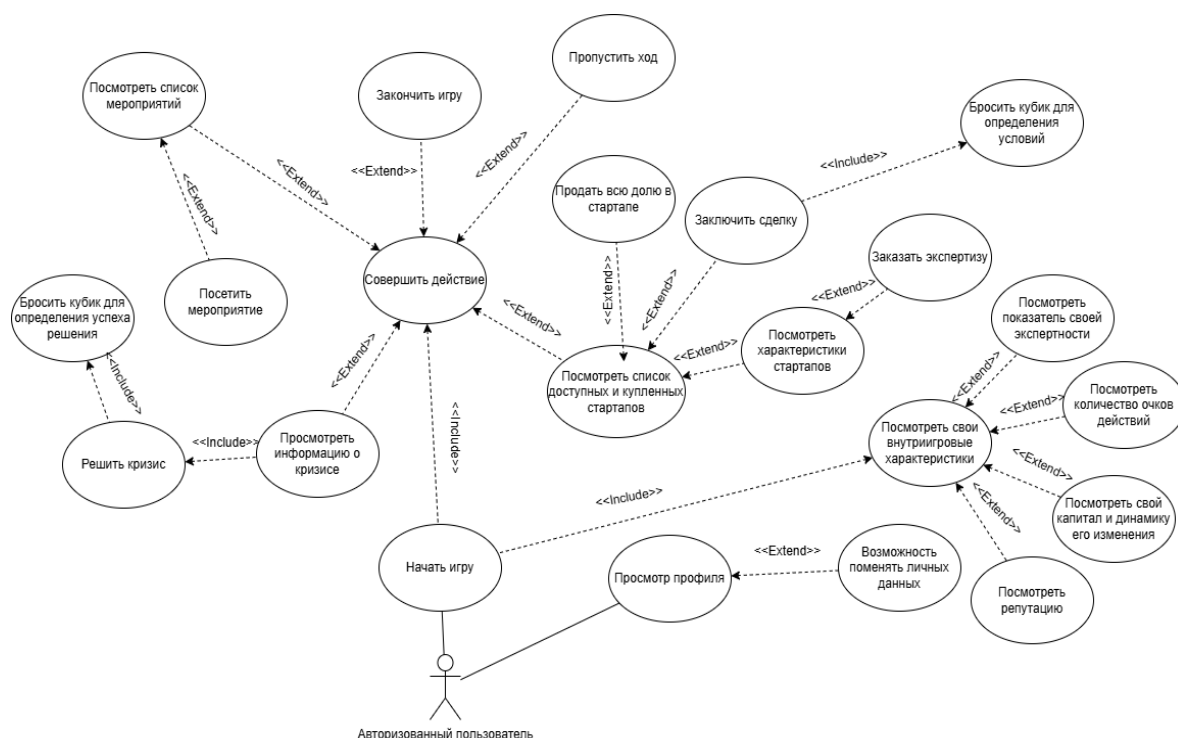


Рисунок 5 - Диаграмма прецедентов для авторизованного пользователя

### 2.3.2 Диаграмма последовательности

Диаграмма последовательностей (Рисунок 6) иллюстрирует взаимодействие между участниками системы в рамках ключевых сценариев. Основные объекты:

- Пользователь – инициирует действия через интерфейс (например, регистрацию или инвестирование в стартап).

- Клиент (Frontend) – обрабатывает запросы пользователя и отправляет их на сервер.
- Сервер (Backend) – выполняет бизнес-логику: аутентификацию, генерацию игровых событий, обработку сделок.
- База данных – хранит данные пользователей, стартапов, сессий и кризисов.

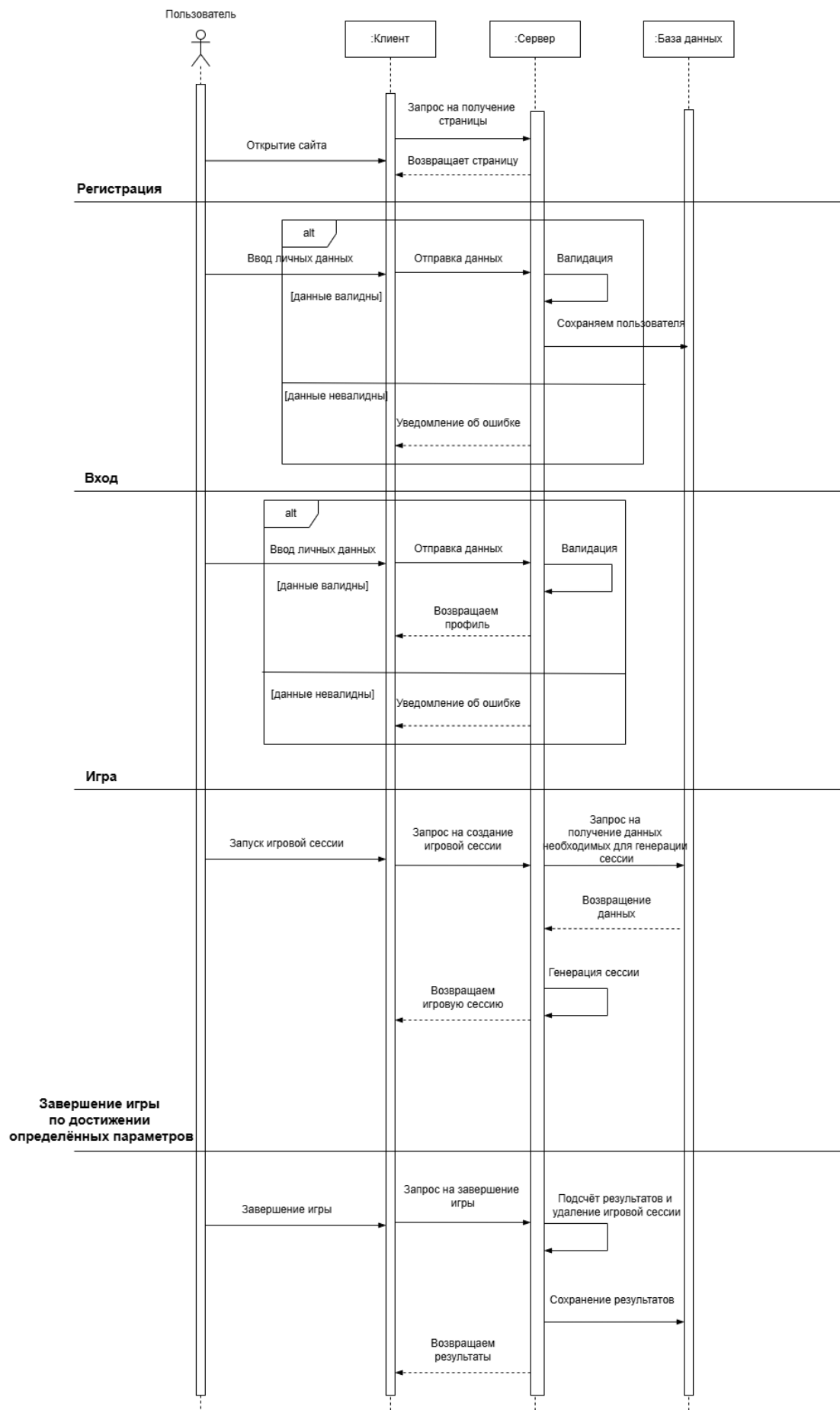


Рисунок 6 - Диаграмма последовательностей

### 2.3.3 Диаграмма состояний

Диаграмма состояний (Рисунок 7) отражает внутренние состояния объекта в течение его жизненного цикла от момента создания до разрушения.

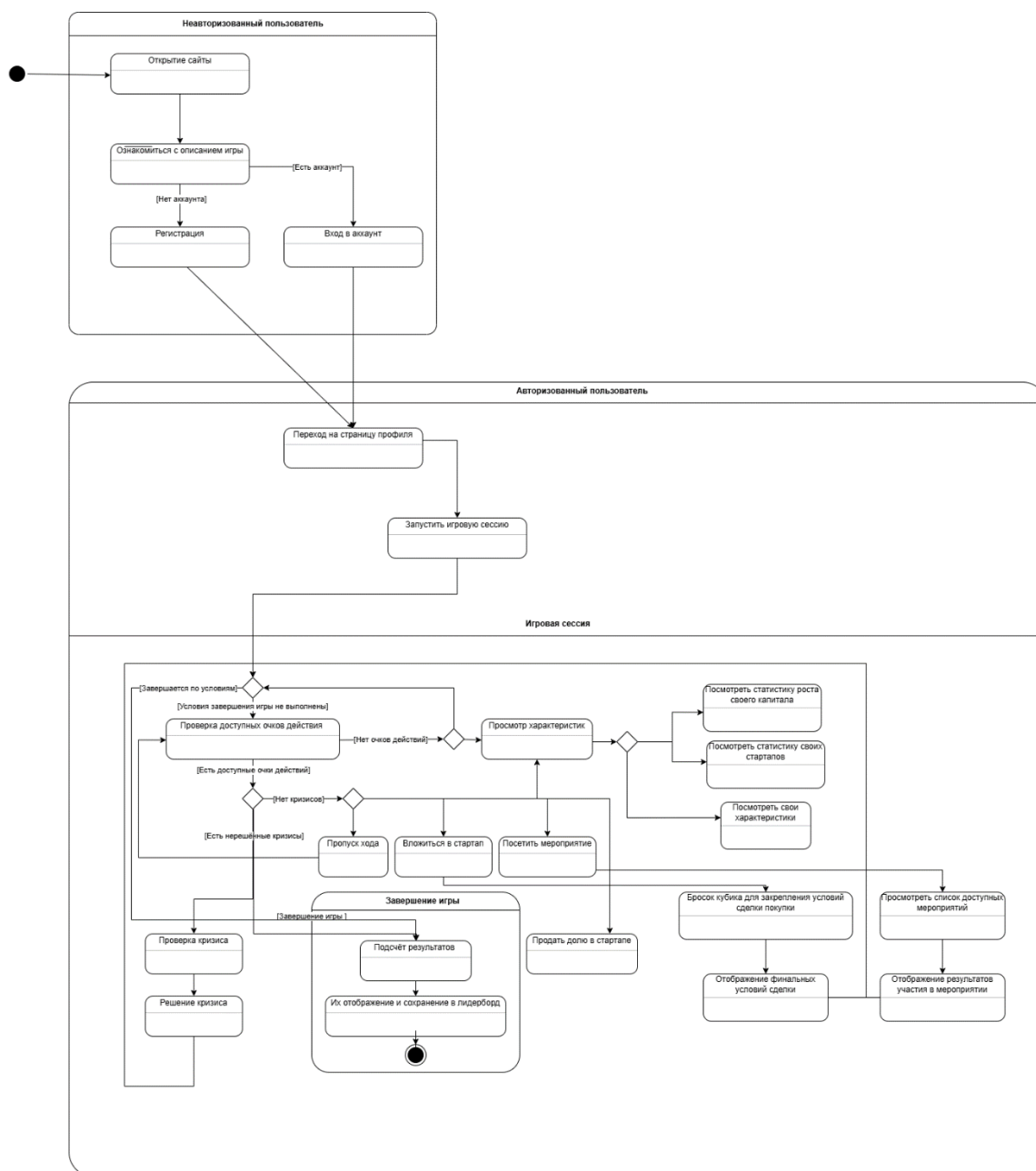


Рисунок 7 - Диаграмма состояний

### 2.3.4 Диаграмма активностей

Диаграмма активностей на Рисунке 8 описывает последовательность действий пользователя в системе "TechTrackInvest", начиная с

неавторизованного состояния. Пользователь сначала открывает сайт и знакомится с описанием игры. Затем система проверяет наличие аккаунта. Если аккаунта нет, пользователь проходит регистрацию и входит в систему. Если аккаунт есть, он авторизуется и переходит к игровому процессу.

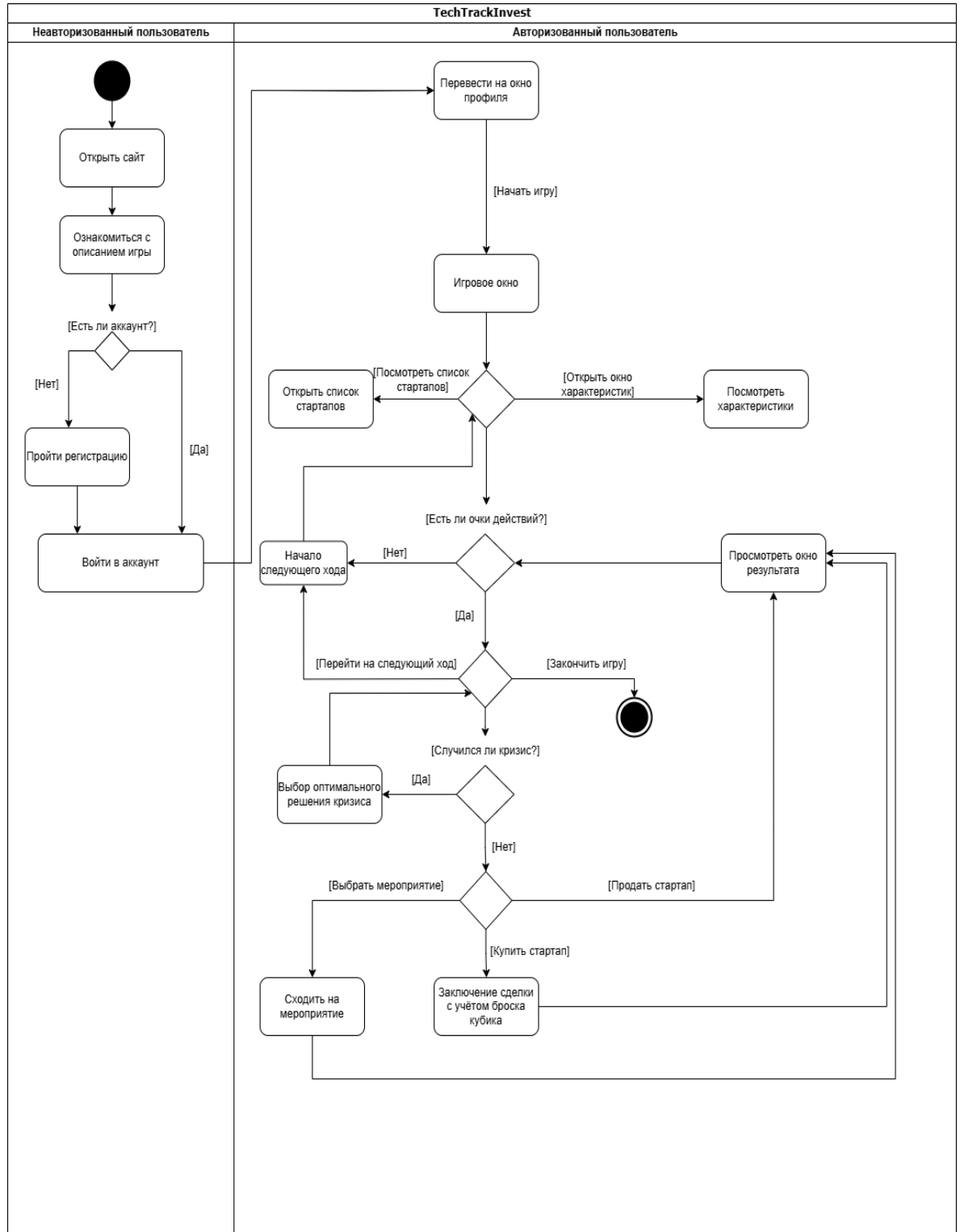


Рисунок 8 - Диаграмма активностей

### 2.3.5 Диаграмма классов

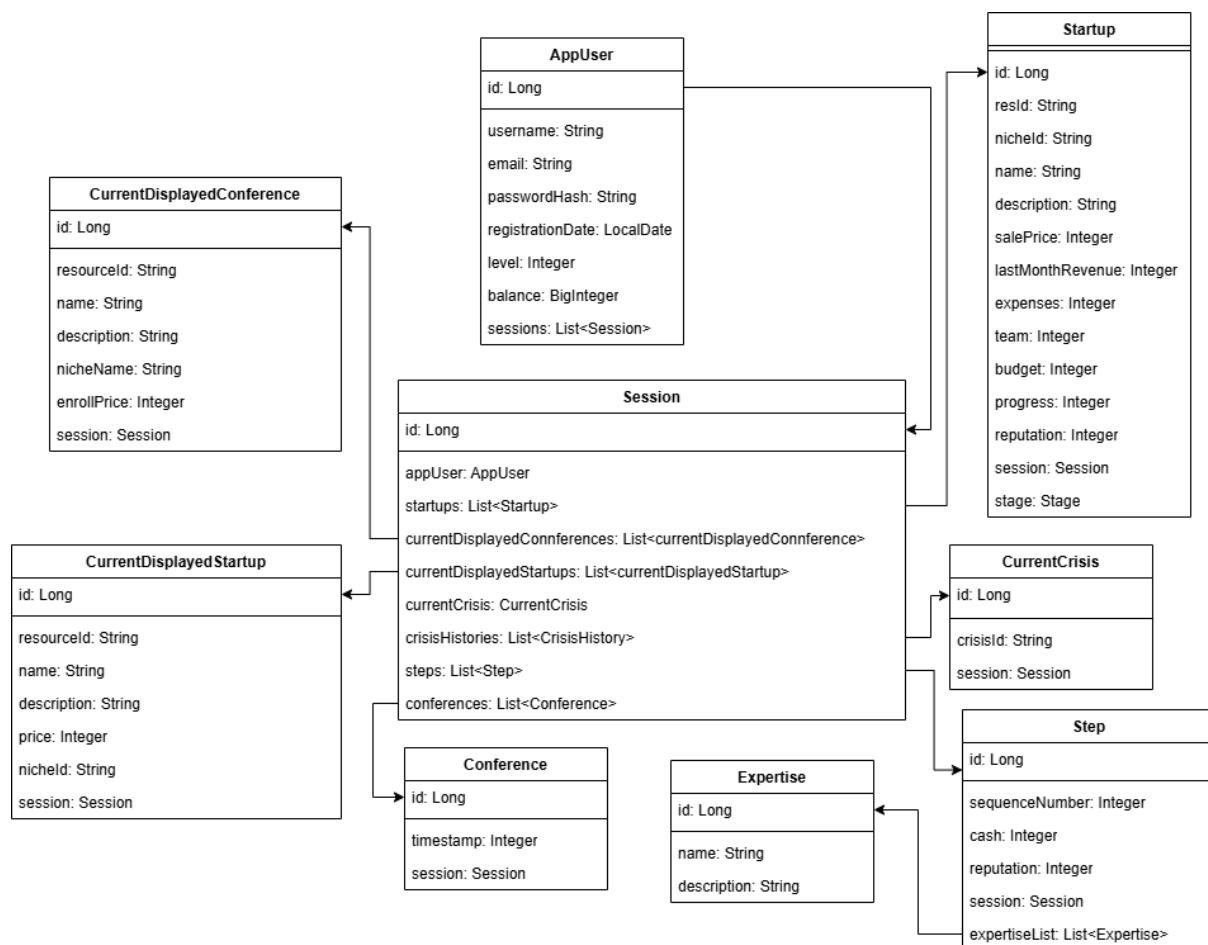


Рисунок 9 - Диаграмма классов

### 2.3.6 ER-диаграмма

ER-диаграмма — это графическое представление модели данных, которая используется для описания концептуальной структуры базы данных.

Проект использует две принципиально разные модели хранения данных, представленные соответствующими диаграммами. Реляционная схема на PostgreSQL (Рисунок 10) организует данные в виде строго структурированных таблиц с четкими связями между ними. Она включает основные сущности: пользователей с их учетными данными и балансом, игровые сессии, хранящие прогресс каждого игрока, стартапы с их финансовыми и репутационными

характеристиками, а также отдельные таблицы для мероприятий и кризисных событий.

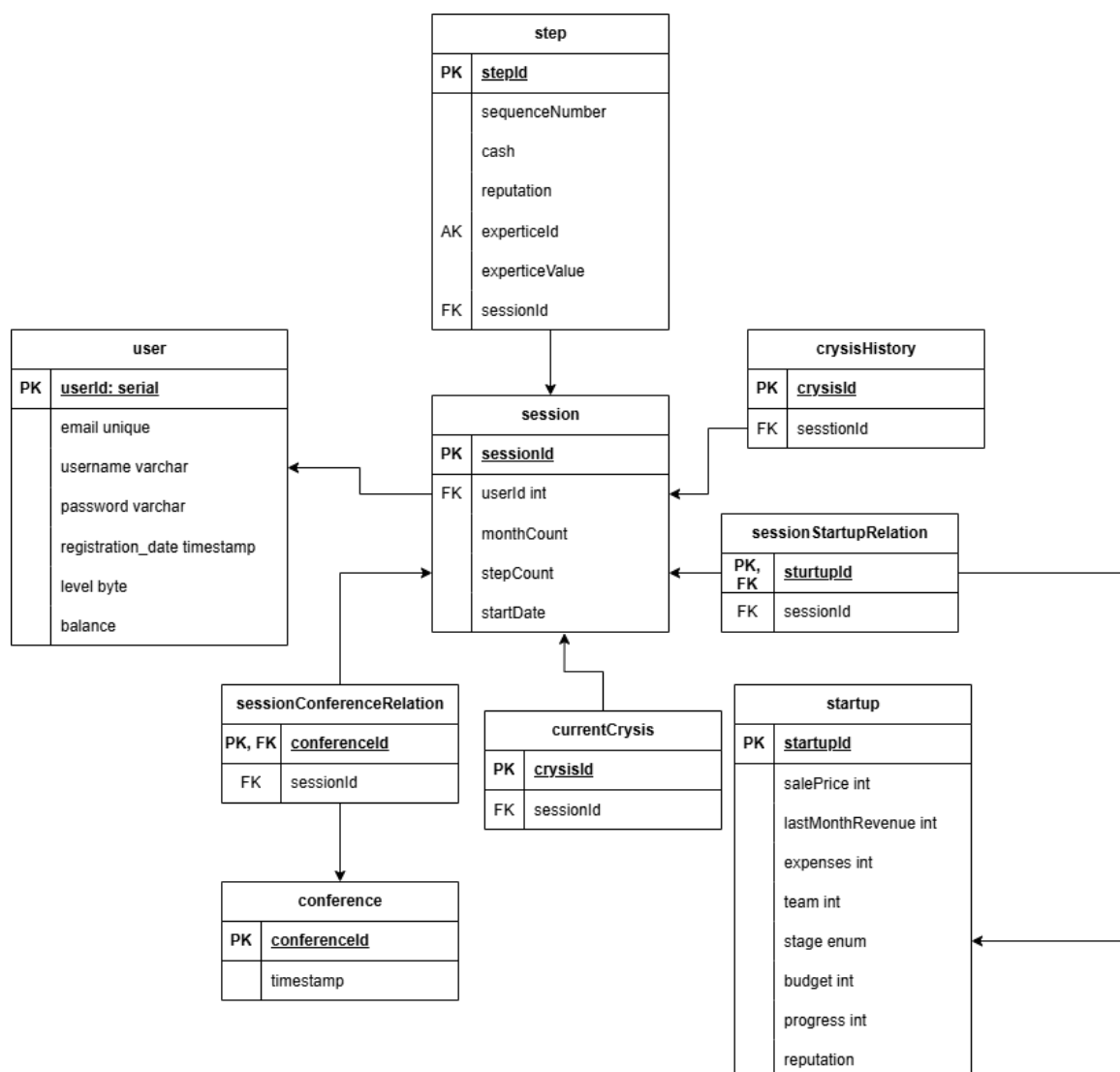


Рисунок 10 - Диаграмма PostgreSQL

Параллельно используется документоориентированный подход в MongoDB (Рисунок 11), который благодаря гибкой JSON-структуре идеально подходит для хранения изменчивых игровых данных. В этой схеме стартапы описываются с учетом их уникальных особенностей и динамических характеристик, кризисы хранятся вместе с возможными решениями и их эффектами, а контракты содержат сложные условия инвестирования. Такой формат позволяет легко модифицировать игровой контент без изменения структуры базы данных.



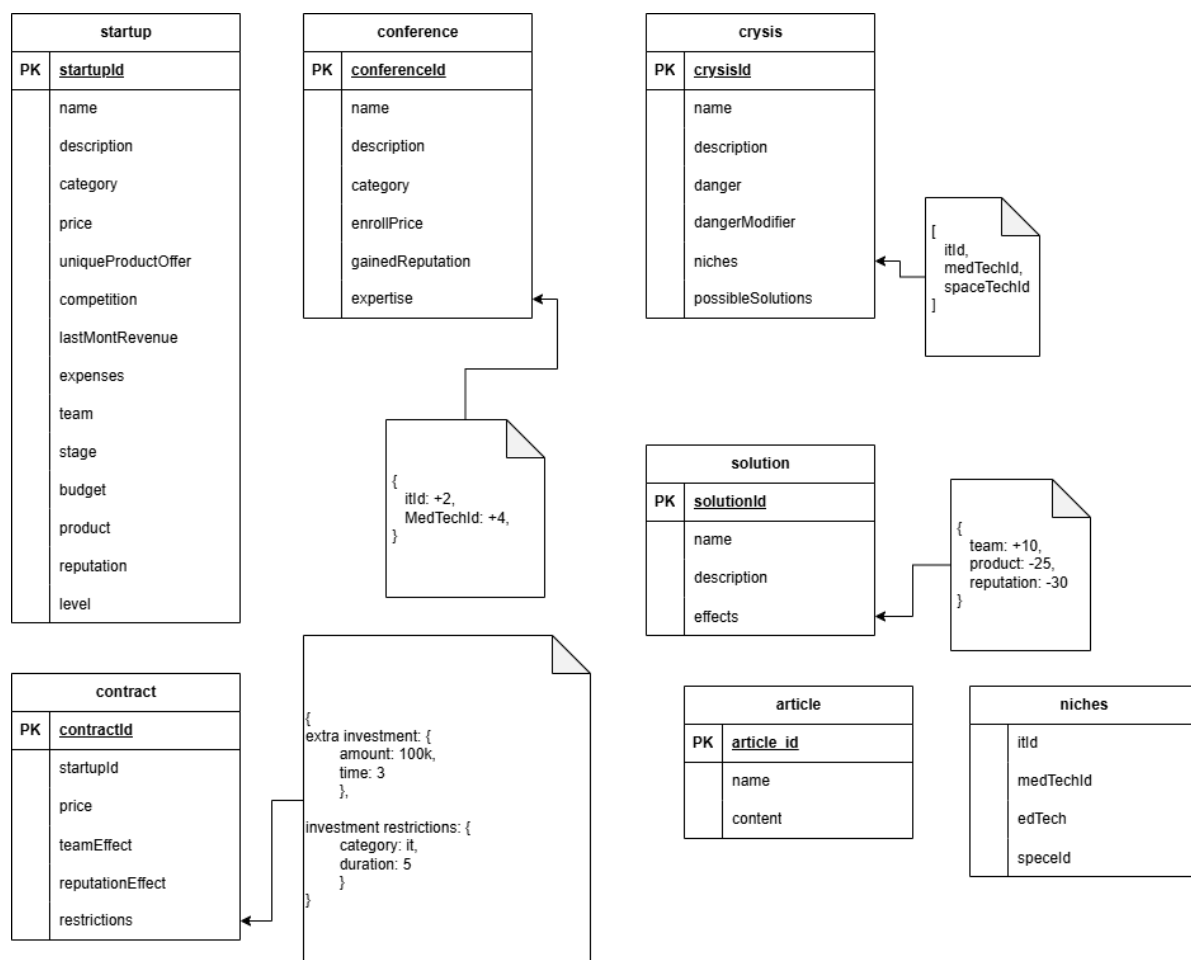


Рисунок 11 - Диаграмма MongoDB

### 2.3.7 Диаграмма развёртывания

Диаграмма развёртывания (Рисунок 12) предназначена для представления общей конфигурации или топологии распределенной программной системы.

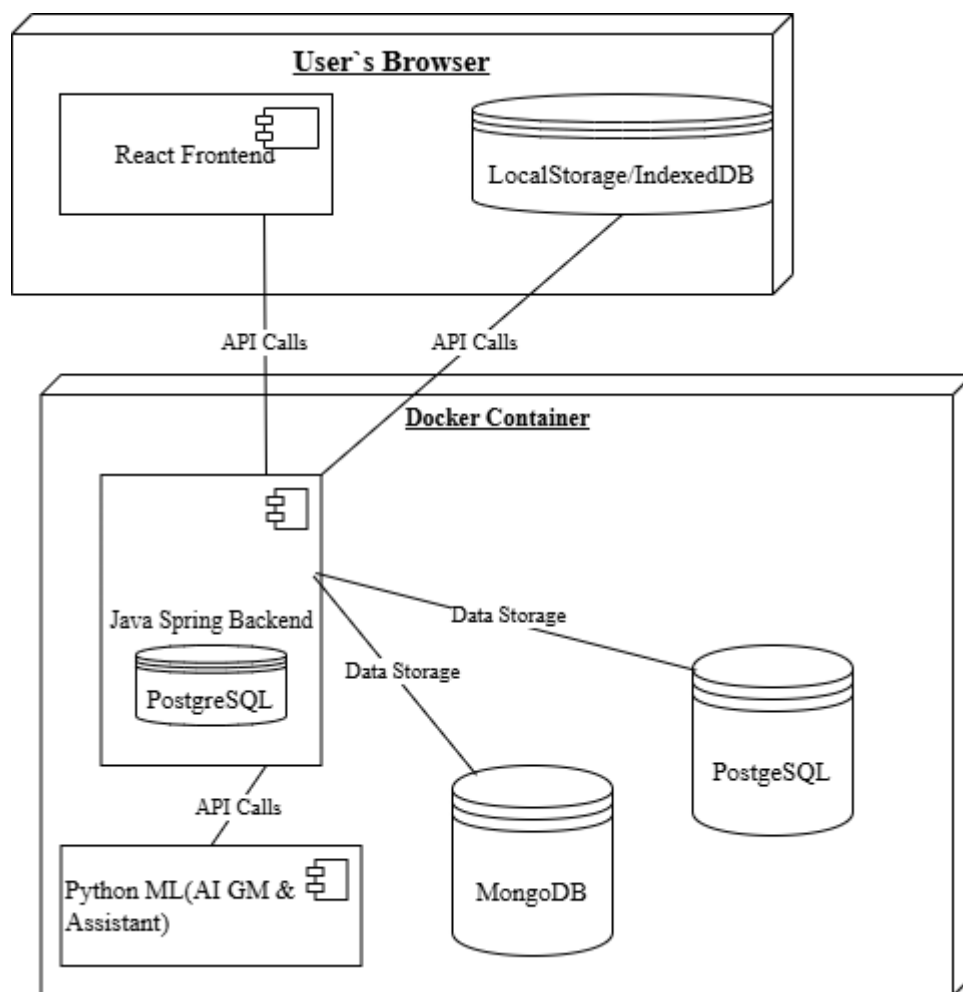


Рисунок 12 - Диаграмма развёртывания

### 2.3.8 Диаграмма в стиле методологии IDEF0

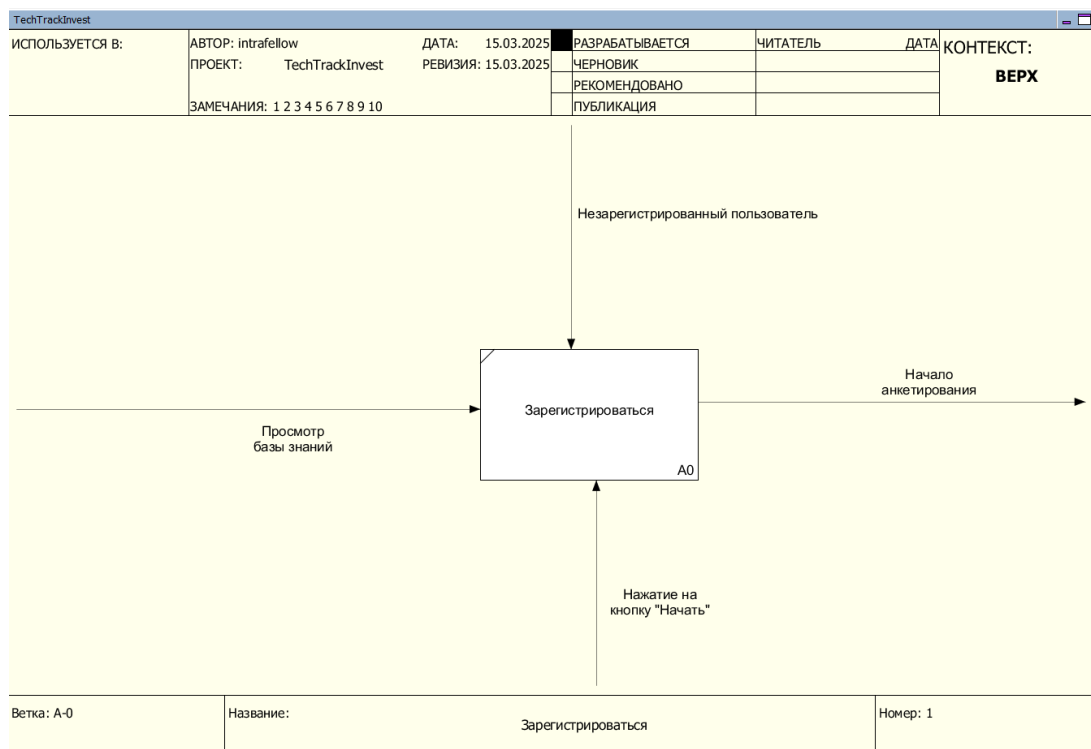


Рисунок 13 - Контекстная диаграмма со стороны неавторизованного пользователя

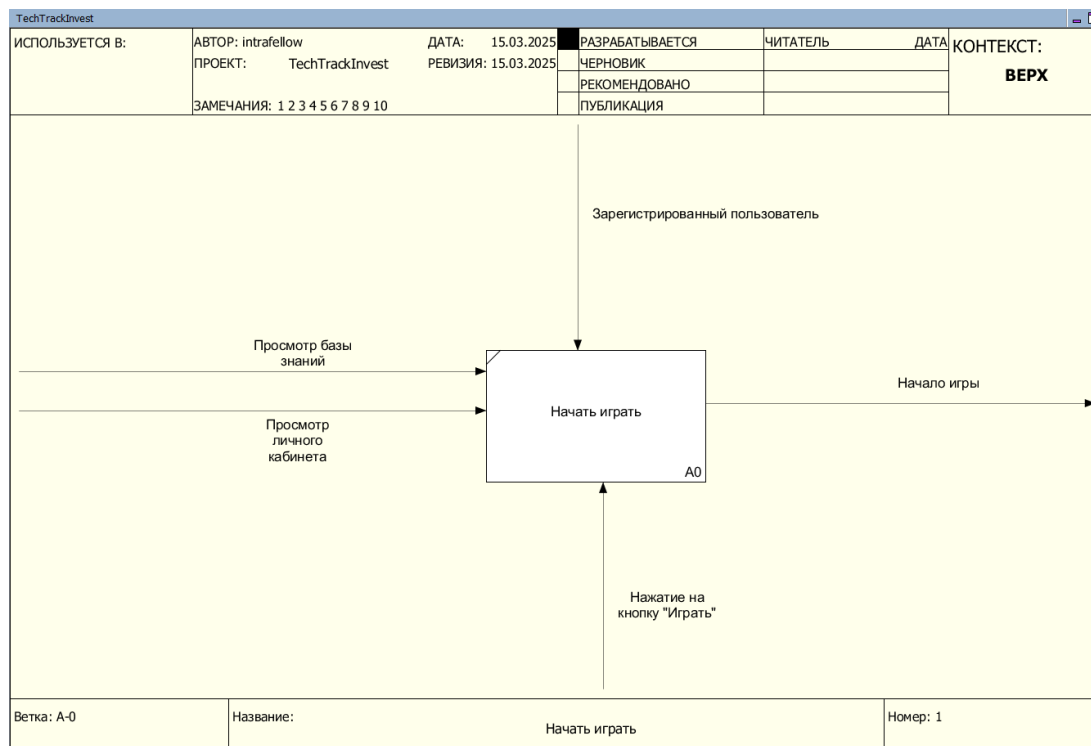


Рисунок 14 - Контекстная диаграмма со стороны авторизованного пользователя

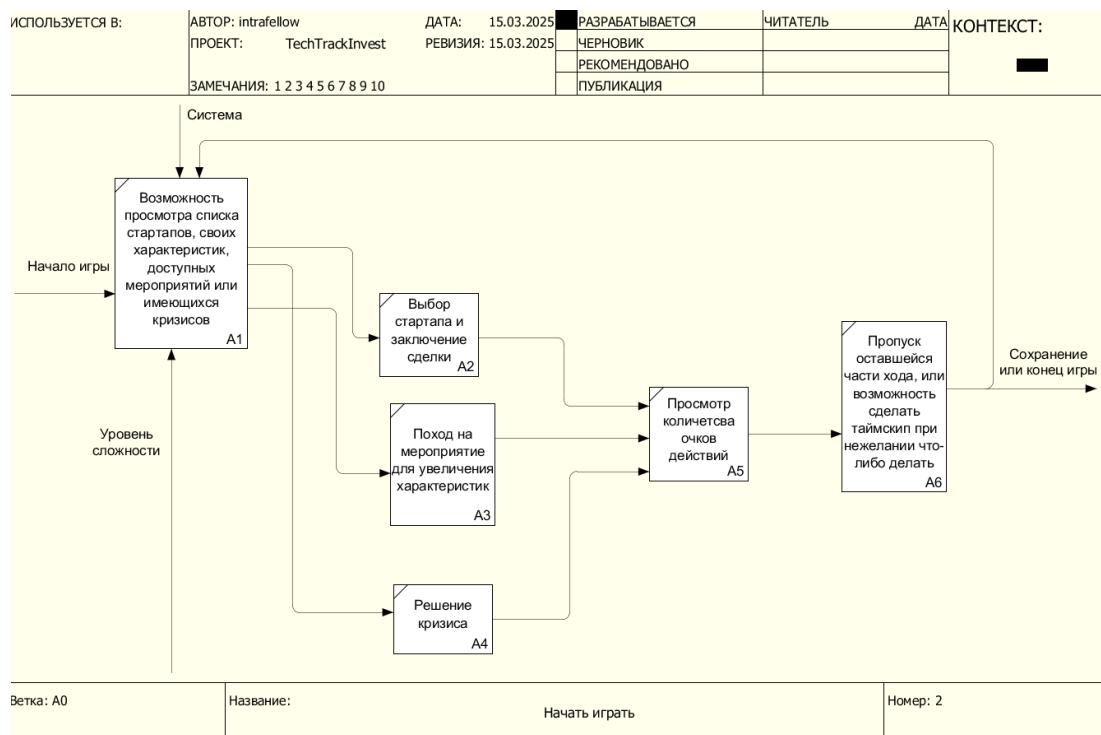


Рисунок 15 - Дочерняя диаграмма со стороны авторизованного пользователя

### 2.3.9 Диаграмма объектов

Рисунок 16 - Диаграмма объектов

## **2.4 Аналитика веб-приложения**

## 2.5 Границы проекта

Основными границами проекта являются:

- обеспечение базовых функций: регистрация, авторизация, сохранение игрового прогресса;
- включение динамических событий (кризисы, мероприятия) и системы развития характеристик игрока (экспертность, репутация, финансы);
- генерация уникальных стартапов для каждой игровой сессии;
- ограничение взаимодействия с системой через веб-интерфейс без поддержки мобильных приложений или оффлайн-режима;
- разработка клиентской части на React и серверной части на Java (Spring Framework);
- поддержка только русскоязычной версии;

На фоне основных сценариев и функциональных требований, можно сформулировать то, что точно выходит за основные границы проекта:

- мультиплеер или кооперативный режим;
- интеграция с реальными финансовыми системами или криптовалютами;
- поддержка других языков, кроме русского.

## **3 Реализация**

### **3.1 Средства реализации**

Веб-приложение "TechTrackInvest" построено по клиент-серверной архитектуре с четким разделением на backend и frontend части, взаимодействующие через REST API. Такой подход соответствует ключевым требованиям проекта:

- обеспечение авторизации и аутентификации пользователей;
- реализация игровой механики с сохранением прогресса;
- генерация динамического контента (стартапы, кризисы, мероприятия);
- Развитие характеристик игрока (экспертность, репутация, финансы).

Для backend-разработки выбран следующий технологический стек:

- язык программирования Java[4];
- фреймворк Spring[5];
- СУБД PostgreSQL[6];
- документоориентированная NoSQL база MongoDB;
- язык программирования Python.

Для frontend-разработки применены следующие технологии:

- язык программирования TypeScript;
- язык стилей CSS;
- фреймворк React[7];
- язык разметки HTML[8].

### **3.2 Реализация серверной части веб-приложения**

Серверная часть интерактивной игры TechTrackInvest реализована с использованием языка программирования Java и фреймворка Spring Boot, построена по архитектурному шаблону MVC и организована в виде модульной структуры. Веб-приложение функционирует как REST-сервис,

обрабатывающий запросы от клиентской части и обеспечивающий выполнение бизнес-логики игры.

Корневая точка входа в систему — файл `TechTrackInvestApplication.java`, аннотированный `@SpringBootApplication`, объединяющий конфигурацию, автоматическое сканирование компонентов и автоконфигурацию. Проект логически разделен на модули:

- `entities` — описывает JPA-сущности, соответствующие структуре реляционной базы данных PostgreSQL;
- `repositories` — реализует доступ к данным через интерфейсы `JpaRepository`;
- `service` — содержит бизнес-логику;
- `rest` — контроллеры, обрабатывающие HTTP-запросы и возвращающие JSON-ответы;
- `security` — отвечает за JWT-аутентификацию и авторизацию, конфигурирует фильтры безопасности;
- `dto` — классы передачи данных между слоями;
- `mapper` — маппинг DTO и сущностей с использованием `MapStruct`;
- `configs` — конфигурация Swagger, CORS, безопасности, подключения к базам данных;
- `exceptions` — централизованная обработка ошибок;
- `utils, filters, annotation` — вспомогательные классы, фильтры запросов, кастомные аннотации.

Аутентификация и авторизация реализованы с помощью JWT. Пользователь получает токен после входа, который используется для доступа к защищённым ресурсам. Все чувствительные маршруты защищены фильтрами, реализованными в модуле `security`. Конфиденциальные данные хранятся как переменные окружения.

Проект использует две базы данных:



- PostgreSQL – основная СУБД для хранения пользователей, игровых сессий, характеристик и сделок;
- MongoDB – документоориентированная база данных для хранения сгенерированного игрового контента.

Конфигурация подключения реализована в `application.properties`. Поддерживается стратегия `spring.jpa.hibernate.ddl-auto=update` для автоматического обновления схемы базы данных на этапе разработки.

### 3.3 Реализация клиентской части веб-приложения

Клиентская часть интерактивной инвестиционной игры TechTrackInvest реализована с использованием TypeScript и React.js, что обеспечивает компонентный подход, строгую типизацию и высокую надёжность пользовательского интерфейса. Архитектура frontend-приложения построена на современных принципах модульности, переиспользуемости и масштабируемости кода.

Проект имеет стандартную структуру React-приложения:

- `public/` — содержит статический файл `index.html`, в который внедряется React-приложение;
- `src/` — основной каталог исходного кода;
- `package.json` — файл зависимостей;
- `Dockerfile` и `nginx.conf` — используются для сборки и развертывания frontend-приложения в Docker-контейнере;
- `yarn.lock` — зафиксированные версии всех зависимостей.

Основные папки в `src`:

- `pages/` — содержит отдельные страницы приложения, где каждая страница представляет собой компонент верхнего уровня и может использовать общие компоненты;
- `components/` — в этой папке находятся переиспользуемые UI-компоненты: кнопки, карточки стартапов, формы, навигационные элементы и модальные окна;

- `api/` — содержит функции для взаимодействия с REST API backend-сервиса. Включает типы, обработчики ошибок, авторизацию по токену и конфигурацию базового клиента;
- `context/` — реализует глобальное состояние приложения через React Context API. Содержит провайдеры и хуки для управления пользователем, текущей сессией, статистикой и навигацией;
- `theme/` — настройки темы и UI-стилей: цветовая палитра, шрифты, отступы, стили компонентов. Используется Material UI для стилизации;
- `icons/` — содержит SVG-иконки, используемые в UI, а также обёртки React-компонентов для иконок.

Приложение запускается с `index.tsx`, который монтирует корневой компонент `App`. Внутри `App.tsx` происходит инициализация маршрутов, подключение провайдеров состояния, а также отрисовка страниц на основе текущего пути и состояния пользователя.

При первом входе пользователь попадает на экран приветствия или авторизации. После успешной аутентификации в глобальное состояние заносится информация о пользователе и токен, а маршрутизация автоматически переключает пользователя на основной игровой экран.

Каждое игровое действие инициирует вызов соответствующего API. После получения ответа состояние обновляется и изменения отображаются на экране без перезагрузки страницы.

Для стилизации интерфейса применяется кастомизированная тема Material UI, поддерживающая адаптивность. Компоненты оформлены в виде модулей, что позволяет масштабировать систему без дублирования логики.

Для удобства взаимодействия с REST API используется `axios`, где также реализованы:

- автоматическая установка JWT-токена в заголовок `Authorization`;
- глобальная обработка ошибок;
- управление сессией и редиректы при потере авторизации.

## 3.4 Навигация по сайту

### 3.4.1 Для неавторизованного пользователя

Первое что видит пользователь при открытии сайта – приветственное окно (Рисунок 17).

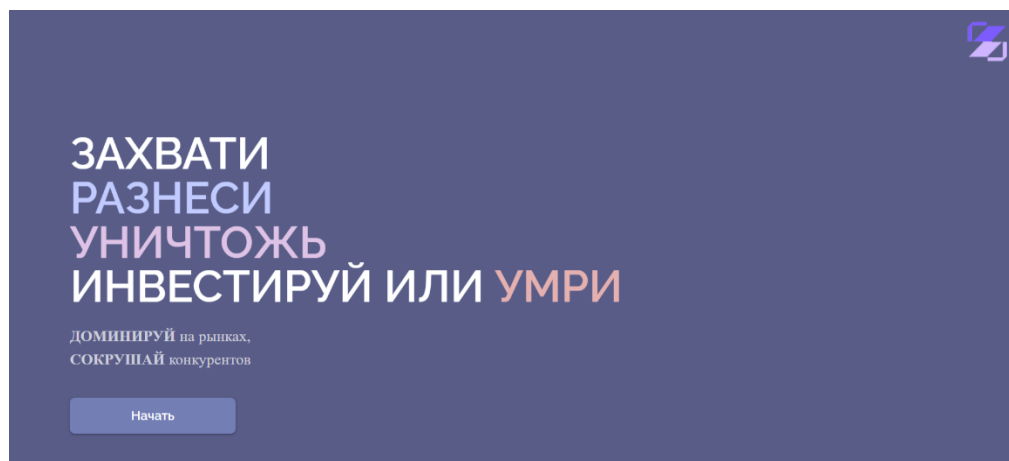


Рисунок 17 - Приветственное окно

Нажав на кнопку «Начать» пользователь попадает на окно входа в аккаунт (Рисунок 18).

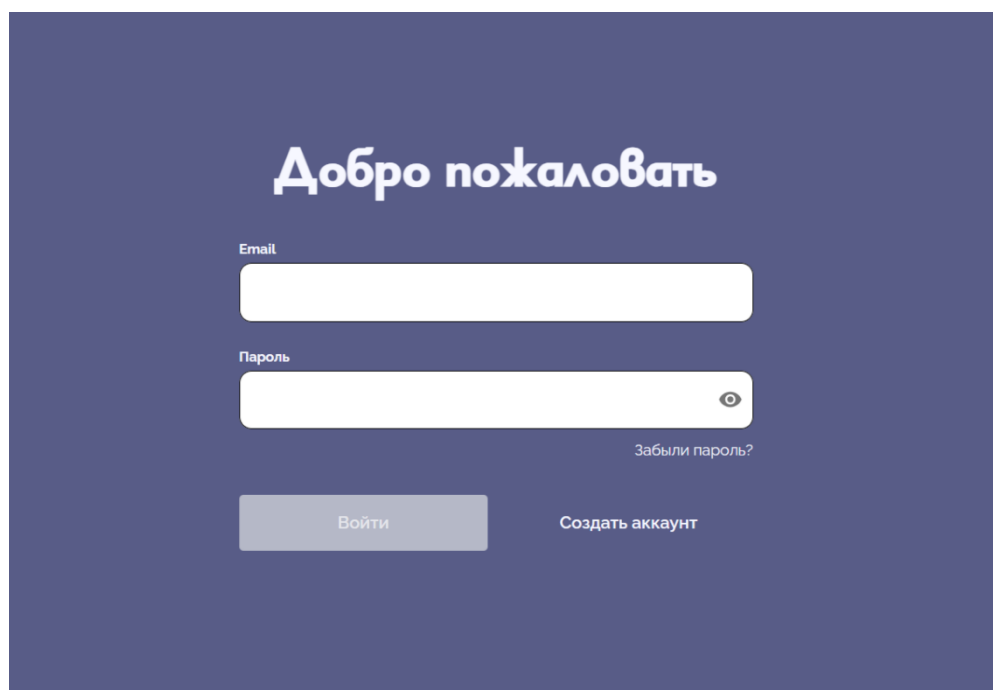


Рисунок 18 ->Login

Выбрав регистрацию, пользователю требуется ввести почту для подтверждения (Рисунок 19).

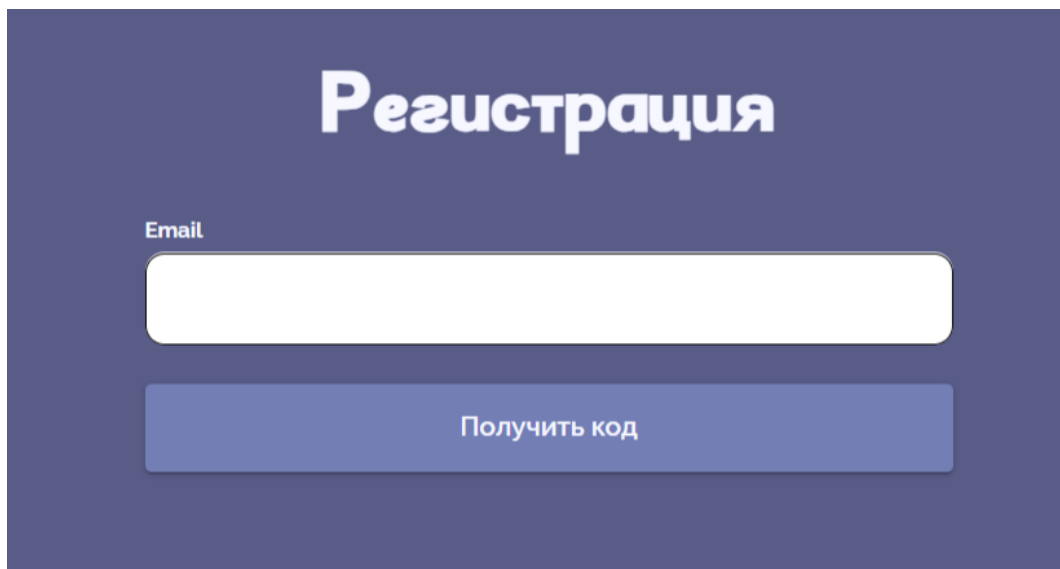
The image shows a registration confirmation screen with a dark blue background. At the top, the word "Регистрация" (Registration) is written in large, white, bold Cyrillic letters. Below it, the word "Email" is written in small, white, sans-serif font. Underneath "Email" is a white rectangular input field with rounded corners. Below the input field is a blue rectangular button with rounded corners, containing the text "Получить код" (Get code) in white, sans-serif font.

Рисунок 19 - Подтверждение почты

Подтвердив почту, введя токен, который присылается по его данным, в следующее окно, пользователю предоставляется возможность регистрации (Рисунок 20).

The image shows a registration form on a dark blue background. At the top, the word 'Регистрация' is written in large white letters. Below it are three white input fields with rounded corners. The first field is labeled 'Логин'. The second field is labeled 'Пароль' and has a small eye icon on its right side. The third field is labeled 'Подтвердите пароль' and also has an eye icon on its right side. At the bottom of the form is a wide, light gray button with the text 'Создать аккаунт' in dark gray.

Рисунок 20 - Регистрация

### 3.4.2 Для авторизованного

После создания нового аккаунта или входа по данным уже существующего, пользователь автоматически попадает в свой профиль (Рисунок 21).

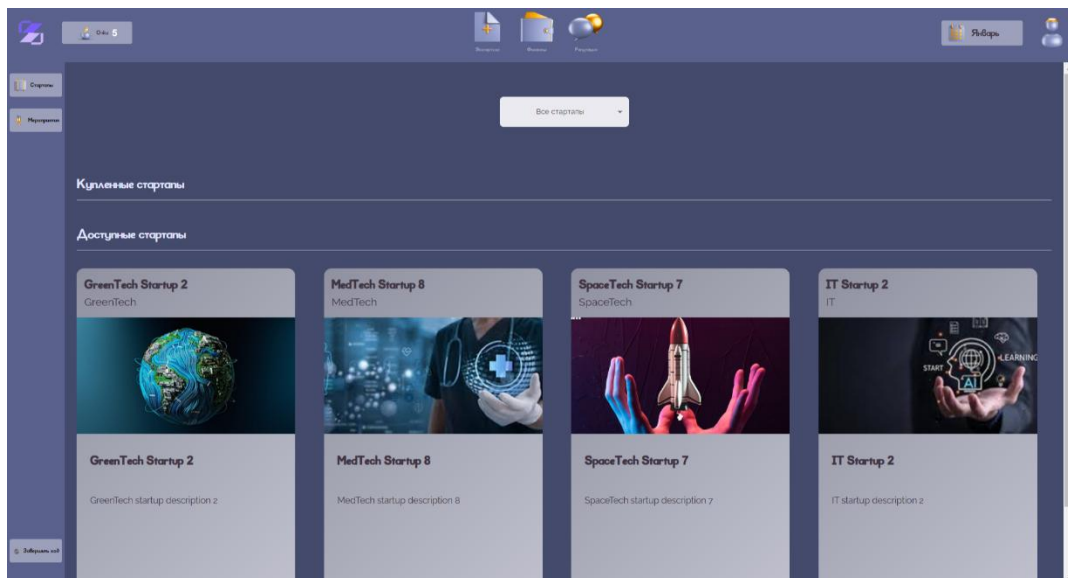
The image shows a user profile page on a dark blue background. At the top right is a small purple logo. In the center is a 3D avatar of a person with a blue body and a gold helmet, enclosed in a thin white circle. Below the avatar, the email address 'techtrackinvest@gmail.com' is displayed in white, with a small link 'Сменить email' underneath. Below the email is the login 'TechTrackInvest' in white, with a small link 'Сменить логин' underneath. At the bottom is a wide, light blue button with the text 'Начать играть' in white. Below this button is a small link 'Сменить пароль' in white.

## Рисунок 21 - Профиль

Из профиля ему предоставляется возможность изменить любые личные данные и начать игру или же загрузить уже существующую игровую сессию.

При переходе в саму игру его встречает основной экран (Рисунок 22), который включает:

- список доступных и купленных стартапов, с кнопками для проведения экспертизы и заключения сделки;
- выпадающий список для фильтрации по отраслям;
- левое меню с возможностью переключиться на окно мероприятий, обратно на окно стартапов или завершить ход;
- верхний раздел показывает оставшееся количество очков действий, текущий месяц и характеристики игрока: экспертность, деньги, репутация.



## Рисунок 22 - Игровое окно

Перед заключением сделки на покупку понравившегося стартапа пользователь может заказать экспертизу и заранее узнать его характеристики (Рисунок 23).



Рисунок 23 - Экспертиза стартапа

Заключая сделку по покупке стартапа, игрок может предложить свои условия, что показано на Рисунке 24.

Минимальная цена  
17600

Максимальная цена  
26400

Желаемая цена

ПРЕДЛОЖИТЬ УСЛОВИЯ

0

D20

Рисунок 24 - Сделка

После броска кубика на успешное продавливание своих условий, где окажут существенное влияние характеристики, пользователь может посмотреть подробную статистику по купленному стартапу (Рисунок 25).

**Статистика стартапа "GreenTech Startup 3"**

Цена продажи <b>22 000 Р</b>	Команда <b>20 чел.</b>
Доход за последний месяц <b>6 000 Р</b>	Бюджет <b>8 500 Р</b>
Расходы <b>4 500 Р</b>	Прогресс <b>98%</b>
	Репутация <b>35</b>
Стадия развития: SCALE	

ЗАКРЫТЬ ПРОДАТЬ

Рисунок 25 - Статистика купленного стартапа

С наступлением нового месяца может случиться кризис и игроку будет предоставлен выбор одного из трёх возможных решений (Рисунок 26), где каждое решение окажет существенное влияние на характеристики как игрока, так и его стартапов в зависимости от ниши, в которой случился этот кризис.



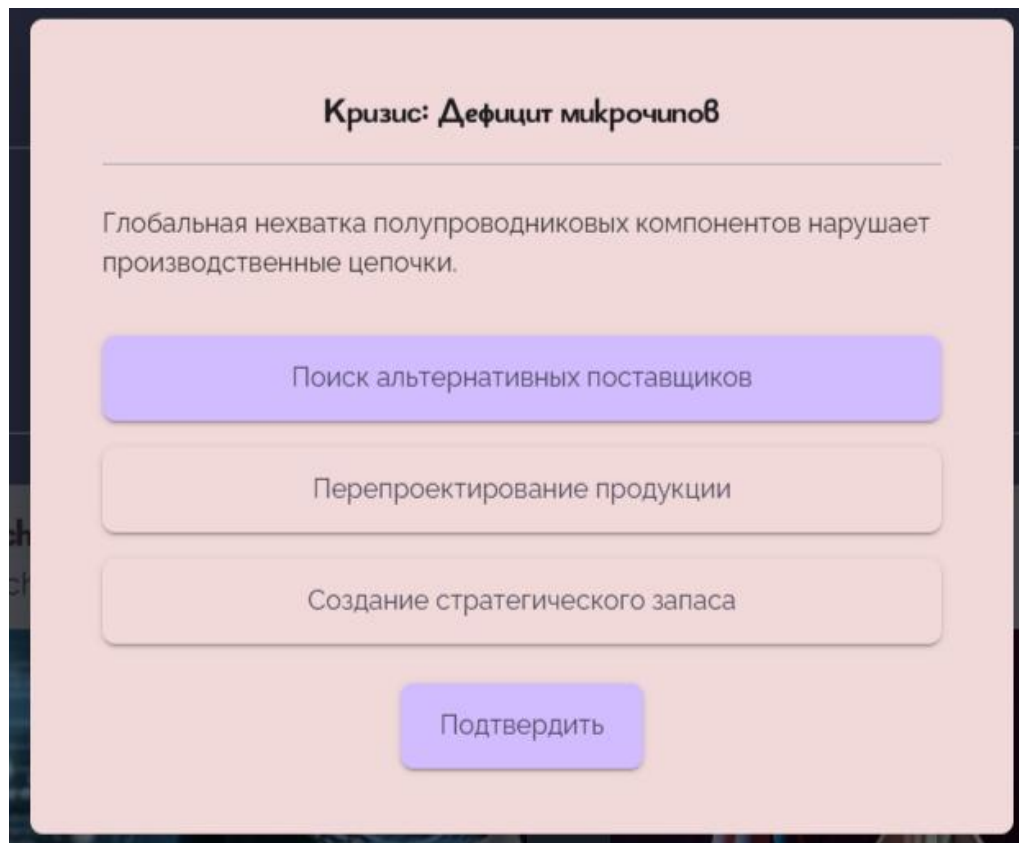


Рисунок 26 - Кризис

## 4 Тестирование

### 4.1 Ручное UI–тестирование

UI–тестирование[9] – этап комплексного тестирования программного обеспечения, направленный на проверку пользовательского интерфейса.

В данном проекте выбран ручной метод тестирования, так как он обеспечивает наилучшее покрытие тестами и позволяет детально оценить взаимодействие пользователя с интерфейсом.

В работе были рассмотрены следующие аспекты данного тестирования:

- визуальные характеристики — проверка адаптивности интерфейса под разные устройства и разрешения экрана;
- композиция — корректность отображения компонентов на странице, их расположение и соответствие дизайн-макетам;
- взаимодействие — обработка событий (клики, скроллинг, ввод данных), работа input-полей и кнопок;
- доступность — удобство навигации, логичность расположения элементов, соответствие стандартам UX/UI;
- пользовательские потоки — проверка полного прохождения ключевых сценариев.

Для каждого аспекта (кейса) тестирования были введены критерии успешности прохождения UI–тестирования для той или иной страницы веб-приложения:

- 0 – тестируемая страница полностью не прошла тестирование;
- 1 – тестируемая страница частично не прошла тестирование;
- 2 – тестируемая страница полностью прошла тестирование.

Тестирование веб–страниц для сценария основных функций веб-приложения приведено в соответствии с таблицей 1.

Таблица 1 -Тестирование основных функций веб-приложения для неавторизованного пользователя

Кейс	Приветственная страница	Логин	Регистрация
Визуальные характеристики	2	2	2
Композиция	2	2	2
Взаимодействие	2	2	2
Доступность	2	2	2
Пользовательские потоки	2	2	2

Тестирование веб–страниц для основных игровых механик приведено в соответствии с таблицей 2, таблицей 3, таблицей 4, таблицей 5, таблицей 6, таблицей 7.

Таблица 2 -Тестирование профиля

Кейс	Профиль	Старт игровой сессии
Визуальные характеристики	1	1
Композиция	2	2
Взаимодействие	2	2
Доступность	2	2
Пользовательские потоки	2	2

Таблица 3 -Тестирование взаимодействия со стартапами

Кейс	Просмотр стартапов	Заклучение сделки	Проведение экспертизы
Визуальные характеристики	2	2	2
Композиция	2	2	2

Взаимодействие	2	2	2
Доступность	2	2	2
Пользовательские потоки	2	2	2

Таблица 4 -Тестирование взаимодействия с мероприятиями

Кейс	Просмотр мероприятий	Посещения мероприятия
Визуальные характеристики	2	2
Композиция	2	2
Взаимодействие	2	2
Доступность	2	2
Пользовательские потоки	2	2

Таблица 5 -Тестирование взаимодействия с кризисами

Кейс	Просмотр кризиса	Выбор решения
Визуальные характеристики	2	1
Композиция	2	2
Взаимодействие	2	2
Доступность	2	2
Пользовательские потоки	2	2

Таблица 6 -Тестирование просмотра характеристик

Кейс	Просмотр экспертности	Просмотр финансов	Просмотр репутации	Просмотр очков действий
Визуальные характеристики	2	1	2	2
Композиция	2	2	2	2
Взаимодействие	2	2	2	2

Доступность	2	2	2	2
Пользовательские потоки	2	2	2	2

Таблица 7 -Тестирование взаимодействия с ходом и окончанием игры

Кейс	Смена хода	Завершение игры с сохранением	Завершение игры без сохранения
Визуальные характеристики	2	2	2
Композиция	2	2	2
Взаимодействие	2	2	2
Доступность	2	2	2
Пользовательские потоки	2	2	2

## Заключение

В ходе выполнения курсового проекта командой было разработано веб-приложение TechTrackInvest — интерактивная игра для обучения инвестированию в стартапы с использованием механик случайных событий. Проект соответствует основным целям и задачам, а также требованиям, зафиксированным в техническом задании.

На этапе проектирования был проведён детальный анализ предметной области, определены ключевые функциональные и нефункциональные требования, разработана архитектура системы, включая UML-диаграммы и ER-модели баз данных.

В процессе реализации были успешно решены следующие задачи:

- регистрация;
- авторизация;
- система стартапов с динамической генерацией характеристик;
- механика кризисов и мероприятий, влияющих на игровой процесс;
- система очков действий и пошагового управления;
- Backend: Java Spring Boot, PostgreSQL, MongoDB;
- Frontend: TypeScript, React, Material UI;
- JWT-аутентификация, защита от SQL-инъекций и XSS-атак;
- поддержка вертикального и горизонтального масштабирования.

Разработанное приложение предоставляет пользователям интуитивно понятный интерфейс, соответствующий современным стандартам UX/UI, и позволяет в игровой форме освоить основы инвестирования, стратегического планирования и управления ресурсами.

## Список используемых источников

1. CashGo [Электронный ресурс]. Режим доступа: <https://cashgo.ru/> - Заглавие с экрана. (Дата обращения: 26.04.2025).
2. BuildYourStax [Электронный ресурс]. Режим доступа: <https://buildyourstax.com/> - Заглавие с экрана. (Дата обращения: 26.04.2025).
3. Симулятор инвестора (от medoborodyj) [Электронный ресурс]. Режим доступа: <https://yandex.ru/games/app/176658/> - Заглавие с экрана. - (Дата обращения: 26.04.2025).
4. Java documentation [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/en/java/>.
5. Spring Framework Documentation [Электронный ресурс]. – Режим доступа: <https://docs.spring.io/spring-framework/reference/index.html>.
6. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.
7. Документация JavaScript-библиотеки React.js [Электронный ресурс]. Режим доступа: <https://ru.react.js.org/docs/getting-started.html>
8. Основы HTML [Электронный ресурс] Режим доступа: <https://html5book.ru/osnovy-html/>
9. Ручное UI-тестирование [Электронный ресурс]. Режим доступа: <https://ux-journal.ru/kak-provodit-ui-testirovanie-sravnenie-instrumentov.html>.