

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук
Кафедра программирования и информационных технологий

Техническое задание
на разработку приложения
«Многопользовательская игра escadre.io»

Исполнители

_____ И. Семькин
_____ Д. Александров
_____ А. Сопельняк
_____ Н. Ларичев
_____ А. Костин

Заказчик

_____ В.С. Тарасов

Воронеж 2025

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
1 Термины и сокращения	4
2 Общие положения	6
2.1 Полное наименование системы и ее условное обозначение	6
2.2 Разработчики и заказчик.....	6
2.3 Перечень документов	6
2.4 Плановые сроки начала и окончания работ.....	7
2.5 Порядок контроля и приема работ по созданию приложения.....	7
2.6 Цели и назначение создания приложения	7
2.6.1 Цели создания приложения.....	7
2.6.2 Назначение приложения	8
3 Функциональные требования	9
3.1 Требования к Авторизации и Регистрации	9
3.2 Требования к игровому процессу	9
3.3 Требования к системе администрирования	10
4 Нефункциональные требования.....	11
4.1 Требования к архитектуре приложения	11
4.2 Требования к программному обеспечению приложения.....	12
4.3 Общие технические требования	13
4.3.1 Целевые устройства и совместимость.....	13
4.3.2 Производительность.....	13
4.3.3 Масштабируемость и поддержка.....	14
4.3.4 Надёжность.....	14
4.3.5 Безопасность	14
4.3.6 Оформление и дизайн	14
4.3.7 Локализация.....	15
5 Группы пользователей.....	15
6 Описание интерфейсов приложения	16
6.1 Экран авторизации и выбора сервера.....	16
6.1.1 Экран для незарегистрированного пользователя	16
6.1.2 Экран для зарегистрированного пользователя	16
6.1.3 Экран регистрации	17
6.2 Основной игровой интерфейс	17

6.3 Интерфейс игрового магазина	17
6.4 Интерфейс для расстановки кораблей	18
6.5 Интерфейс для выхода на экран авторизации	18
7 Пользовательские сценарии	19
Приложение	23

1 Термины и сокращения

В настоящей работе используются следующие термины и сокращения с соответствующими определениями:

- **Пользователь** – человек, пользующийся функционалом приложения;
- **Администратор** – человек, имеющий доступ к расширенному функционалу приложения;
- **Игра** – интерактивное взаимодействие между игроками или между игроком и системой;
- **Игрок** – участник игры, взаимодействующий с игровым миром.
- **Никнейм** – псевдоним игрока;
- **Скролл** – перемещение содержимого экрана;
- **Low poly** – стиль 3D-графики с минимальным количеством полигонов;
- **Лидерборд** – таблица лучших игроков в текущей игровой сессии;
- **Эскадра** – группа кораблей, которой управляет игрок;
- **Клиентская часть** – программно–аппаратная часть вебприложения или мобильного приложения. Отвечает за функционирование внутренней части веб-приложения или мобильного приложения;
- **Сервер, серверная часть** – компьютер, обслуживающий другие компьютеры (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач;
- **Dependency Injection** – паттерн программирования, при зависимости передаются объекту извне, а не создаются внутри него;
- **SignalR** – библиотека для .NET, упрощающая создание веб-приложений с реальным временем;

- **GPU Instancing** – техника рендеринга, позволяющая отображать множество одинаковых 3D-объектов с минимальными затратами ресурсов;
- **UI** – User Interface (Интерфейс пользователя) ;
- **ПО** – программное обеспечение.

2 Общие положения

2.1 Полное наименование системы и ее условное обозначение

Полное наименование приложения: «Многопользовательская игра про морские баталии escadre.io с мобильной и веб-версией».

Условное обозначение приложения: «escadre.io».

2.2 Разработчики и заказчик

Заказчик: старший преподаватель Тарасов Вячеслав Сергеевич, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

Исполнитель: 2 команда группы 2

Состав команды исполнителя:

- Семькин Иван;
- Александров Дмитрий;
- Сопельняк Андрей;
- Ларичев Никита;
- Нагорных Константин;
- Костин Артем.

2.3 Перечень документов

Данное приложение будет создаваться на основе следующих документов:

- федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ;
- настоящее техническое задание, составленное в соответствии с ГОСТ 34.602-2020 и ГОСТ 7.32.

2.4 Плановые сроки начала и окончания работ

Плановый срок начала работ – 5 марта 2025 г.

Плановый срок окончания работ – 28 июня 2025.

2.5 Порядок контроля и приема работ по созданию приложения

Предварительные отчёты по работе будет проводиться во время рубежных аттестаций:

- Первая аттестация (конец марта 2025) – создан репозиторий проекта на GitHub, распределены задачи проекта в таск-менеджере Trello, предоставлены диаграммы общей логики системы, предоставлено готовое техническое задание;
- Вторая аттестация (конец апреля 2025) – написан программный код, реализующий 60% описанной функциональности приложения, реализована БД и ее взаимодействие с сервером, проведена отладка и доработка кода программы, проведено тестирование по работе системы;
- Третья аттестация (конец мая 2025) – разработан курсовой проект, выполнены завершающие работы по доработке приложения, предоставлена готовая система.

2.6 Цели и назначение создания приложения

2.6.1 Цели создания приложения

Целями создания приложения являются:

- предоставить пользователям увлекательный игровой опыт, эмоции и новый вариант игры для развлечения;
- получение прибыли с интегрированной сторонней рекламы.

2.6.2 Назначение приложения

Приложение позволяет решать следующие задачи:

- управлять своей эскадрой;
- атаковать других игроков;
- собирать игровые ресурсы;
- прокачивать свою эскадру;
- просматривать лидерборд;
- просматривать личную игровую статистику.

3 Функциональные требования

3.1 Требования к Авторизации и Регистрации

- игрок может находиться в неавторизованном режиме;
- при первом входе в игру пользователь может зарегистрироваться и продолжить сеанс в авторизованной режиме;
- при повторном входе в игру пользователь может авторизоваться в ранее созданный аккаунт;
- при регистрации приложение запрашивает адрес электронной почты пользователя с созданием пароля;
- при авторизации приложение запрашивает сохраненные в системе данные пользователя;
- при неверном вводе данных пользователь будет оповещен.

3.2 Требования к игровому процессу

- игрок может ввести никнейм перед началом игры;
- игрок может выбрать сервер перед началом игры;
- игрок может управлять своей эскадрой, задавая ей точку направления движения путем нажатия на экран;
- игрок может атаковать эскадры других игроков, задавая цели своим кораблям;
- игрок может собирать ресурсы для улучшения своей эскадры;
- игрок может улучшать уже имеющиеся и/или покупать новые корабли за собранные ресурсы;
- игрок может задавать расстановку кораблей в своей эскадре;
- игрок может выйти в стартовое меню игры;
- игрок может просматривать свою игровую статистику.

3.3 Требования к системе администрирования

- администратор может свободно наблюдать за игровым процессом каждого из игроков;
- администратор может отключать игроков от игры, а также производить блокировку их аккаунтов.

4 Нефункциональные требования

4.1 Требования к архитектуре приложения

Приложение должно иметь слоистую архитектуру и строгое разделение на слои. Архитектура должна учитывать особенности различных платформ (мобильных устройств и веб-браузеров), для этого должно быть предусмотрена инкапсуляция платформо-зависимого кода в отдельные модули.

Подразумеваются следующие компоненты архитектуры:

- a) Клиент:
 - World View
 - UI View
 - Controller
 - Client Model
 - Client Network
- b) Сервер:
 - Server Network
 - Controller
 - Server Model
- c) Мастер-сервер

Необходима модель обмена данными:

- использование событийной модели для обмена данными между слоями
- двунаправленный обмен контроллерами между клиентом и сервером

Предусмотрено распределение ответственности:

- View-слои отвечают исключительно за визуализацию (презентацию) данных;

- Controller-слои управляют игровой логикой и обработкой пользовательского ввода;
- Model-слои содержат бизнес-логику и состояние игры;
- Network-слои абстрагируют сетевые операции от остальной логики.

Компоненты должны быть спроектированы со следующими требованиями:

- архитектура должна обеспечивать повторное использование компонентов;
- компоненты должны иметь низкую связность и высокую сплоченность;
- допустимо применение подходящих архитектурных решений по мере выявления потребностей проекта.

Взаимодействие компонентов строится по следующим требованиям:

- минимизация связности между компонентами с использованием Dependency Injection;
- реализация сетевого взаимодействия между клиентом и сервером в реальном времени через SignalR;
- четко определенные интерфейсы между слоями для обеспечения гибкости и тестируемости.

4.2 Требования к программному обеспечению приложения

Приложение должно иметь слоистую архитектуру с четким разделением ответственности, а также модель Клиент-Серверного взаимодействия.

Для реализации клиентской части приложения будет использоваться следующее ПО:

- Unity 2021 LTS в качестве основного игрового движка;
- Язык программирования C#.

Для реализации серверной части приложения будет использоваться следующее ПО:

- ASP.NET Core;
- SignalR для сетевой коммуникации;
- СУБД PostgreSQL;
- Docker для автоматизации развертывания и управления;
- Язык программирования C#.

4.3 Общие технические требования

4.3.1 Целевые устройства и совместимость

- мобильное приложение должно поддерживать Android 7.0+;
- WebGL-версия должна работать в современных браузерах (Chrome, Firefox, Safari, Edge);
- UI должен корректно отображаться на экранах с различным соотношением сторон и разрешениями.

4.3.2 Производительность

- клиентское приложение должно обеспечивать стабильную частоту кадров не менее 30 FPS на целевых мобильных устройствах;
- время загрузки приложения не должно превышать 10 секунд на целевых устройствах;
- оптимизация с использованием GPU Instancing для эффективного рендеринга множества сущностей;
- WebGL-версия должна демонстрировать производительность не ниже 40 FPS в браузерах на десктопных системах;
- время обработки сервером запросов пользователя и отправки ответов не должно превышать 100мс при допустимом количестве игроков.

4.3.3 Масштабируемость и поддержка

- архитектура должна позволять горизонтальное масштабирование серверов игры;
- мастер-сервер должен эффективно распределять нагрузку между игровыми серверами;
- архитектура должна позволять легко добавлять новые игровые функции;
- модульная структура приложения для возможности независимой доработки и тестирования компонентов.

4.3.4 Надёжность

- серверная часть должна иметь доступность не менее 99% времени;
- клиентское приложение должно корректно обрабатывать потерю соединения с возможностью переподключения;
- система должна обеспечивать сохранность данных пользователей при сбоях.

4.3.5 Безопасность

- коммуникация между клиентом и серверами должна использовать защищённое соединение (TLS);
- аутентификация пользователей должна соответствовать современным стандартам безопасности.

4.3.6 Оформление и дизайн

- приложение должно быть оформлено в едином стиле;
- интерфейс приложения должен быть понятным;
- приложение должно быть разработано в одной цветовой палитре с использованием ограниченного набора шрифтов;

- компоненты игры должны быть выполнены в 3D графике и low poly стилистике;
- необходимо корректное и одинаковое отображение приложения на экранах различного размера.

4.3.7 Локализация

- приложение должно поддерживать русский язык;
- приложение должно поддерживать возможность локализации на разные языки;
- UI должен корректно отображать тексты разной длины при локализации.

5 Группы пользователей

Предусмотрены следующие группы пользователей:

- неавторизованный пользователь – пользователь, не зарегистрированный в системе или не прошедший авторизацию;
- авторизованный пользователь – пользователь, прошедший авторизацию системы;
- администратор – авторизованный пользователь с возможностью модерирования системы.

6 Описание интерфейсов приложения

6.1 Экран авторизации и выбора сервера

6.1.1 Экран для незарегистрированного пользователя

Экран отображается перед любым пользователем, который зашел в приложение. На нем находятся следующие элементы:

- кнопка с выпадающим списком для выбора режима входа: позволяет пользователю переключать режимы “С аккаунтом” и “Без аккаунта”. Изначально выставлен режим “Без аккаунта”. Режимы “С аккаунтом” переносит пользователя на экран для зарегистрированных пользователей;
- кнопка с выпадающим списком для выбора сервера;
- поле для ввода никнейма;
- кнопка для начала игры.

6.1.2 Экран для зарегистрированного пользователя

Экран отображается, если вошедший в приложение пользователь переключил режим входа на “С аккаунтом”. На нем находятся следующие элементы:

- кнопка с выпадающим списком для выбора режима входа;
- кнопка с выпадающим списком для выбора сервера;
- поле для ввода никнейма;
- поле для ввода пароля;
- кнопка для начала игры;
- кнопка для перехода на экран регистрации.

6.1.3 Экран регистрации

Экран отображается, если вошедший в приложение пользователь переключил режим входа на “С аккаунтом” и нажал на кнопку регистрации. На нем находятся следующие элементы:

- поле для ввода никнейма;
- поле для ввода электронной почты;
- поле для ввода пароля;
- кнопка “Зарегистрироваться”.

6.2 Основной игровой интерфейс

Интерфейс отображается перед игроком после того, как он ввел все необходимые данные и нажал на кнопку начала игры. На нем находятся следующие элементы:

- кнопка игрового магазина: открывает интерфейс игрового магазина;
- кнопка расстановки кораблей: открывает интерфейс расстановки кораблей;
- кнопка выхода на экран авторизации: открывает интерфейс выхода на экран авторизации;
- счетчик игровых ресурсов, которые собрал игрок;
- лидерборд.

6.3 Интерфейс игрового магазина

Интерфейс отображается перед игроком после того, как он нажал на кнопку игрового магазина. На нем находятся следующие элементы:

- верхний скролл, который содержит список кораблей для покупки;
- нижний скролл, который содержит список улучшений для имеющихся кораблей;

- счетчик игровых ресурсов, которые собрал игрок;
- элемент для выхода из магазина.

6.4 Интерфейс для расстановки кораблей

Интерфейс отображается перед игроком после того, как он нажал на кнопку расстановки кораблей. На нем находятся следующие элементы:

- круг, в границах которого производится расстановка;
- символически обозначенные позиции каждого корабля, которые можно перетаскивать;
- зона, при перетаскивании в которую корабль будет продан (списан).

6.5 Интерфейс для выхода на экран авторизации

Интерфейс отображается перед игроком после того, как он нажал на кнопку выхода на экран авторизации. На нем находятся следующие элементы:

- плашка подтверждения “Вы уверены, что хотите выйти?”;
- кнопка “Да”;
- кнопка “Нет”.

7 Пользовательские сценарии

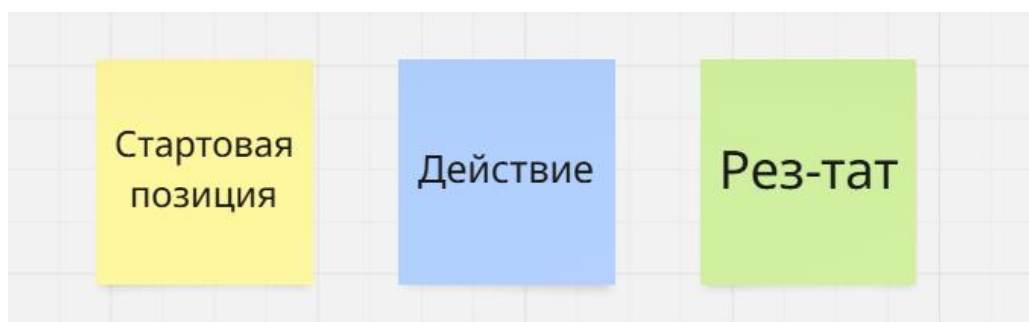


Рисунок 1 - Пояснение к обозначениям

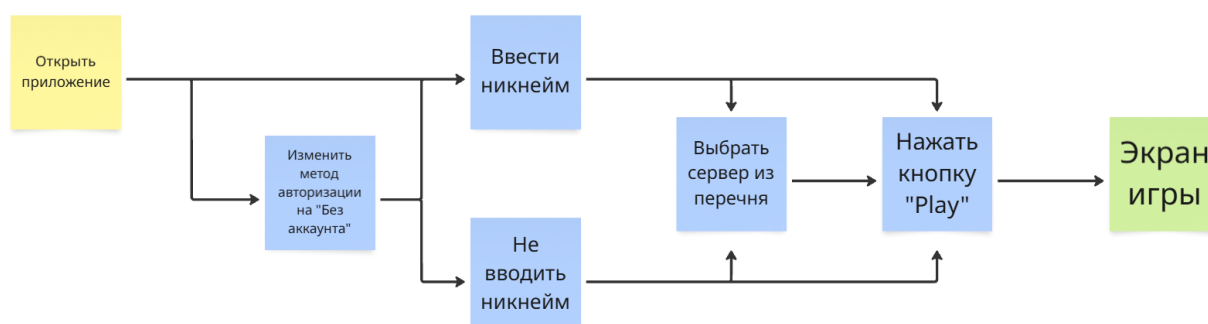


Рисунок 2 - Зайти в игру без аккаунта

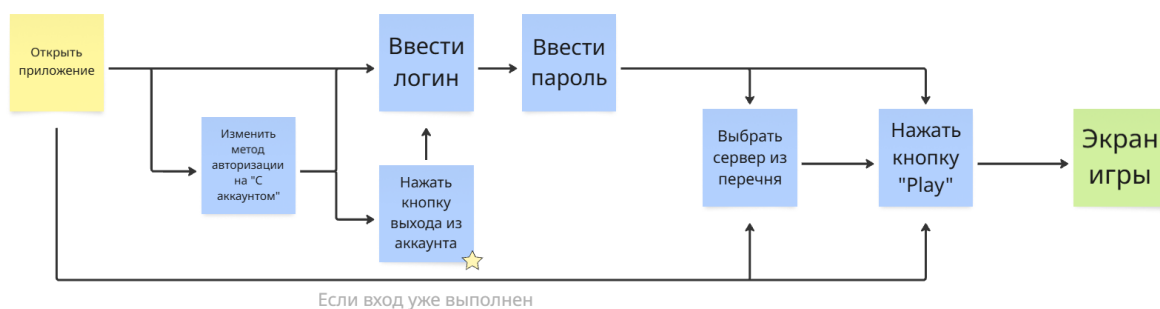


Рисунок 3 - Зайти в игру с аккаунтом

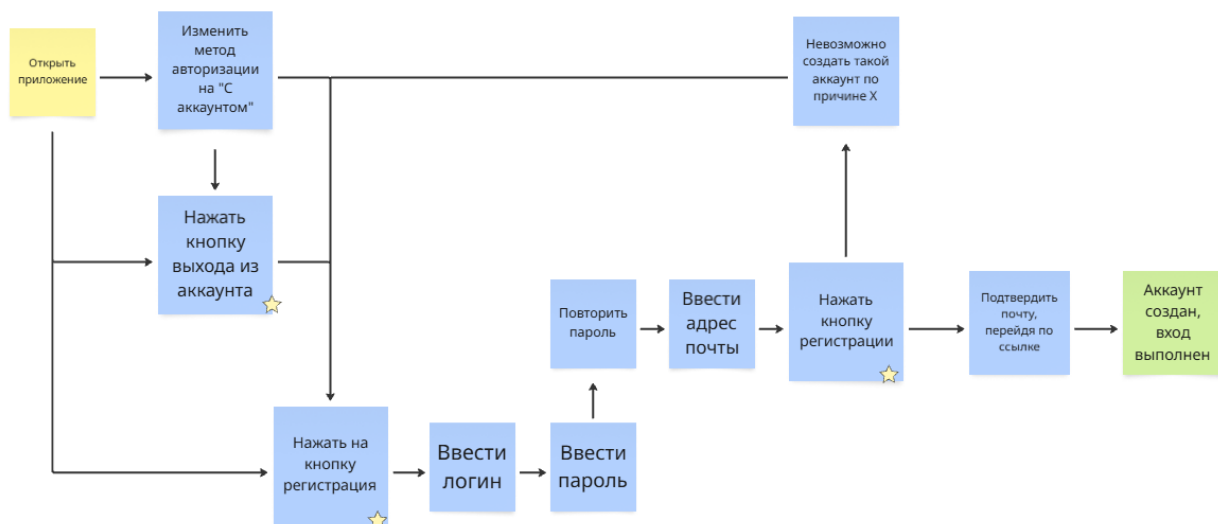


Рисунок 4 - Регистрация

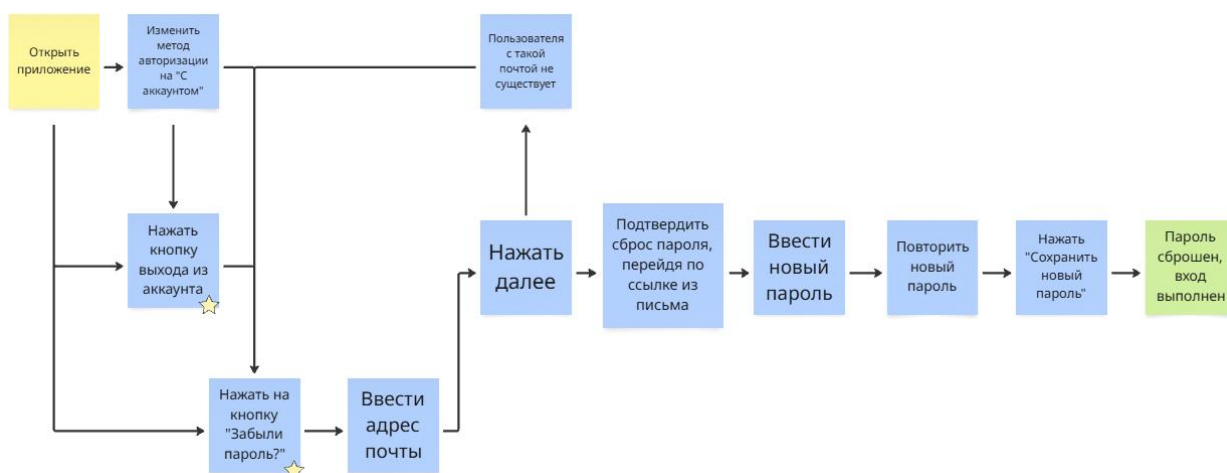


Рисунок 5 - Восстановление пароля от аккаунта



Рисунок 6 - Улучшить корабль

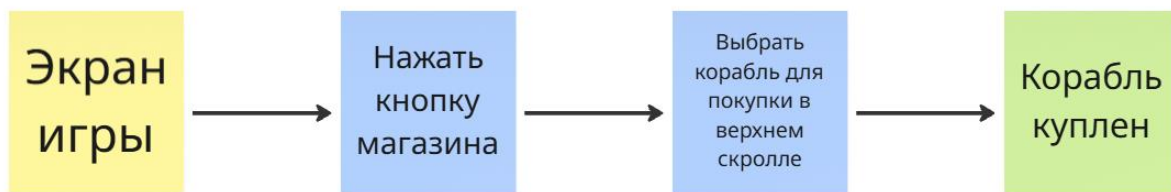


Рисунок 7 - Докупить корабль

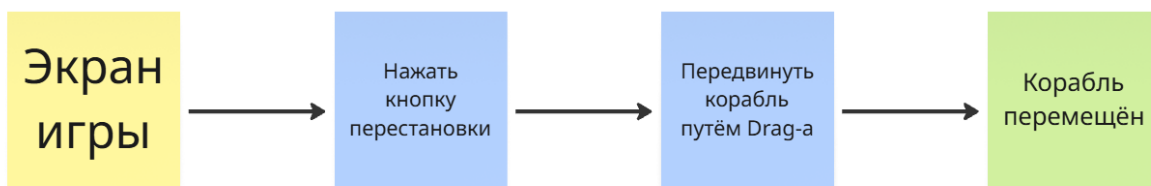


Рисунок 8 - Переставить корабль

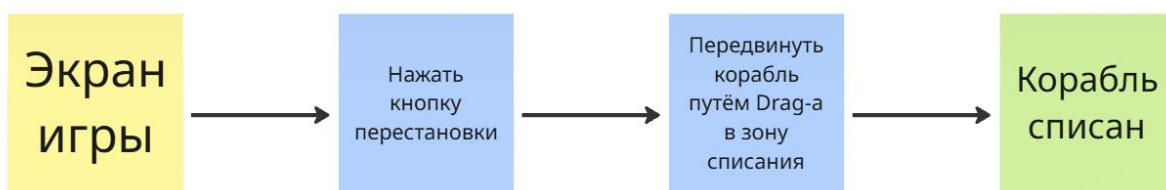


Рисунок 9 - Списать корабль

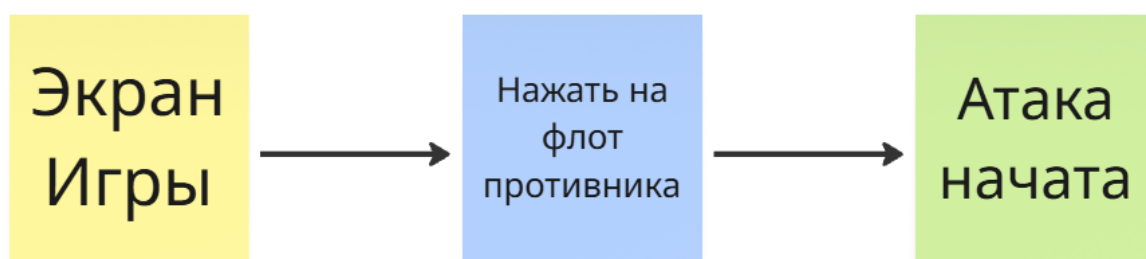


Рисунок 10 - Атаковать флот противника

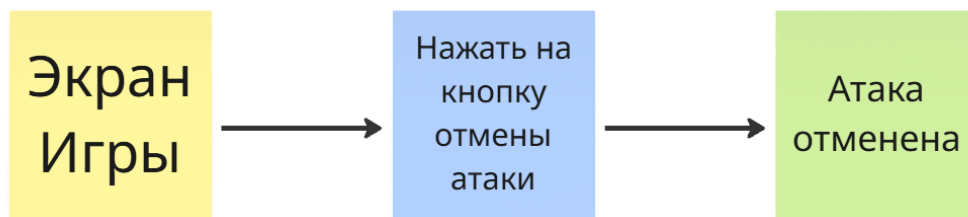


Рисунок 11 - Отменить атаку на всех противников



Рисунок 12 - Переключить звук



Рисунок 13 - Выйти на экран входа

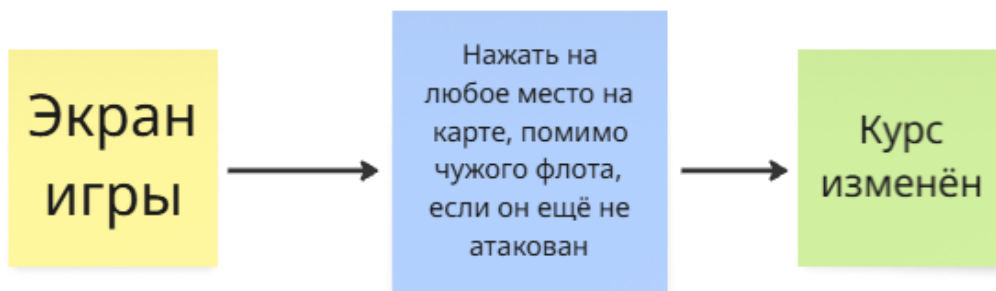


Рисунок 14 - Установить курс эскадре

Приложение

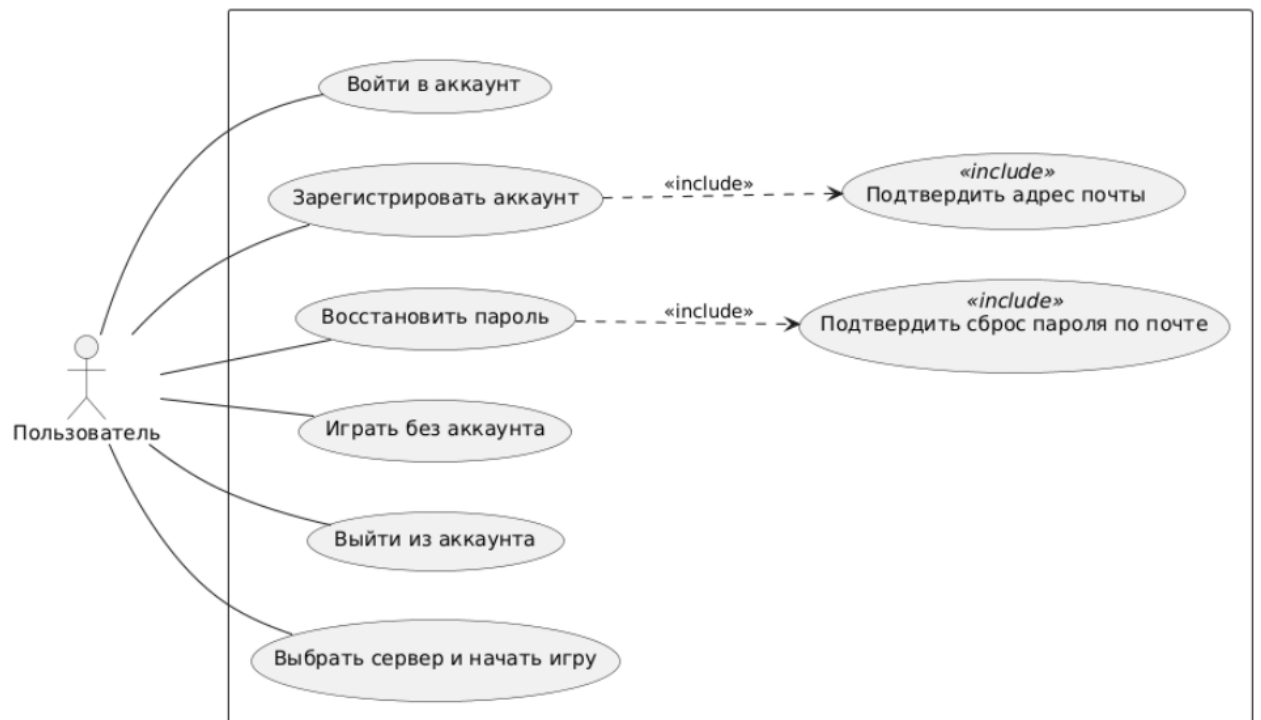


Рисунок 15 - Диаграмма прецедентов главного меню



Рисунок 16 - Диаграмма прецедентов игрового процесса

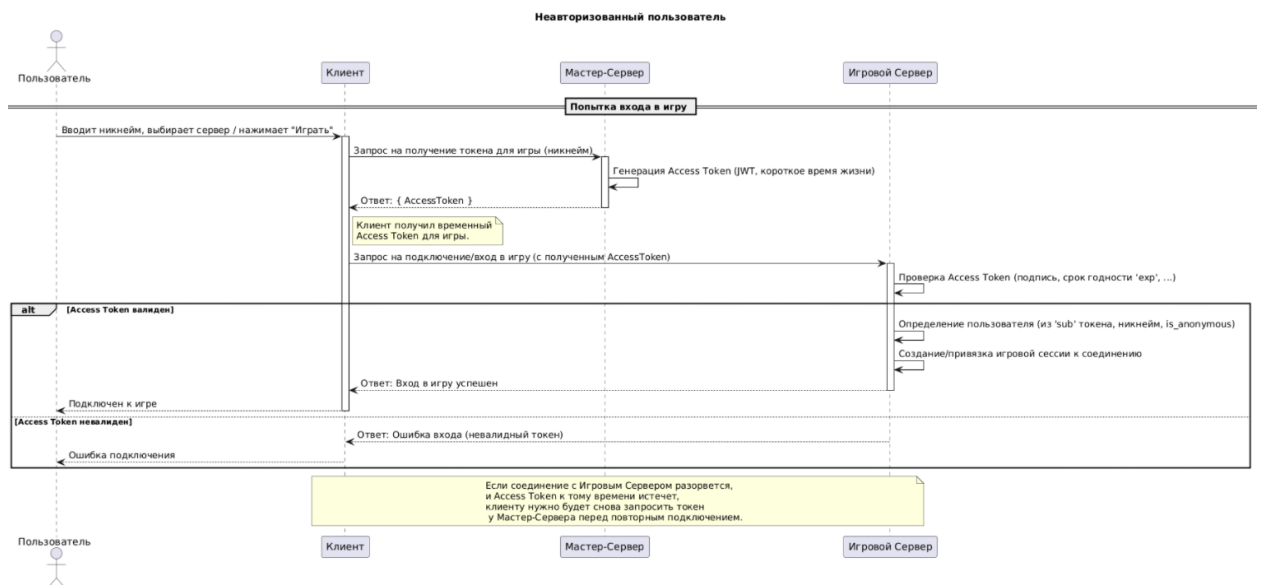


Рисунок 17 - Диаграмма последовательности входа в игру неавторизованного пользователя

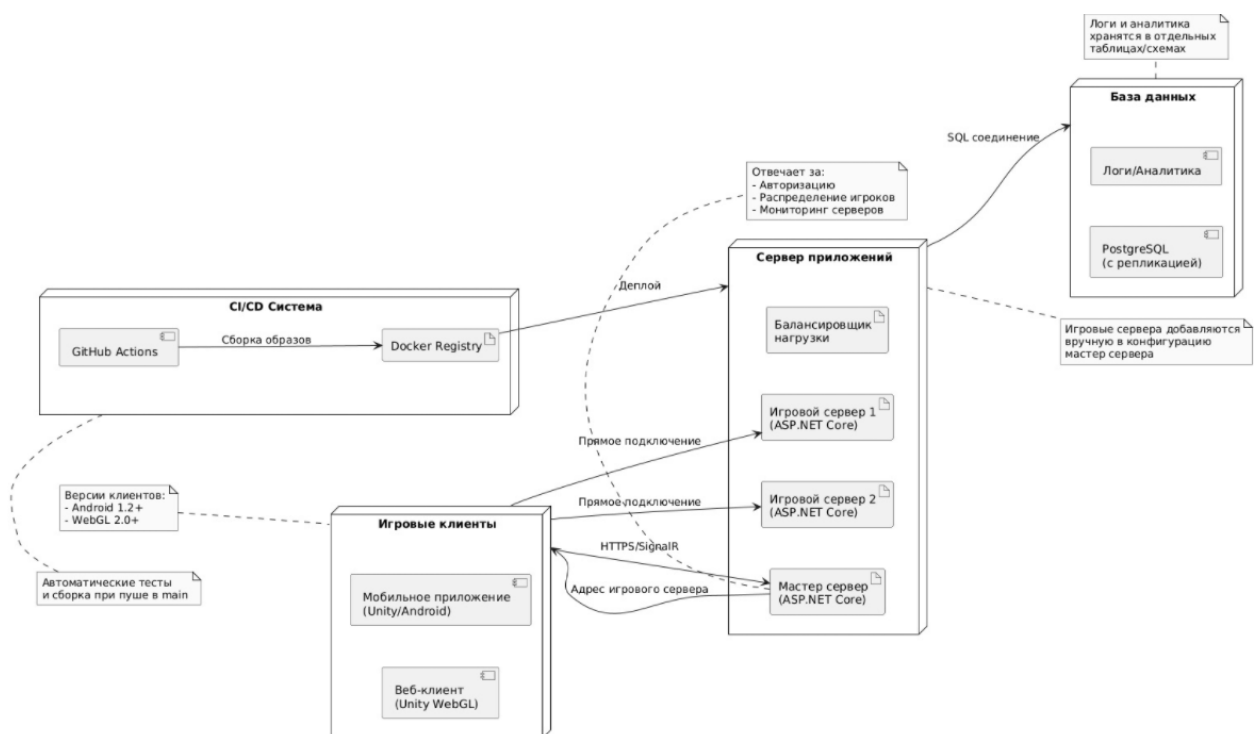


Рисунок 18 - Диаграмма развёртывания

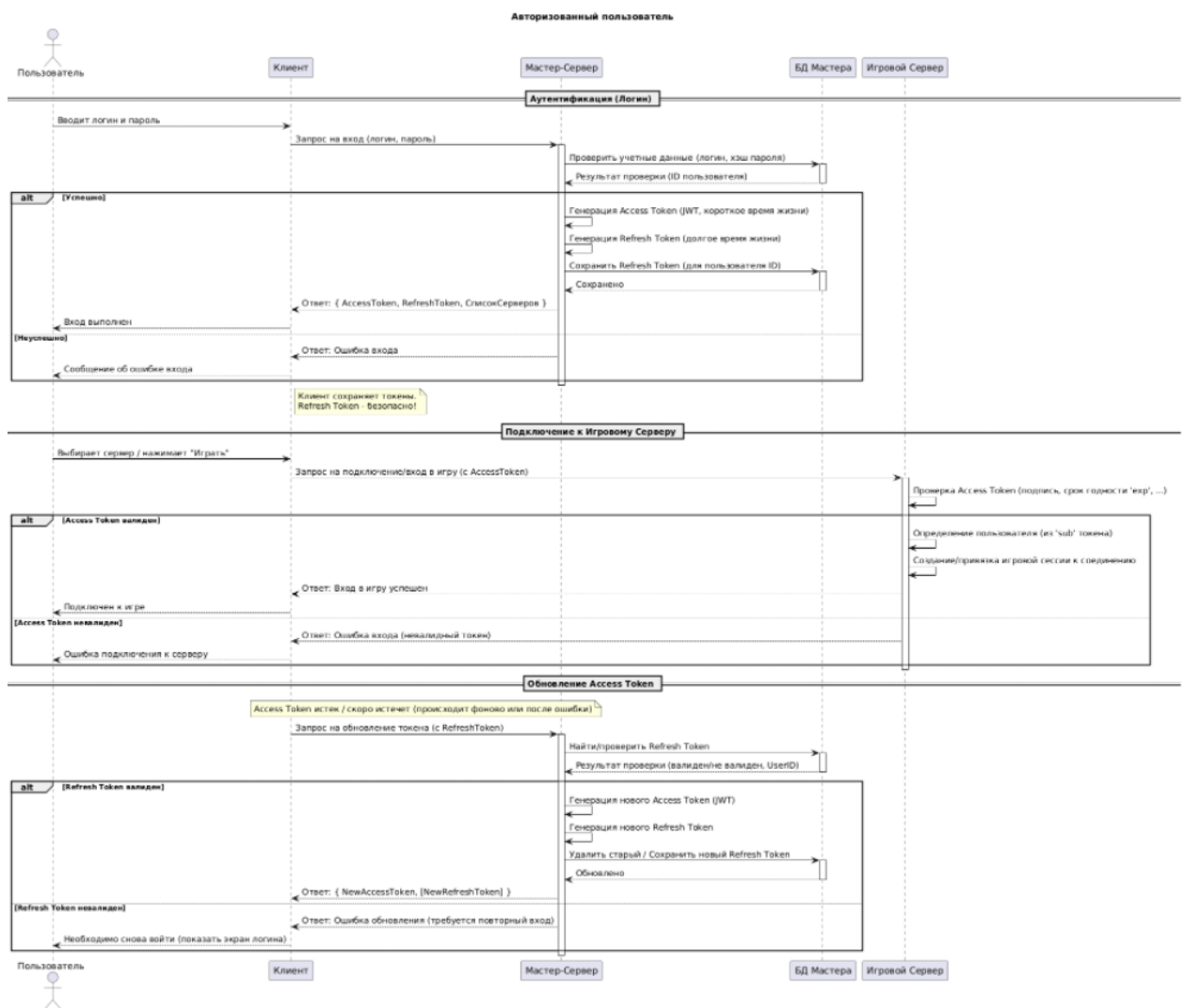


Рисунок 19 - Диаграмма последовательности входа в игру авторизованного пользователя