

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 Программная инженерия Отделение
школы (НОЦ) Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка комплексного модуля для проведения технических соревнований между разработчиками, основанного на симметричной коммуникации

УДК 004.422.833:004.925.84

Студент

Группа	ФИО	Подпись	Дата
8К61	Соловьева Анастасия Дмитриевна		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Саврасов Фёдор Витальевич	к.т.н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСНГ ШБИП	Спицына Любовь Юрьевна	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Белоенко Елена Владимировна	к.т.н.		

ДОПУСТИТЬ ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Евгений Сергеевич	к.т.н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результата	Результат обучения (выпускник должен быть готов)
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
P11	Демонстрировать способность к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
 Направление подготовки 09.03.04 Программная инженерия
 Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Чердынцев Е.С.
 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

Студенту:

Группа	ФИО
8К61	Соловьевой Анастасии Дмитриевной

Тема работы:

Разработка комплексного модуля для проведения технических соревнований между разработчиками, основанного на симметричной коммуникации	
Утверждена приказом директора (дата, номер)	№59-51/с от 28.02.2020 г.

Срок сдачи студентом выполненной работы:	11.06.2020
------------------------------------------	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Литература по SignalR Исследования и статьи на тему эффективного поиска IT-специалистов. Стратегии игры в «Морской бой» и их эффективность.
---------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------

Перечень подлежащих исследованию, проектированию и разработке вопросов	Изучение способов эффективной коммуникации между частями программного модуля в режиме реального времени. Разработка алгоритма для эффективной игры в программе-игре «Ганчики». Разработка программного модуля для проведения технических соревнований между разработчиками. Разработка шаблонов для участников хакатона
Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i>	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Доцент ОСГН Спицына Любовь Юрьевна
Социальная ответственность	Доцент ОБД Белоенко Елена Владимировна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	10.10.2019
-------------------------------------------------------------------------------------------------	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Саврасов Фёдор Витальевич	к.т.н.		10.10.2019

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К61	Соловьева Анастасия Дмитриевна		10.10.2019

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
 Направление подготовки 09.03.04 Программная инженерия
 Отделение школы (НОЦ) Отделение информационных технологий
 Период выполнения осенний/весенний семестр 2019/2020 учебного года

Форма представления работы:

Бакалаврская работа

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	10.06.2020
------------------------------------------	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
10.11.2019	Исследование предметной области	20
20.02.2020	Проектирование программного модуля	20
20.03.2020	Разработка программного модуля	20
02.06.2020	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	20
02.06.2020	Социальная ответственность	20

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Саврасов Фёдор Витальевич	К.Т.Н.		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Евгений Сергеевич	К.Т.Н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8К61	Соловьевой Анастасии Дмитриевной

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Бюджет проекта - не более 300 тыс. рублей Затраты на специальное оборудование – 55 944 руб Затраты по основной заработной плате исполнителей темы – 101 541 руб Отчисления во внебюджетные фонды – 48 449 руб Накладные расходы составили 32 949 рублей
2. <i>Нормы и нормативы расходования ресурсов</i>	Значение показателя интегральной финансовый показатель разработки– не более 1,95 Значение показателя интегральной ресурсоэффективности разработки – не менее 4 Значение показателя интегральной эффективности – не менее 4,5
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Коэффициент отчислений во внебюджетные фонды – 22%. Коэффициент накладных расходов – 16% от всех остальных затрат

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	Анализ конкурентных технических решений, QuaD-анализ, SWOT-анализ
2. <i>Планирование и формирование бюджета научных исследований</i>	Планирование этапов работ, определение трудоемкости и построение календарного графика, формирование бюджета
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	Оценка сравнительной эффективности исследования

Перечень графического материала (с точным указанием обязательных чертежей):

1. <i>Оценка конкурентоспособности технических решений</i>
2. <i>Матрица SWOT</i>
3. <i>Альтернативы проведения НИ</i>
4. <i>График проведения и бюджет НИ</i>
5. <i>Оценка ресурсной, финансовой и экономической эффективности НИ</i>

Дата выдачи задания для раздела по линейному графику	10.03.2020
-------------------------------------------------------------	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Спицына Любовь Юрьевна	К. Э. Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К61	Соловьева Анастасия Дмитриевна		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8К61	Соловьевой Анастасии Дмитриевной

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

Разработка комплексного модуля для проведения технических соревнований между разработчиками, основанного на симметричной коммуникации

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является программный модуль для проведения технических соревнований между разработчиками, который используется для поиска сотрудников во внутренний штат компании
--------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. Правовые и организационные вопросы обеспечения безопасности:</p> <ul style="list-style-type: none"> • специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; • организационные мероприятия при компоновке рабочей зоны. 	<p>Трудовой кодекс Российской Федерации от 30.12.2001 (ред. от 24.04.2020)</p> <p>ГОСТ 12.1.005 – 88 Оборудование производственное. Ограждения защитные</p> <p>ГОСТ 12.2.062-81 Система Общие санитарно-гигиенические требования к воздуху рабочей зоны стандартов безопасности труда</p> <p>ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя.</p> <p>СанПиН 2.2.4.3359-16 "Санитарно-эпидемиологические требования к физическим факторам на рабочих местах"</p> <p>СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений»</p> <p>Санитарно-эпидемиологические правила и нормативы СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»</p> <p>Санитарно-эпидемиологические правила и нормативы СанПиН 2.2.2.1332-03 «Гигиенические требования к организации работы на копировально-множительной технике»</p> <p>СП 44.13330.2011 Административные и бытовые здания. Актуализированная редакция СНиП 2.09.04-87</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	СанПиН 2.1.2.2645-10 Санитарно-эпидемиологические требования к условиям проживания в жилых зданиях и помещениях ГОСТ 12.0.003-2015 «Опасные и вредные производственные факторы. Классификация»
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	1. Отклонение показателей микроклимата 2. Превышение уровня шума 3. Отсутствие или недостаток естественного света 4. Недостаточная освещенность рабочей зоны
3. Экологическая безопасность:	Воздействие на литосферу с помощью бытовых отходов и макулатуры
4. Безопасность в чрезвычайных ситуациях:	Пожар в офисном помещении, способы защиты и ликвидации последствий

Дата выдачи задания для раздела по линейному графику	10.03.2020
------------------------------------------------------	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Белоенко Елена Владимировна	К. Т. Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К61	Соловьева Анастасия Дмитриевна		

Содержание

Реферат	13
Термины и определения	14
Глава 1. Исследование предметной области	15
1.1 Определение задачи в контексте рассматриваемой предметной области	15
1.2 Постановка целей	18
Глава 2. Проектирование программного комплекса	19
2.1 Разработка и анализ требований	19
2.1.1 Функциональное назначение	19
2.1.2 Эксплуатационное назначение	19
2.1.3 Нефункциональные требования	19
2.1.4 Требования к внутренней коммуникации	19
2.1.5 Требования к функциональным характеристикам	20
2.1.6 Правила игры	22
2.2 Выбор технологий разработки	25
2.2.1 Single-page Application (SPA)	25
2.2.2 React	26
2.2.3 Redux	27
2.2.4 SignalR	29
2.2.5 TypeScript	31
2.2.6 Styled Components	31
Глава 3. Разработка программного модуля	33
3.1 Общая архитектура системы	33
3.2 Внутренняя SignalR-коммуникация	34
3.2.1 Обмен данными сервера с ботом	35

3.2.2	Обмен данными сервера с frontend-частью.....	38
3.2.3	Проблема в организации коммуникации.....	39
3.3	Интерфейс программного модуля.....	40
3.3.1	Личный кабинет участника.....	40
3.3.2	Личный кабинет администратора	42
3.3.3	Демонстрация на больших экранах.....	45
3.4	Тестовый бот-клиент	48
3.4.1	Алгоритм расстановки тестового бота	48
3.4.2	Алгоритм атаки тестового бота	49
3.4.3	Алгоритм уничтожения подбитого танка для тестового бота	51
3.4.4	Статистические результаты для тестового бота	53
3.5	Шаблон (Boilerplate) для участников хакатона.....	56
3.6	Тестирование веб-приложения.....	58
Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....		60
4.1	Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения	60
4.1.2	Потенциальные потребители результатов исследования	60
4.1.3	Анализ конкурентных технических решений.....	61
4.1.4	Технология QuaD	64
4.1.5	SWOT-анализ.....	65
4.2	Определение возможных альтернатив проведения научных исследований	67
4.3	Планирование научно-исследовательских работ	68
4.3.1	Структура работ в рамках научного исследования	68

4.3.2	Определение трудоемкости выполнения работ	69
4.3.3	Разработка графика проведения научного исследования	70
4.4	Бюджет научно-технического исследования (НТИ)	73
4.4.1	Расчет затрат на специальное оборудование для научных работ	73
4.4.2	Основная заработная плата исполнителей	73
4.4.3	Дополнительная заработная плата исполнителей темы	77
4.4.4	Отчисления во внебюджетные фонды (страховые отчисления)	77
4.4.5	Накладные расходы	78
4.4.6	Формирование бюджета затрат проекта.....	78
4.5	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	79
4.6	Выводы по главе.....	80
Глава 5.	Социальная ответственность.....	81
5.1	Введение.....	81
5.2	Правовые и организационные вопросы обеспечения безопасности	82
5.3	Производственная безопасность	83
5.4	Экологическая безопасность.....	88
5.5	Безопасность в чрезвычайных ситуациях	89
5.6	Выводы по главе.....	90
Заключение	91
Список литературы	92

Реферат

Выпускная квалификационная работа содержит 94 страницы, 34 рисунка, 18 таблиц, 23 использованных источника.

Ключевые слова: хакатон, веб-приложение, программный модуль, SignalR, поиск IT-специалистов.

Актуальность разрабатываемого программного обеспечения обусловлена тем, что на данный момент наблюдается сильная нехватка профессионально-подготовленных кадров в IT-сфере. Проблемой многих IT-компаний становится дефицит специалистов, обладающих компетенциями в области информационных технологий.

Целью выпускной квалификационной работы является проектирование и разработка программного комплекса, предназначенного для проведения технических соревнований. Участники такого мероприятия должны разработать клиент-бота, который будет реализовывать алгоритм для игры.

Для достижения поставленной цели необходимо реализовать программный комплекс, который состоит из frontend-части, игрового сервера, тестового бота-клиента, и шаблонов для участников соревнования.

Frontend-часть программного модуля была выполнена с помощью инструментов React, Redux и TypeScript, Игровой сервер – ASP.Net Core, коммуникация между игровым сервером и ботом-клиентом – SignalR, коммуникация между игровым сервером и frontend-частью – SignalR и REST API, бот-клиент и шаблон для участников соревнования – Node.JS.

Термины и определения

1. Хакатон – соревнование для разработчиков, во время которого специалисты разработки программного обеспечения совместно (группой) решают определенную задачу в течение ограниченного времени.
2. Backend – игровой сервер, реализующий логику работы игры.
3. Frontend – клиентская часть, отображающая игровой процесс для действий участников.
4. Бот – программа-клиент, разработанная участником для осуществления обращений к серверу игры в целях выиграть игру по заданному в ней алгоритму.
5. Администратор – сотрудник компании, осуществляющий настройку игровой среды и БД (обнуление результатов, выгрузку статистики и т.п.). В системе интерфейс администратора отсутствует.
6. SignalR – веб-библиотека с открытым исходным кодом, разработанная командой Microsoft ASP.NET. SignalR имеет простой API для вызовов удаленных процедур от сервера к клиенту (RPC server-to-client), которые вызывают Javascript функции в клиентских браузерах из кода.
7. HTTPS – расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.
8. REST – архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа-системы.
9. Boilerplate – шаблонный код или часть кода, которую неоднократно подразумевается использовать во многих случаях практически без изменений.

Глава 1. Исследование предметной области

1.1 Определение задачи в контексте рассматриваемой предметной области

На данный момент рынок IT-сферы постоянно растет и развивается. Появляются не только новые вакансии, но и растет потребность в уже устоявшихся технических профессиях. Вследствие этого наблюдается сильная нехватка профессионально-подготовленных кадров. Проблемой многих IT-компаний становится дефицит специалистов, обладающих компетенциями в области информационных технологий, к ним относятся как рядовые программисты, так и профессионалы на стыке технологических и бизнес-направлений.

Перед IT-компаниями встает задача создать интересное и нестандартное решение для привлечения сторонних разработчиков. HR- и PR-службы ищут как можно больше интересных способов привлечения новых сотрудников в компанию. Основной задачей этих служб является не только привлечение новых сотрудников, но и повышение уровня лояльности и заинтересованности к компании, а также выстраивание доверительных отношений с потенциальными кандидатами для дальнейшего сотрудничества.



Рисунок 1. Диаграмма Fishbone

Как правило, у компании существует несколько основных способов для привлечения новых разработчиков:

- конференции, митапы;
- размещение вакансии на сайте по подбору специалистов;
- стажировки для студентов и молодых разработчиков.

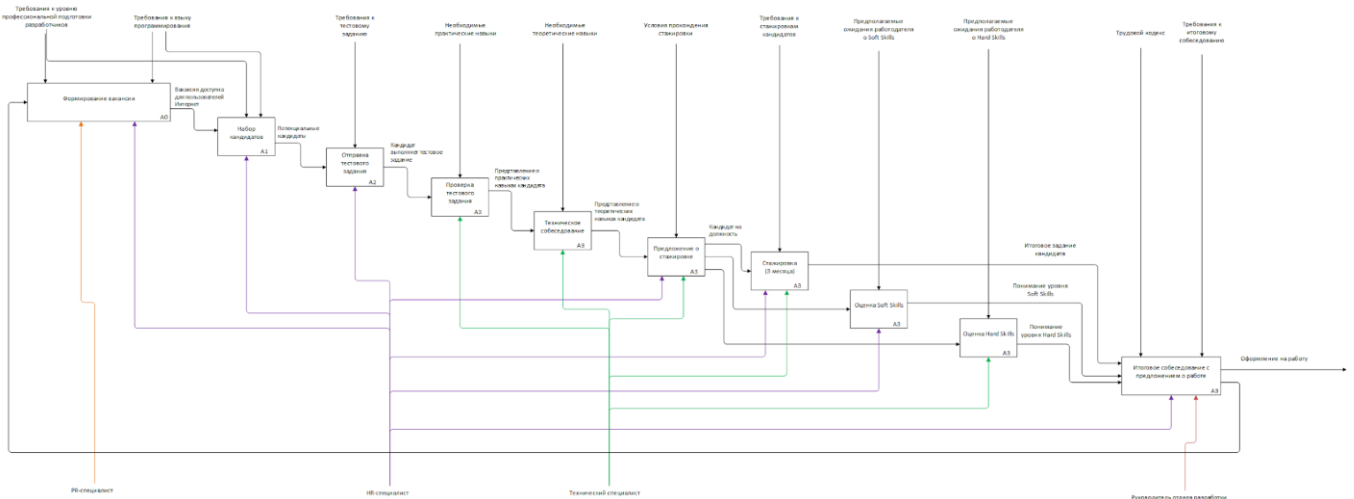


Рисунок 2. Диаграмма IDEF0 для метода набора сотрудников путем стажировок

Оригинальным решением для привлечения новых кадров является формат хакатона. Это эффективный HR-инструмент, так как он позволяет искать новые кадры без больших затрат. Хакатоны постепенно становятся площадкой для взаимодействия бизнеса с целевой аудиторией, проверки гипотез, получения отзывов о своих сервисах, что является возможностью снизить дефицит кадров, обнаружить новые таланты и оценить их с профессиональной стороны. При этом подразумевается, что бизнесу нужны от хакатонов в первую очередь свежие технологические идеи [1].

Стоит добавить, что не только работодатели наблюдают на хакатонах за потенциальными сотрудниками, но и наоборот. Разработчики оценивают сложившуюся на мероприятии ситуацию, пытаются в неформальной обстановке почувствовать общую корпоративную атмосферу, увидеть, как компания относится к сотрудникам.

Человек во время проведения хакатона проявляет свои профессиональные качества в формате «дали задачу – выполнил». Как показывают некоторые исследования, такой способ в разы эффективнее, чем дистанционное тестовое задание или беседа.

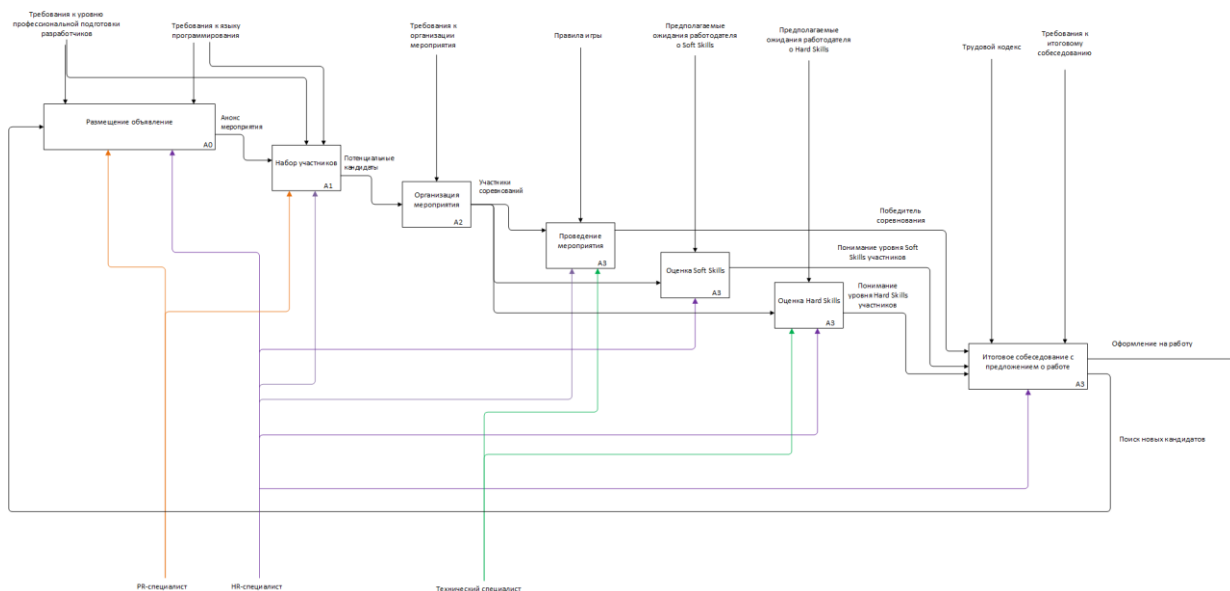


Рисунок 3. Диаграмма IDEF0 для метода набора сотрудников путем проведения хакатонов

Исходя из вышеперечисленного, преимущества хакатона перед альтернативными средствами поиска новых сотрудников это:

1. короткий промежуток времени для оценки качества сотрудника;
2. оценка hard- и soft skills;
3. наличие сразу нескольких потенциальных кандидатов;
4. интересный и соревновательный повод для встречи участников;
5. отсутствие лишнего стресса перед собеседованием у потенциального кандидата;
6. повышение уровня лояльности по отношению к компании.

1.2 Постановка целей

Данный проект будет основываться на задачах, которые стоят перед HR-специалистами в процессе поисков сотрудников:

- привлечение сторонних разработчиков в компанию;
- создание нового способа построить доверительную коммуникацию сотрудников HR-службы с потенциальными сотрудниками;
- улучшение компетенций разработчиков для сотрудников компании.

Целью выпускной квалификационной работы является проектирование и разработка программного комплекса, предназначенного для проведения технических соревнований. Участники такого мероприятия должны разработать клиент-бота, который будет реализовывать алгоритм для игры.

Глава 2. Проектирование программного комплекса

2.1 Разработка и анализ требований

2.1.1 Функциональное назначение

Программный комплекс предоставляет возможность участнику соревнований подключаться к игровому серверу и принимать участие в техническом чемпионате с возможностью визуализации результатов.

2.1.2 Эксплуатационное назначение

Программа является способом демонстрации итогов технических соревнований для разработчиков. Каждый пользователь данной программы может зарегистрироваться в системе и принять участие в чемпионате.

2.1.3 Нефункциональные требования

1. Серверная часть будет реализована с помощью C# .Net Core 3.0.
2. Frontend-часть должна быть разработана с использованием JavaScript-библиотеки React, Redux.
3. Взаимодействие между ботом и серверной частью будет происходить по протоколу SignalR.
4. Должны быть разработаны шаблоны (boilerplates) для таких языков программирования, как C#, Javascript, Java, Python. Шаблон представляет собой проект, который реализует SignalR-коммуникацию между ботом-клиентом и игровым сервером.

2.1.4 Требования к внутренней коммуникации

В качестве протокола взаимодействия между клиентской и серверной частью используются API-концентраторы SignalR ASP.NET, поскольку они позволяют выстроить событийно-ориентированный обмен сообщениями.

Вся коммуникация происходит асинхронно.

2.1.5 Требования к функциональным характеристикам

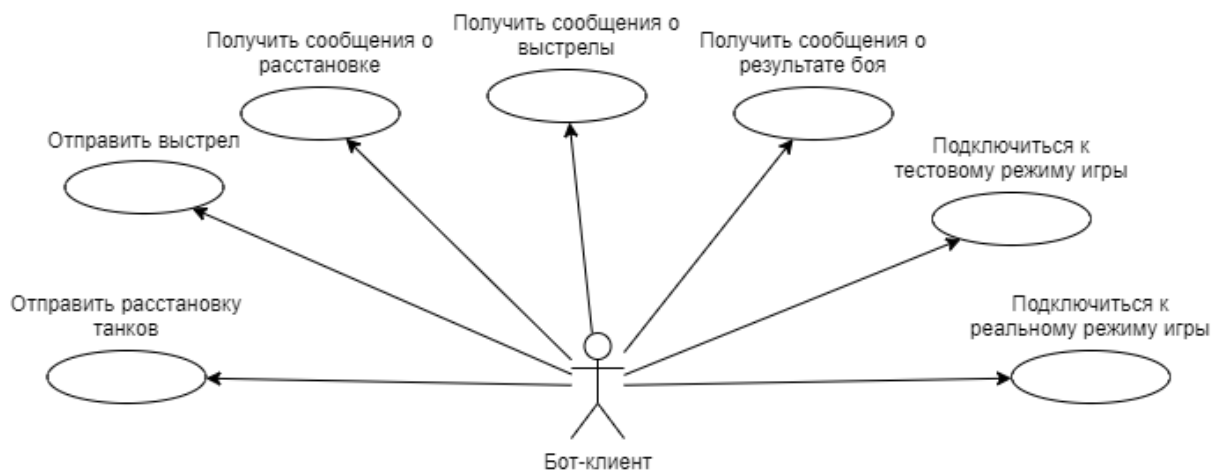


Рисунок 4. Диаграмма вариантов использования для бота-клиента

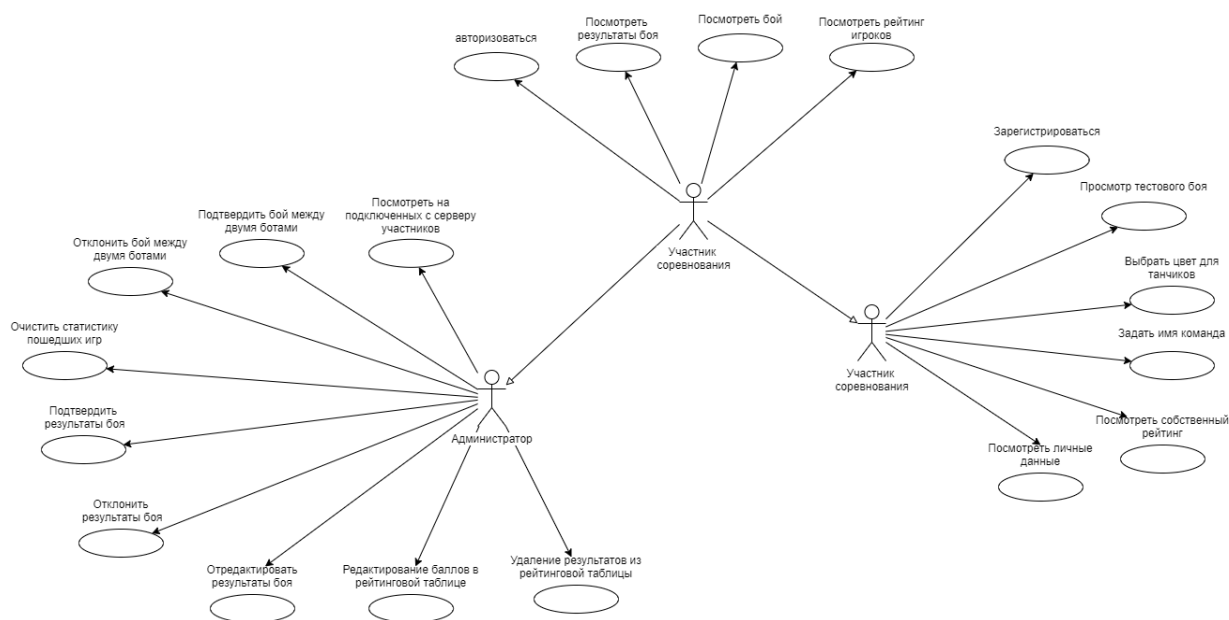


Рисунок 5. Диаграмма вариантов использования для участников

1. Визуализация текущего боя между двумя участниками (что идёт на «на вход», что показывается «на выходе»).
2. Клиентская часть должна отображать победителя сразу после отображения хода всего боя.
3. Клиентская часть должна отображать рейтинговую таблицу баллов игроков через 10 секунд после отображения победителя боя.
4. Возможности бота-клиента:

- a. подключиться к игровому серверу через SignalR;
- b. отправлять расстановку серверу через SignalR;
- c. отправлять выстрел серверу через SignalR;
- d. получать сообщения об (не)успешном подключении к игре с помощью подписки на метод сервера, который уведомит о новых сообщениях;
- e. получать сообщения от сервера об (не)успешной расстановке в ответ на отправленную расстановку танков;
- f. получать сообщения о результате каждого выстрела после осуществления (отправки на сервер) выстрела.

5. Требования к компоненту управления

Для администратора системы предусмотрен отдельный личный кабинет, вход в который осуществляется через логин и пароль.

Возможности компонента управления:

- a. подтвердить бой двух подключенных участников;
- b. отклонить бой при одном или двух подключенных участниках;
- c. очистить рейтинги игроков за прошедшие бои;
- d. подтвердить результаты боя; после каждого боя администратор получает уведомления об итогах последней осуществленной игры с возможностью:
 - i. подтвердить результаты боя;
 - ii. отклонить результаты боя;
 - iii. редактировать результат боя.
- e. редактировать баллы игроков в рейтинговой таблице;
- f. удалить игроков из рейтинговой таблицы.

6. Компонент личного кабинета (каждый зарегистрированный в системе пользователь имеет личный кабинет) включает следующие возможности:

- a. вход в личный кабинет осуществляется через логин и пароль;
- b. регистрация участника осуществляется через логин и пароль;

- c. наличие возможности восстановления пароля через электронную почту;
- d. наличие возможности смены пароля;
- e. наличие возможности выбрать цвет для танка при визуализации боя;
- f. возможность задать имя игрока/команды;
- g. просмотр личных данных (имя пользователя, название команды, цвет танков);
- h. просмотр визуализации тестового боя собственного бота;
- i. просмотр рейтинга игр всех игроков.

7. Поведение системы при заиклировании алгоритма

- a. Если один из алгоритмов участника попадает в одну и ту же ячейку n -раз, то при каждом результате “мимо” ход передается его сопернику. Шансы на победу у такого алгоритма уменьшаются пропорционально количеству оставшихся шагов из 100. При достижении шага 100 для каждого из соперников игра завершается с подсчетом баллов.
- b. Если оба алгоритма двух соревнующихся участников бьют в одну и ту же ячейку n -раз, то на каждом шаге “мимо” ход передается его сопернику. Количество шагов уменьшается, по достижению лимита в 100 шагов игра завершается с подсчетом баллов.

2.1.6 Правила игры

1. Игра ведется на поле размером 10x10 клеток.
2. В одной игре принимают участие два игрока.
3. В режиме отладки одновременно может происходить любое количество игр, но один игрок всегда может играть только в одну

игру. В тестовом или реальном режиме одновременно происходит только одна игра, которая отображается на экране.

4. Каждому игроку дается 10 танков.
5. Танки представляют собой объекты длиной 2 смежные клетки, т.е. танк обязательно имеет «линейную» форму.
6. Танки не могут касаться друг друга, в том числе по диагонали.
7. Игра делится на фазы: размещение танков, бой, подведение итогов.
8. В фазе размещения танков необходимо указать координаты всех клеток, занятых танками.
9. В игре есть три режима: боевой (с таймером), тестовый (с таймером) и режим отладки (без таймера). В режимах с ограничением времени на расстановку дается 3 секунды. Сервер считает, что размещение задано после получения матрицы 10x10 с расстановкой. Если сервер сочтет, что расстановка противоречит правилам, то танки будут размещены случайно. Это сделано для исключения случая, когда алгоритм всегда будет расставлять танки неправильно, и игра не сможет перейти к следующей фазе.
10. Закончив расстановку танков, игроки начинают «обстрел» друг друга. Игрок, делающий первый ход, выбирается сервером.
11. В фазе выстрелов игроки получают команду от сервера, на которую они отвечают координатами ячейки, в которую они хотели бы «выстрелить». Каждому игроку дается 100 ходов. Сервер сообщает результаты.
12. Если выстрел попал в «подбитый» танк, то он считается уничтоженным. Игрок может продолжать свой ход.
13. Если выстрел не попал в танк, ход передается другому игроку.
14. Игра завершается, если у одного из игроков не осталось танков. Выигрывает тот игрок, у которого они остались.

15. Игра завершается при достижении хода 100 для каждого из участников. Победителем становится игрок с наибольшим количеством «нетронутых» клеток с танками. Если клеток поровну, результатом игры является ничья.
16. Игра завершается, если один из игроков превысил таймаут на ход (3 сек, для режима игры «Бой» и «Тест»). В этом случае победителем становится другой игрок.

Типы исходов игры:

- Победа. Игра окончилась уничтожением всех танков одного из игроков или окончанием ходов у обоих игроков. В обоих случаях выигрывает игрок с большим количеством не уничтоженных противником ячеек танков.
- Техническая победа. Игра окончилась по таймауту, один из игроков не успел произвести выстрел в отведенное время. Этот игрок становится проигравшим.
- Ничья. Игра окончилась, у обоих игроков не осталось больше ходов при одинаковом количестве не уничтоженных противником ячеек танков.

2.2 Выбор технологий разработки

2.2.1 Single-page Application (SPA)

Традиционно браузер получает HTML-код с сервера и отображает его. Когда пользователь переходит на другой URL-адрес, требуется полное обновление страницы, и сервер отправляет новый HTML-код для новой страницы. Это называется рендерингом на стороне сервера.

Однако в современных SPA вместо этого используется рендеринг на стороне клиента. Браузер загружает начальную страницу с сервера вместе со скриптами (фреймворками, библиотеками, кодом приложения) и таблицами стилей, необходимыми для всего приложения. Когда пользователь переходит на другие страницы, обновление страницы не происходит. URL-адрес страницы обновляется с помощью API истории HTML5. Данные, необходимые для новой страницы, обычно в формате JSON, извлекаются браузером через запросы AJAX на сервер. Затем SPA динамически обновляет страницу данными (с помощью JavaScript), которые он уже получил при начальной загрузке страницы. Эта модель похожа на работу нативных мобильных приложений [2].

При таком подходе на сервер поступает меньше HTTP-запросов, поскольку одни и те же ресурсы не нужно загружать повторно для каждой загрузки страницы.

В то время как традиционные приложения на стороне сервера по-прежнему являются жизнеспособным вариантом, четкое разделение клиент-сервер лучше масштабируется для более крупных инженерных групп, поскольку код клиента и сервера можно разрабатывать и выпускать независимо. Это особенно актуально в Grab, когда у нас несколько клиентских приложений, работающих на одном сервере API.

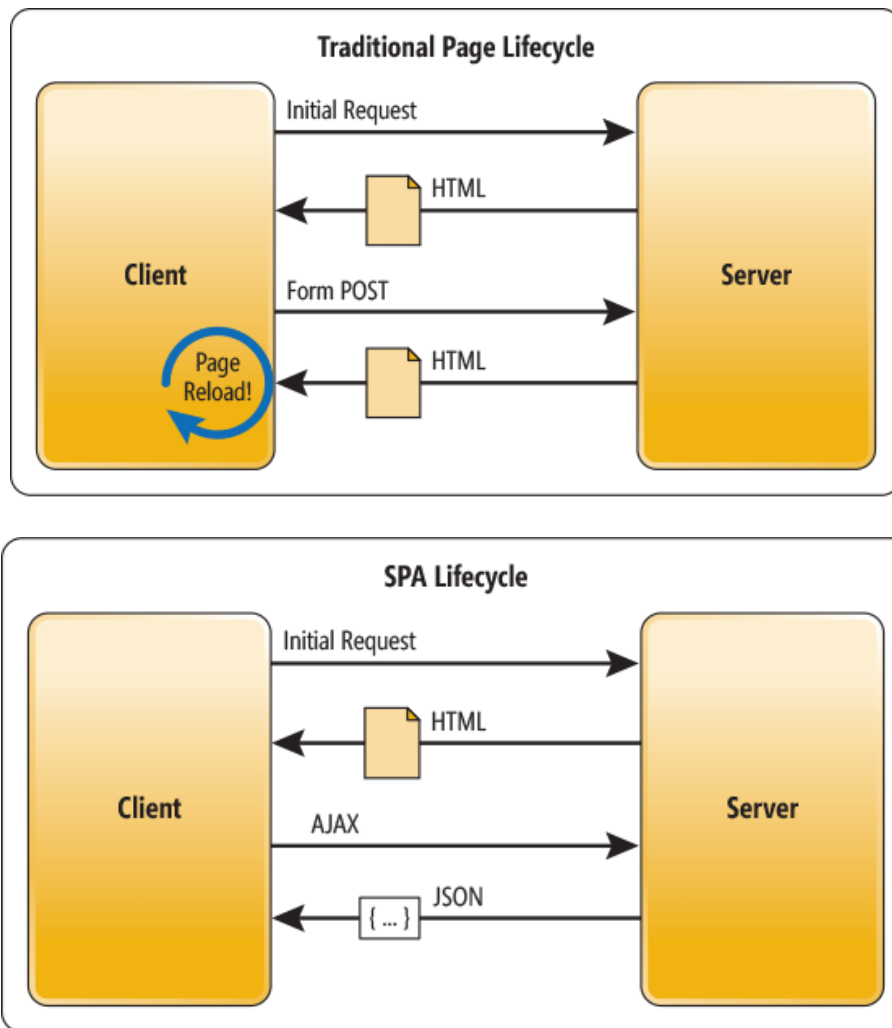


Рисунок 6. Принцип функционирования SPA

2.2.2 React

React JS – это библиотека JavaScript, используемая в веб-разработке для создания интерактивных элементов на веб-сайтах. React считается одной из самых популярных технологий для работы с JavaScript.

Основные преимущества использования React:

1. упрощение общего процесса написания компонентов;
2. повышение производительности и оптимизация дальнейшего обслуживания;
3. обеспечение более быстрого рендеринга.

Часто обновления программы могут являться проблемой, потому что приложение имеет сложную логику, и изменения в одном компоненте могут повлиять на другие. Чтобы решить эту проблему, Facebook дополнил React возможностью повторного использования компонентов системы, и разработчики определяют его как одну из лучших функций React JS.

При создании приложения с высокой нагрузкой важно учитывать, как структура повлияет на общую производительность приложения. Даже новейшие платформы и движки не могут обеспечить отсутствие раздражающих моментов, так как DOM (объектная модель документа) имеет древовидную структуру, и даже небольшие изменения в верхнем уровне могут вызвать значительную смену интерфейса. Чтобы решить эту проблему, команда разработчиков Facebook представила Virtual DOM – в настоящее время одно из преимуществ использования React для высоконагруженных и динамических программных решений.

Рассматривая плюсы и минусы React, его можно легко обобщить в нескольких словах: популярный, удобный и многофункциональный. Основная идея React – «создавать крупномасштабные приложения с данными, которые неоднократно меняются со временем». Он предоставляет разработчикам возможность работать с виртуальным браузером (DOM), который намного быстрее и удобнее, чем реальный. Кроме того, он предлагает более простое создание интерактивных интерфейсов, поддержку JSX, компонентную структуру, и многое другое. Сочетание вышеупомянутых факторов делает его разумным выбором как для стартапов, так и для промышленных решений [3].

2.2.3 Redux

Redux – это инструмент управления состоянием внутри клиентской части веб-приложения. Он занимает мало места при установке (2 КБ, включая зависимости), поэтому практически не влияет на увеличение размера ресурсов приложения.

Redux используется в основном для управления состоянием приложений. Фактически, он поддерживает состояние всего приложения в одном неизменяемом дереве состояний (объекте), которое нельзя изменить напрямую. Когда что-то меняется, создается новый объект

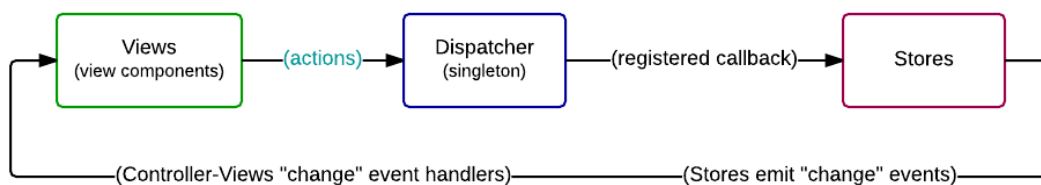


Рисунок 7. Схема Redux

По мере роста приложения можно заметить, что его структура становится всё более запутанной и сложной. Компоненты во всем приложении, возможно, должны совместно использовать и отображать общие данные, однако в самом React нет эффективного способа справиться с этим. Чтобы решить эту проблему, Facebook изобрела Flux, архитектуру приложения, которая дополняет компоненты составного представления React, используя однонаправленный поток данных [4].

Таким образом, шаблон потока имеет следующие характеристики:

- однонаправленный поток данных – делает приложение более предсказуемым по поведению, поскольку обновления можно легко отслеживать;
- разделение проблем – каждая часть в архитектуре Flux имеет четкие обязанности и сильно отделена от других;
- хорошо работает с декларативным программированием – хранилище может отправлять обновления в представление без указания способа перехода между представлениями.

2.2.4 SignalR

В последнее время в сети Интернет широко представлены интерактивные сервисы. Во многих веб-приложениях необходим интерактивный интерфейс, который постоянно изменяется. В связи с постоянным обновлением контента встаёт вопрос о синхронном постоянном обновлении страницы и способах сделать это как можно эффективнее при наименьших затратах ресурсов, а также без излишних и принудительных перезагрузок. Отличным способом для создания динамического контента является технология «SignalR».

Изначально самым распространённым методом являлся подход AJAX, когда клиентская часть с определенной заданной периодичностью посылает запросы на сервер, и сервер отвечает только в том случае, если изменения произошли (тогда клиентская часть обновится). Однако в больших приложениях использование метода AJAX приводит к большой нагрузке на сервер из-за большого количества запросов [5].

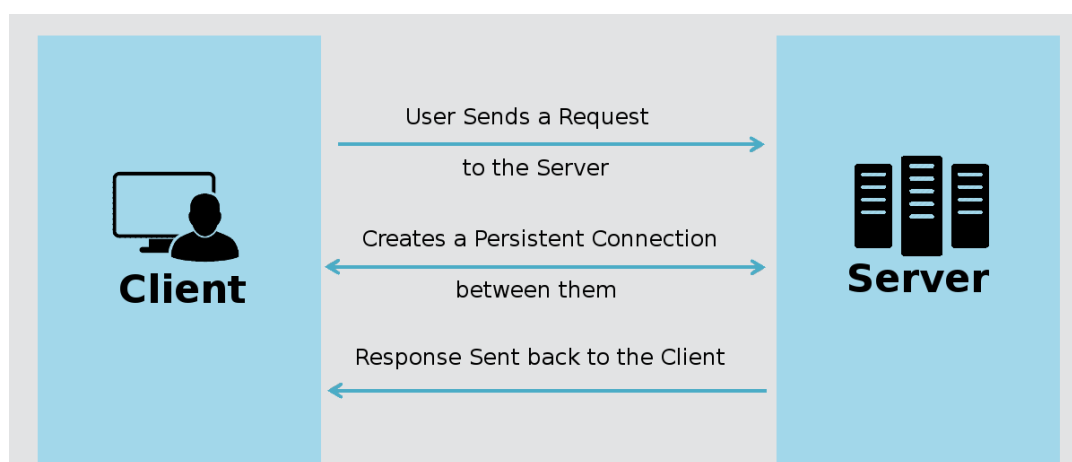


Рисунок 8. Принцип работы SignalR

Актуальным способом решения вышеуказанной проблемы является использование библиотеки «SignalR». «SignalR» – это многофункциональная высокоуровневая библиотека, которая обеспечивает простой способ передачи данных между клиентом и сервером. «SignalR»

управляет соединениями автоматически и позволяет отправлять данные, используя широковещательные или одноадресные передачи.

Клиент «подписывается» на серверный метод, который в случае изменения внутреннего состояния оповестит об этом клиента. Таким образом, клиенту не требуется постоянно отсылать запросы на сервер, чтобы узнать, изменилось ли внутреннее состояние серверной части веб-приложения, сервер сам оповещает клиента о новых данных.

С помощью библиотеки «SignalR» сервер может вызывать методы JavaScript на всех клиентах самостоятельно, при необходимости обновления. Клиент JavaScript также может вызывать методы сервера. В случае возможности использования WebSocket, работа будет происходить через WebSocket, если такой возможности нет, то будет осуществлён поиск другого приемлемого способа.

Библиотеку «SignalR» можно использовать в приложениях, в которых пользователю необходимо обновлять веб-страницу, чтобы получить актуальные данные. Обычно такими приложениями являются веб-панели мониторинга, в которых данные должны постоянно самостоятельно обновляться (без ожидания обновления страницы пользователем) [6].

Преимущества библиотеки «SignalR» можно использовать для разработки следующих приложений:

- онлайн-чаты;
- совместные приложения (например, многопользовательские дашбоарды или инструменты управления проектами);
- онлайн-игры, автоматическое обновление приложений в реальном времени (таких как информационные панели или карты);
- уведомления и оповещения.

2.2.5 TypeScript

TypeScript – это язык с открытым исходным кодом, основанный на JavaScript, путем добавления определений статических типов.

Статическая типизация дает множество преимуществ при написании приложений. Такой подход может помочь быстро обнаружить типичные ошибки в ходе написания. Типы также служат формой документации для кода и улучшают его читаемость.

Преимущества использования TypeScript:

1. Код становится легче для понимания. Обычно, когда разработчики работают над фрагментом кода (например, кодом функции), для полного его осмысления нужно понять следующее:

- какие аргументы принимает функция?
- какое значение возвращает функция?
- какие внешние данные нужны?

TypeScript позволяет указывать типы, тем самым упрощая разработку и понимание программного кода.

2. Код становится проще и быстрее для реализации.
3. Код становится проще для дальнейшего рефакторинга.
4. Меньшая вероятность возникновения ошибок и более быстрое их устранение [7].

2.2.6 Styled Components

Styled Components – это способ стилизации компонентов React с использованием CSS и преимуществ, предлагаемых ECMAScript 6. Styled Components является одним из новых способов использования CSS в современном JavaScript. Он предназначен для того, чтобы стать преемником

модулей CSS, способ написания CSS которых ограничен одним компонентом, а не распространяется к другим элементам на странице.

Существует общий и самый старый метод использования стилей на уровне документов – создание файла `style.css` и ссылки на него в файле HTML, но в последнее время широко используется метод интеграции стилей в JavaScript. Этот метод известен как CSS-in-JS [8].

CSS-in-JS – это стилевой подход, который абстрагирует модель CSS до уровня компонента, а не до уровня документа. Идея в том, что CSS может быть ограничен только определенным компонентом в приложении, а не уровнем документа.

Глава 3. Разработка программного модуля

3.1 Общая архитектура системы

Ниже представлена диаграмма компонентов системы. Игровой сервер обменивается данными с остальными частями модуля с помощью SignalR.

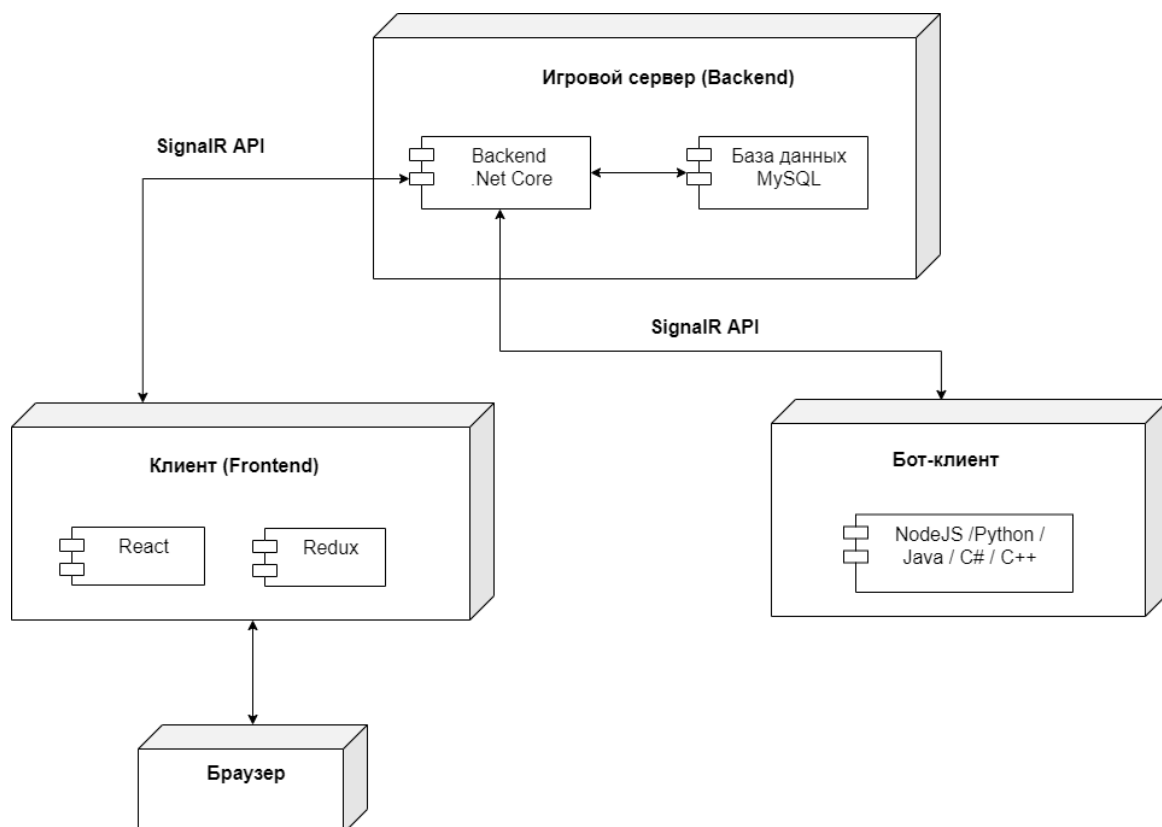


Рисунок 9. Диаграмма компонентов системы

Таблица 1 – Описание элементов схемы

№	Элементы схемы	Описание
Серверная часть		
1	БД	БД для хранения статистики по играм и участникам.
2	Backend .Net Core	Игровой сервер, реализующий логику работы игры.

3	SignalR API	Средства обмена данными между сервером и ботом-клиентом, а также между серверной и клиентской частью приложения.
Клиентская часть		
4	Бот-клиент	Программа-клиент, разработанная участником для осуществления обращений к серверу игры в целях выиграть игру по заданному в ней алгоритму.
5	Frontend с визуальной частью игрового поля	Клиентская часть приложения для визуализации игр бот-клиентов.

3.2 Внутренняя SignalR-коммуникация

Для реализации коммуникации между клиентами и игровым сервером в данном проекте используются два SignalR-концентратора, в терминологии SignalR они называются «Hub». В данном случае «Hub» представляет собой класс, который реализует все необходимые методы, относящиеся к данному концентратору [9].

Первый концентратор («GameHub») предназначен непосредственно для игровой логики – общения между ботами-клиентами с сервером, а также frontend-части с сервером во время боя.

Второй концентратор («ManagementHub») предназначен в основном для действий администратора: отслеживания подключенных игроков и управления боями.

3.2.1 Обмен данными сервера с ботом

Ниже будет описана последовательность взаимодействия с методами SignalR для бота-клиента.

1. `connection.invoke(gameType, userId)`

Бот при старте подключения вызывает метод на сервере для того, чтобы задать тип игры (`gameType`), который может быть «Debug», «Fight» и «Test», а также передать ID пользователя, который был получен при регистрации в личном кабинете участника. Данный метод является начальным и служит серверу сигналом о том, что бот подключен.

2. `connection.on("requestArrangement")`

Бот «подписывается» на метод «requestArrangement», который будет уведомлять о корректности или некорректности отправленной расстановки танков на поле.

3. `connection.invoke("ReceiveArrangement", board)`

Затем бот вызывает серверный метод «ReceiveArrangement» для отправки расстановки танков на поле.

4. `connection.on("requestStep")`

После того, как бот получил уведомление о своей расстановке, вызывается серверный метод, который будет уведомлять о результатах каждого выстрела.

5. `connection.invoke("ReceiveStep", x, y)`

Для отправки выстрела циклично вызывается серверный метод «ReceiveStep», в который передаются координаты точки, куда будет производиться выстрел.

6. connection.on("receiveMessage")

Последним заключительным действием является «подписывание» бота на метод «receiveMessage», который отправляет уведомления в случае определенных неполадок, некорректности подключения (например, если в лобби уже подключены 2 бота, и пытается подключиться третий), а также этот метод получает уведомление о результате прошедшего боя.

Ниже представлена наглядная диаграмма взаимодействия бота и сервера.

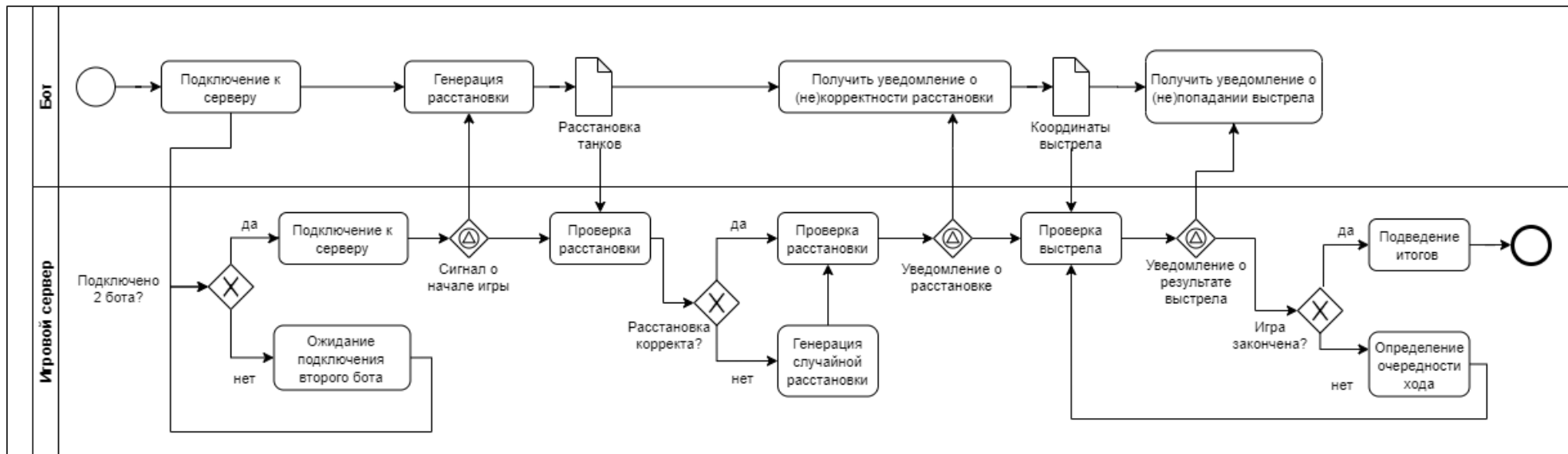


Рисунок 10. Диаграмма BPMN коммуникации бота-клиента с сервером

3.2.2 Обмен данными сервера с frontend-частью

Для frontend-части существует 2 вида SignalR-подключения: для отслеживания подключения ботов к серверу, и для демонстрации боев.

Для демонстрации боя и отображения победителя frontend-клиент использует концентратор «GameHub», а для отслеживания управления подключениями – «ManagementHub».

При запуске frontend-клиент вызывает метод «Watch» с передачей ключа. Данный метод случит идентификатором для сервера, чтобы показать, что подключенный клиент является frontend-клиентом, которому можно передавать данные о бое.

Затем клиент «подписывается» на метод, который будет присылать данные для демонстрации (расстановка двух ботов, хронология выстрелов, определение победителя). Как только боты завершат свой бой, сервер передаст данные в текущий метод. После демонстрации будет вызван метод сервера, который отправит данные о рейтинге участников.

Также frontend-клиент изначально «подписывается» на метод «Control». Как только один или два бота подключатся к серверу, сервер отправит данные этих ботов на frontend-часть при помощи метода «Control». Администратора может смотреть, кто подключен к игровому серверу, и в итоге отклонить или подтвердить начало боя. Для отклонения боя вызывается метод «DeclineFightGame», вследствие чего подключенные боты (или один бот) будут отключены от сервера. При подтверждении игры вызывается серверный метод «AcceptFightGame», и начинается бой между подключенными ботами. Функционал «отклонить/принять бой» нужен для того, чтобы администратор мог отслеживать подключения и не допускать случайных или нежелательных подключений ботов к серверу.

3.2.3 Проблема в организации коммуникации

Во время тестового мероприятия внутри компании обнаружилась возможность для участников соревнования «перехватить» расстановку танков противника через метод «Watch», которая отправляется сервером для frontend-части приложения. В результате, это могло привести к мошенничеству во время соревнований, так как стало возможно узнать расстановку противника до того, как игра завершится. Это сильно противоречило всей концепции технического соревнования, в котором участники должны написать эффективный алгоритм для игры. Возникла существенная уязвимость во всем проекте.

Исходя из этого, было принято решение изменить реализацию коммуникации frontend-части и сервером следующим образом:

- Изначально боты подключаются к серверу и проигрывают полностью весь бой. Боты получают ответы по выстрелам («попал/мимо»), но не получают данных о победителе или проигрыше в конце боя. Таким образом, бот фактически не знает, победил он или проиграл.
- После того, как боты закончили свой бой, сервер передает frontend-части веб-приложения все данные вместе с расстановками игроков, хронологией выстрелов, и конечным победителем. Frontend-часть принимает данные и с установленным таймаутом отображает их на интерфейсе пользователя.

При таком подходе визуализация игрового процесса начинается только тогда, когда фактически бой уже завершен, поэтому «перехватить» данные во время игры становится невозможным.

3.3 Интерфейс программного модуля

Интерфейс программного модуля состоит из трех частей:

1. интерфейс для администратора программной системы;
2. интерфейс личного кабинета участника;
3. интерфейс для проведения демонстрации боя.

3.3.1 Личный кабинет участника

Каждый участник соревнования должен зарегистрироваться в системе. После регистрации каждому участнику выдается личный кабинет (рис. 11).

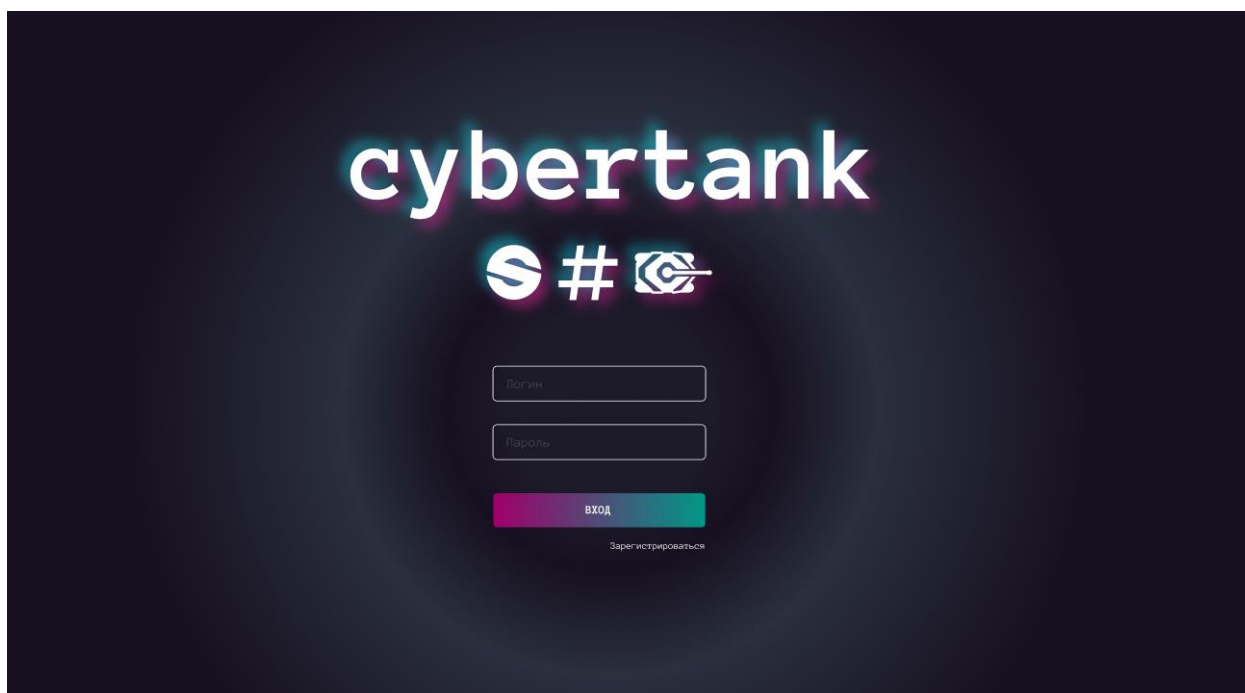


Рисунок 11. Форма авторизации

В личном кабинете имеются контактные данные пользователя, его результаты игр, а также ID, который потребуется для подключения бота к серверу (рис. 12).

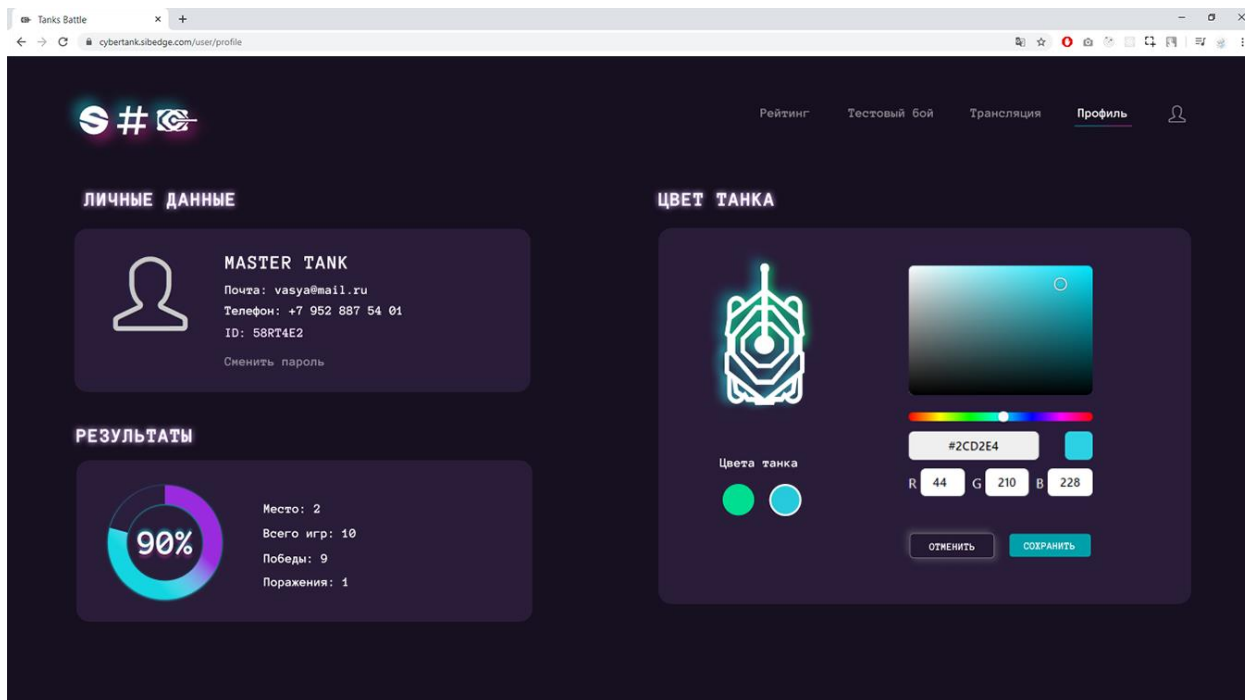


Рисунок 12. Профиль участника

Также в личном кабинете участник может задать цвета своему танку. Пользователь может выбрать два цвета, которые будут формировать цветовую гамму для игрового танка на поле, а также цвет выстрела (точки) на поле противника.

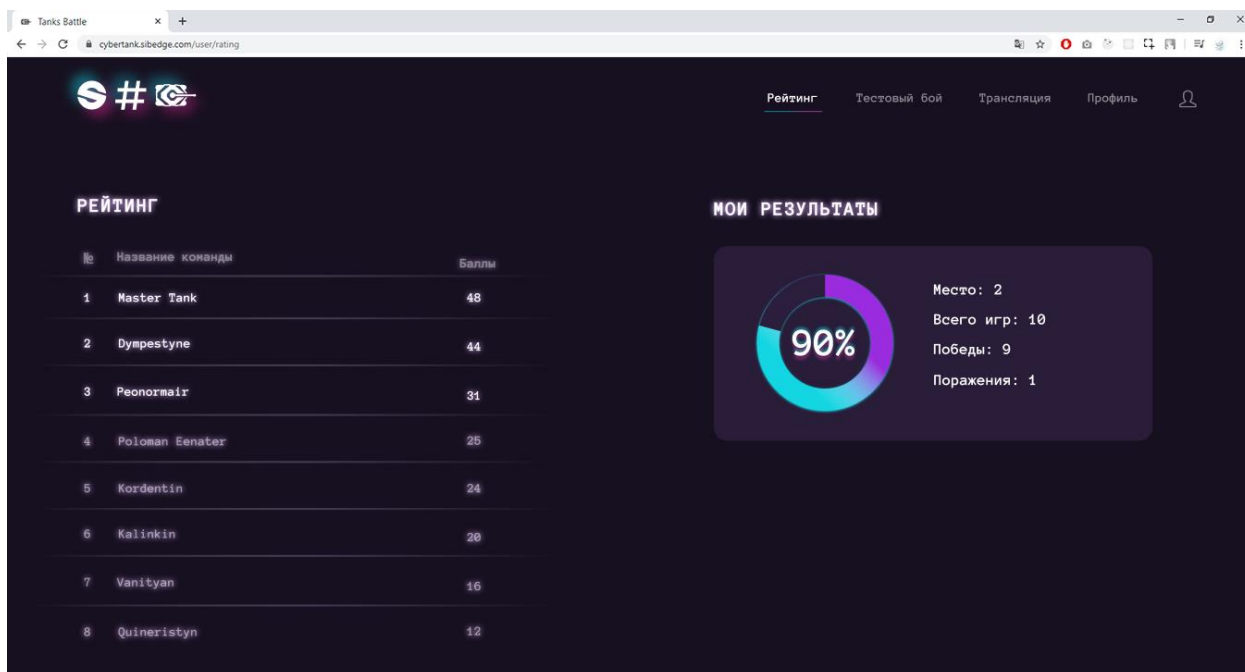


Рисунок 13. Рейтинг результатов в личном кабинете участника

В разделе «Рейтинг» (рис. 13) участник может посмотреть общий рейтинг игроков, а также собственную статистику игр и место в рейтинге.

В разделе «Тестовый бой» участник может увидеть визуализацию игры своего бота в тестовом режиме для отладки и проверки работоспособности алгоритма.

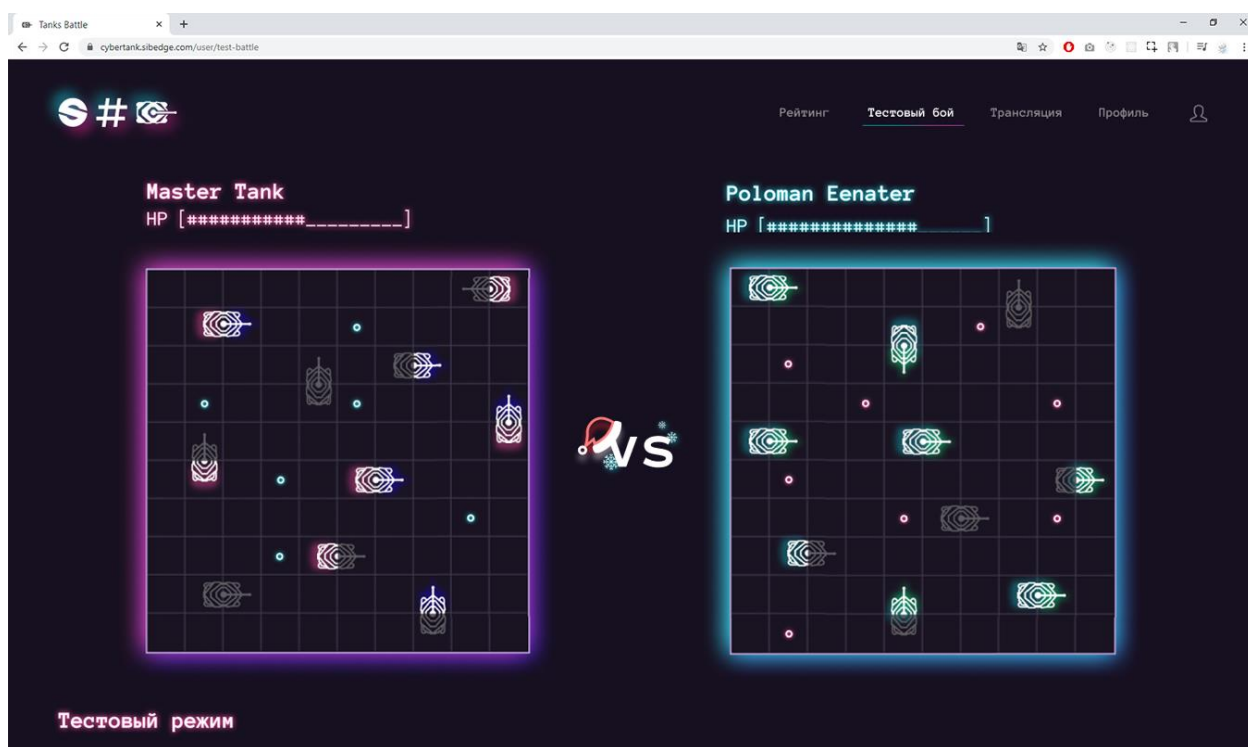


Рисунок 14. Просмотр тестового бота через личный кабинет участника

3.3.2 Личный кабинет администратора

Администратор системы имеет ограниченный функционал личного кабинета, который в основном используется для управления боями и их результатами, а также для управления подключениями ботов к игровому серверу.

В разделе «Управление боями» (рис. 15) администратор может посмотреть список всех прошедших игр, а также удалить или изменить их результаты.

№	Участники	Баллы	Начало боя	Оконание боя	
8	Master Tank – Collavann	0:3	14:12:02	14:25:55	
7	Master Tank – Dympestyle	0:0	13:41:14	14:00:00	
6	Peonormair – Mister Winner	0:3	13:28:01	13:35:32	
5	Vanityan – Kordentin	3:0	12:53:42	13:01:26	
4	Vanityan – Quineristyn	0:3	12:30:03	12:44:17	
3	Master Tank – Quineristyn	0:3	12:17:08	12:24:15	
2	Collavann – Quineristyn	0:3	11:51:55	12:01:44	
1	Dympestyle – Vanityan	0:3	11:11:06	11:25:02	

Рисунок 15. Управление боями в личном кабинете администратора

Для того чтобы изменить бой, необходимо выбрать победителя и тип выигрыша: победа, техническая победа или ничья (рис. 16). Баллы будут обновляться на стороне сервера, так как данные о начислении баллов за каждый тип выигрыша хранятся в БД.

Рисунок 16. Форма редактирования боя

Администратор имеет возможность в своем личном кабинете просмотреть весь список пользователей, которые зарегистрировались на текущий хакатон, вместе с их личными данными. Также имеется

возможность сбросить все статистику игр (рис. 17). Данный функционал нужен для того, чтобы очистить рейтинговую таблицу предыдущих хакатонов.

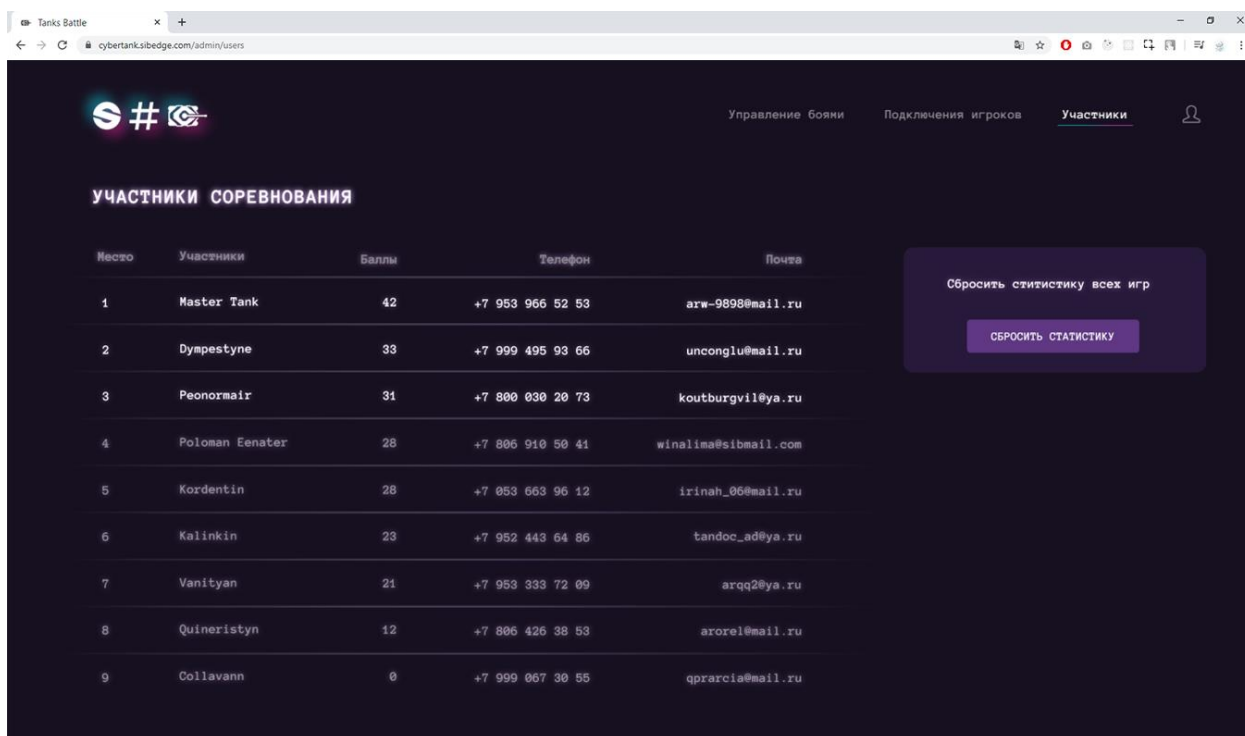


Рисунок 17. Список участников с полными данными в личном кабинете администратора

Основной возможностью администратора является способность отслеживать подключения к серверу, а также подтверждать или отклонять бой. Данная функциональность нужна для того, чтобы не допускать случайных или нежелательных подключений ботов-клиентов к игровому серверу.

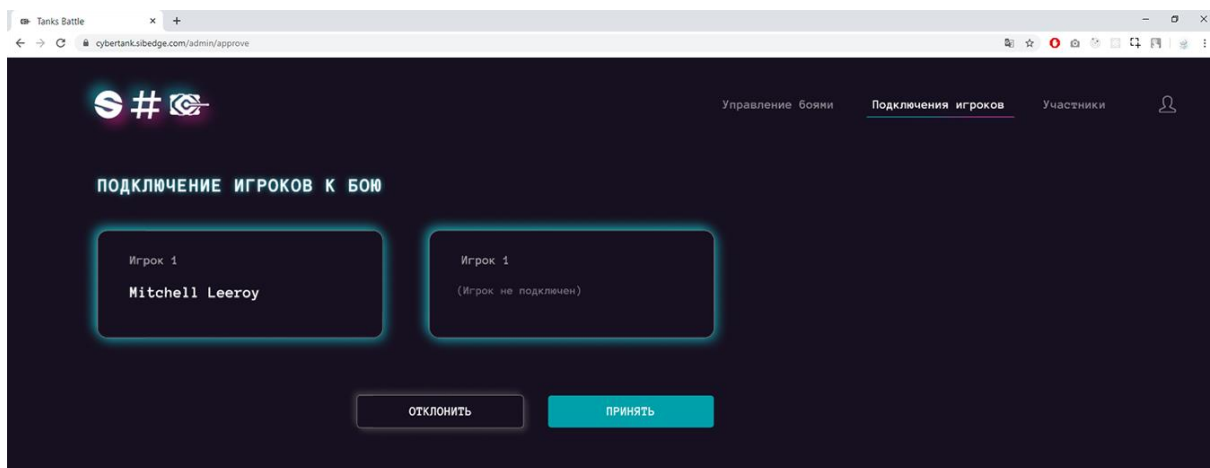


Рисунок 18. Форма отклонения (подтверждения) боя

3.3.3 Демонстрация на больших экранах

Для того чтобы проводить демонстрацию всех игровых боев перед участниками на большом экране, был разработан отдельный интерфейс, который включает в себя ограниченный функционал исключительно для просмотра боя и статистики игр.

Интерфейс для демонстрации на больших экранах имеет возможность отображать тестовый и основной бой. Для отображения боя frontend-часть воспроизводит все выстрелы в хронологическом порядке (рис. 19). Демонстрация на больших экранах отображает только основные бои, очки за которые будут занесены в рейтинговую таблицу.

Для того чтобы бот-клиент начал играть в основном режиме, необходимо запустить его при режиме игры «Fight». При режиме игры «Test» бой отображается в личном кабинете участника, и баллы за эти бои не учитываются, так как данный режим нужен лишь для отладки бота-клиента.

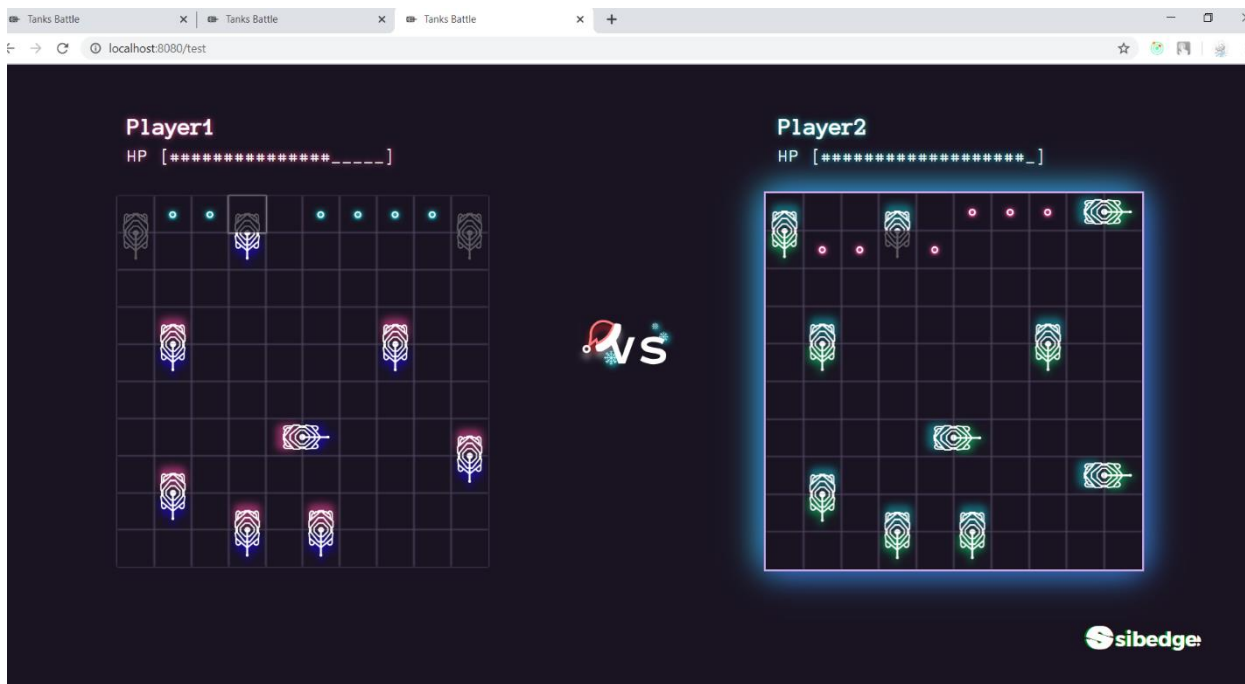


Рисунок 19. Экран отображения боя

После полной презентации хода боя отображается вся игра, демонстрация останавливается, и на экране отображается победитель (рис. 20).

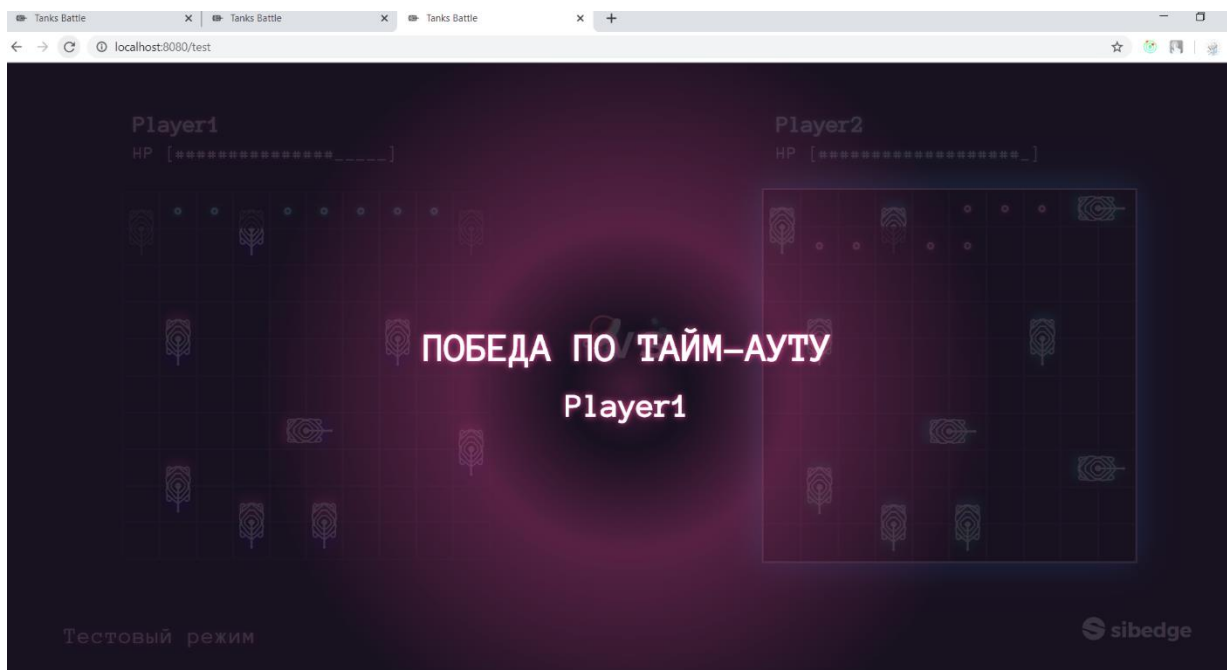


Рисунок 20. Экран отображения победителя

Через определённый период времени после отображения экрана с победителем отображается рейтинг игроков за весь чемпионат.

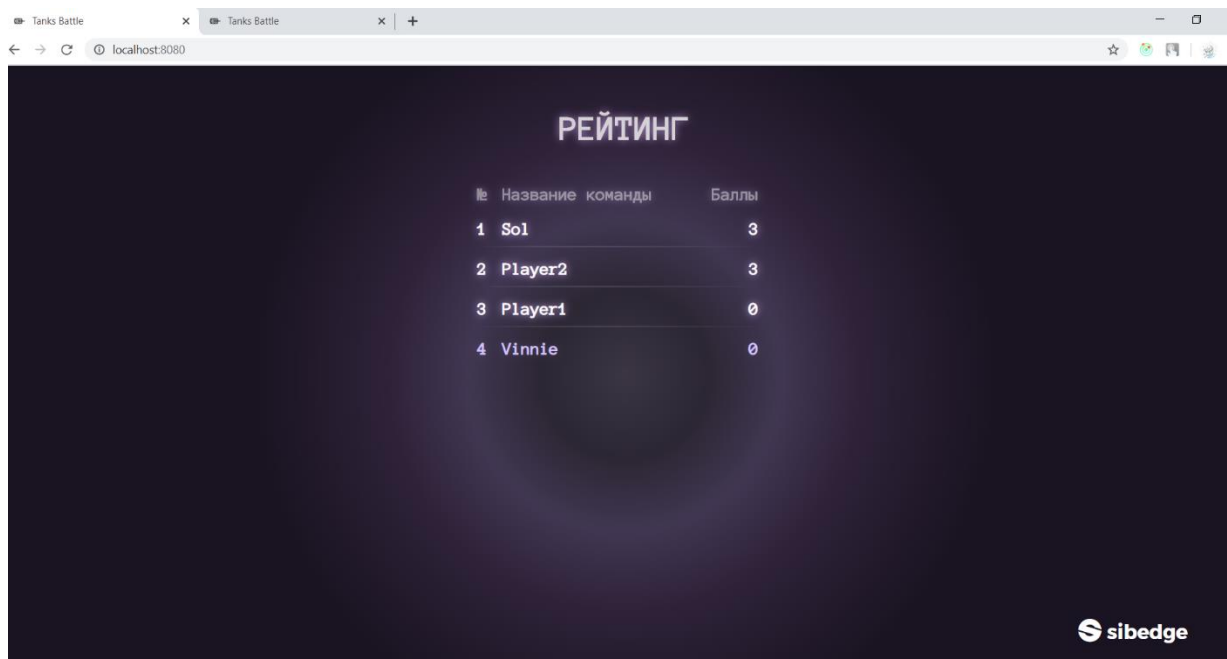


Рисунок 21. Экран отображения рейтинга участников

Экран рейтинга (рис. 21) будет отображаться до тех пор, пока не начнется следующий основной бой

3.4 Тестовый бот-клиент

Для того чтобы участники хакатона смогли протестировать работоспособность и эффективность своего написанного алгоритма (каждый участник должен был иметь возможность провести бой между своим ботом и тестовым ботом), было решено реализовать тестового бота-клиента.

3.4.1 Алгоритм расстановки тестового бота

В соответствии с исследованиями эффективности работы алгоритмов расстановки и выстрелов на примере игры «Морской бой» [10], можно судить о наиболее эффективных алгоритмах расстановки.

Алгоритм	95-й процентиль	85-й процентиль	15-й процентиль	5-й процентиль
По краям	81	76	60	55
Группировка сбоку	81	75	57	52
Случайно в одном направлении	77	71	54	49
Случайно в двух направлениях	75	70	54	49
Полностью случайно	75	70	54	49
Сгруппировать в центре	75	69	51	46

Рисунок 22. Математические расчёты результатов алгоритмов расстановки для игры «Морской бой»

Исходя из отображённых на рисунке 22 данных, можно сделать вывод о том, что алгоритм расстановки по краям является самым эффективным. Однако этот способ расстановки является одним из самых примитивных и очевидных. Скорее всего, человек, знакомый с теорией игры «Морской бой», знает данную особенность. Поэтому высока вероятность того, что противник будет учитывать это. В таком случае тестовый бот использует расстановку «способ группировки танков с одной

или нескольких сторон поля». Пример такой расстановки представлен на рисунке 23.

	А	Б	В	Г	Д	Е	Ж	З	И	К
1	■	■		■		■	■	■	■	46
2				■		■	■	■	■	
3	■	■					■	■	■	
4					■		■	■	■	
5	■		■		■		■	■	■	
6	■		■				■	■	■	
7						■	■	■	■	
8	■	■		■			■	■	■	
9				■			■	■	■	
10	■	■					■	■	■	

Рисунок 23. Пример расстановки с одного края поля

Расстановка делается с учетом того, чтобы было минимальное количество задействованных клеток с учетом обязательных отступов.

3.4.2 Алгоритм атаки тестового бота

Так как алгоритм расстановки по краям является одним из самых популярных, было решено реализовать проверку на то, сделал ли противник бота расстановку по краям поля. Тестовый алгоритм делает небольшую проверку с помощью определенного числа выстрелов. Если теория о том, что противник сделал расстановку по краям, не подтвердилась, алгоритм переходит к диагональному способу атаки.

Условия:

- количество крайних клеток = 36;
- количество клеток, занятых танками = $2 * 10 = 20$;
- количество клеток, занятых танками с учетом обязательных отступов = $(2 + 1) * 10 - 1 = 29$;
- $36 - 29 = 7$ – максимальное расстояние между двумя танками в клетках при наиболее плотной расстановке.

	А	Б	В	Г	Д	Е	Ж	З	И	К
1	1			2			3			4
2	1									
3	2									
4	3									5
5	4									
6	5									
7	6									6
8	7									
9	10									
10		1	9			8			7	

Рисунок 24. Максимальное расстояние между двумя танками при расстановке по краям поля

Для техники выстрелов алгоритм тестового бота делает максимально 3 выстрела по краям поля. Выбирается случайная точка с краю поля, и от нее начинаются выстрелы. Каждый выстрел делается с шагом 2, то есть через одну клетку последовательно в одном направлении.

Если по истечению трех выстрелов не найдено ни одного танка противника, алгоритм меняет стратегию атаки. Изучив статистические показатели алгоритмов выстрелов для игры «Морской бой», было выявлено, что диагональный алгоритм имеет преимущество перед остальными, особенно, если этот вид атаки используется против случайной расстановки.

На рисунке 25 можно посмотреть сравнительный анализ алгоритмов атаки для игры «Морской бой» [10].

Алгоритм	95-й процентиль	85-й процентиль	15-й процентиль	5-й процентиль
Диагональный	70	64	48	44
Случайный	76	71	54	49
Линейный	79	75	59	55
Квадратный	86	78	60	56

Рисунок 25. Математические расчёты результатов алгоритмов атаки для игры «Морской бой»

Принцип работы диагонального алгоритма атаки заключается в следующем: генерация выстрелов происходит вдоль главной диагонали поля. Поскольку у всех танков фиксированная длина, равная двум клеткам поля, то шаг между диагоналями будет равен 2, то есть берется для обстрела каждая вторая диагональ [11].

	А	Б	В	Г	Д	Е	Ж	З	И	К
1	21		13		7		3		1	
2		22		14		8		4		2
3	31		23		15		9		5	
4		32		24		16		10		6
5	39		33		25		17		11	
6		40		34		26		18		12
7	45		41		35		27		19	
8		46		42		36		28		20
9	49		47		43		37		29	
10		50		48		44		38		30

	А	Б	В	Г	Д	Е	Ж	З	И	К
1		2		6		12		20		30
2	1		5		11		19		29	
3		4		10		18		28		38
4	3		9		17		27		37	
5		8		16		26		36		44
6	7		15		25		35		43	
7		14		24		34		42		48
8	13		23		33		41		47	
9		22		32		40		46		50
10	21		31		39		45		49	

Рисунок 26. Пример работы алгоритма обстрела

При диагональной атаке не важно, с какой стороны поля следует начинать обстрел – алгоритм случайным образом выбирает точку на поле, соседнюю от угловой, и с нее начинает свои ходы.

3.4.3 Алгоритм уничтожения подбитого танка для тестового бота

Для алгоритма уничтожения подбитого танка используется следующая стратегия – когда подбита первая часть танка, алгоритм

высчитывает возможные расположения второй части танка с учетом максимального исключения клеток при вероятном расположении второй части танка в предполагаемой клетке.

На рисунке 27 можно посмотреть сравнительный анализ алгоритмов уничтожения для игры «Морской бой» [10].

Алгоритм	95-й процентиль	85-й процентиль	15-й процентиль	5-й процентиль
Крестообразный	23	21	15	13
Исключение макс.	21	19	14	12
Исключение мин.	24	22	15	13

Рисунок 27. Математические расчёты результатов алгоритмов добивания для игры «Морской бой»

Допустим, мы подбили танк в клетке Д3. Вторая часть танка может находиться в клетках Д2, Е3, Д4 и Г3. Если вторая часть танка находится в клетке Д2, то, учитывая, что между танками должно быть минимум одна клетка, и они не могут соприкасаться краями по диагонали, в таком случае будет открыто 10 клеток, при Е3 – 7, Д4 – 7, Г3 – 10. В данном случае претендентами на выстрел будет 2 клетки с максимальным числом исключения – Д2 и Г3.

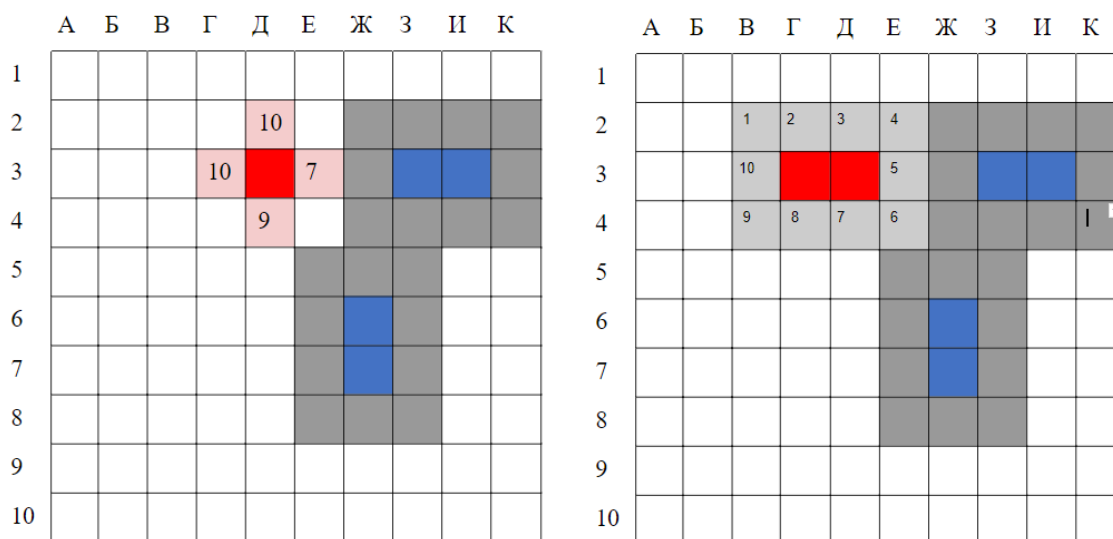


Рисунок 28. Уничтожение танка методом максимального исключения

Если имеется несколько одинаковых чисел максимального исключения клеток, то случайным образом выбирается одна из них.

Также важно отметить, что при попадании в танк, он обязательно должен быть уничтожен прежде, чем продолжить обстрел по другим мишеням.

Исходя из вышеописанного, алгоритм работает в такой последовательности:

1. генерирование расстановки с одного края поля;
2. проверка танков противника на расстановку «по краям»;
3. если теория о расстановке «по краям» подтвердилась, то обстрел производится по краю поля с определенным шагом, если же нет, то обстрел производится вдоль главной диагонали с шагом 2;
4. при попадании в противника, танк обязательно уничтожается методом максимального исключения.

3.4.4 Статистические результаты для тестового бота

Случайная расстановка подразумевает то, что танки могут располагаться в любой части поля и в любом порядке. Поэтому число клеток, где потенциально могут быть расположены танки – 100.

На рисунке 29 представлены результаты статистического анализа проведенных игр тестового бота в разрезе таких показателей, как число побед к числу совершенных шагов (ходов).



Рисунок 29. Статистические результаты тестового бота против алгоритма случайной расстановки

При расстановке танков параллельно и вплотную к каждому краю (чтобы две ячейки танка соприкасались с краем поля), количество клеток, где могут быть расположены танки, равно 36. Если же какие-то из танков соприкасаются только одной ячейкой с краем, то клеток, где потенциально могут быть расположены танки 64, поскольку танк состоит из двух клеток.

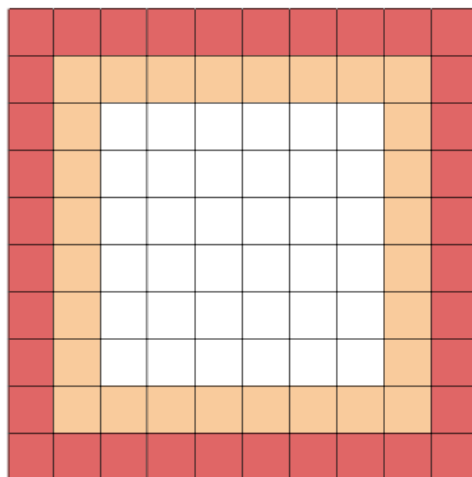


Рисунок 30. Расположение танков при расстановке по краям поля

Исходя из этого, распределение числа побед будет расположено на графике в интервале от 20 до 64.

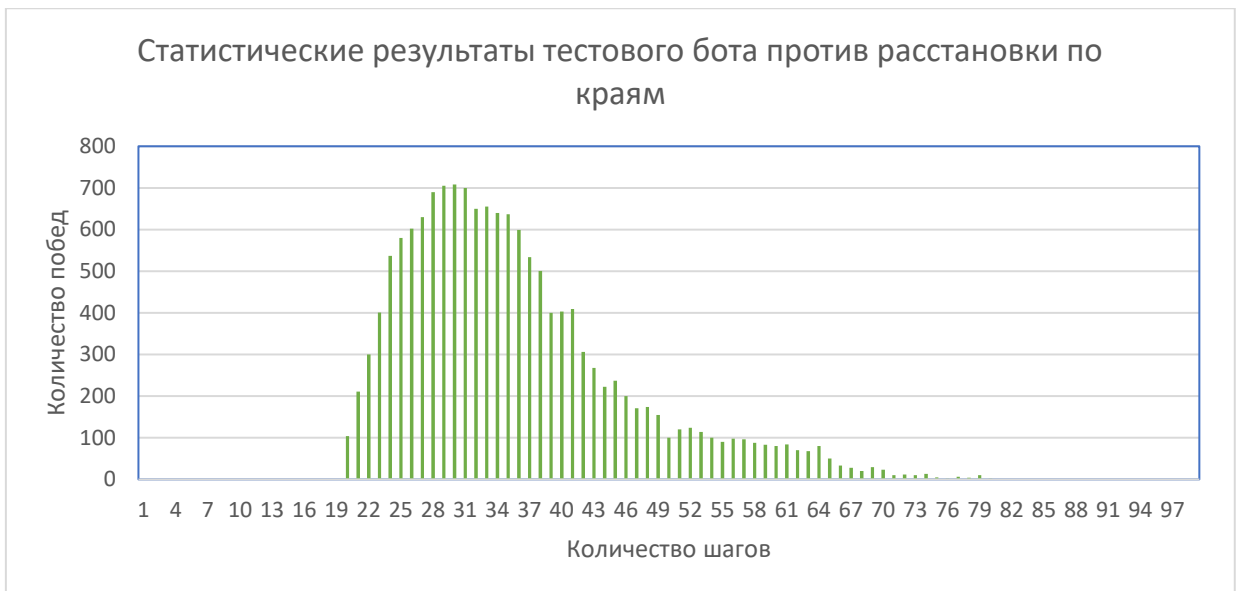


Рисунок 31. Статистические результаты тестового бота против расстановки по краям

Если алгоритму попадается противник с расстановкой по краям, то число шагов для победы заметно сокращается и в среднем составляет от 25 до 50.

3.5 Шаблон (Boilerplate) для участников хакатона

Шаблон (Boilerplate) предназначен для участников соревнований и представляет собой программный проект на определенном языке программирования, служащий основой для будущего алгоритма участника соревнования, который будет писать на данном языке программирования.

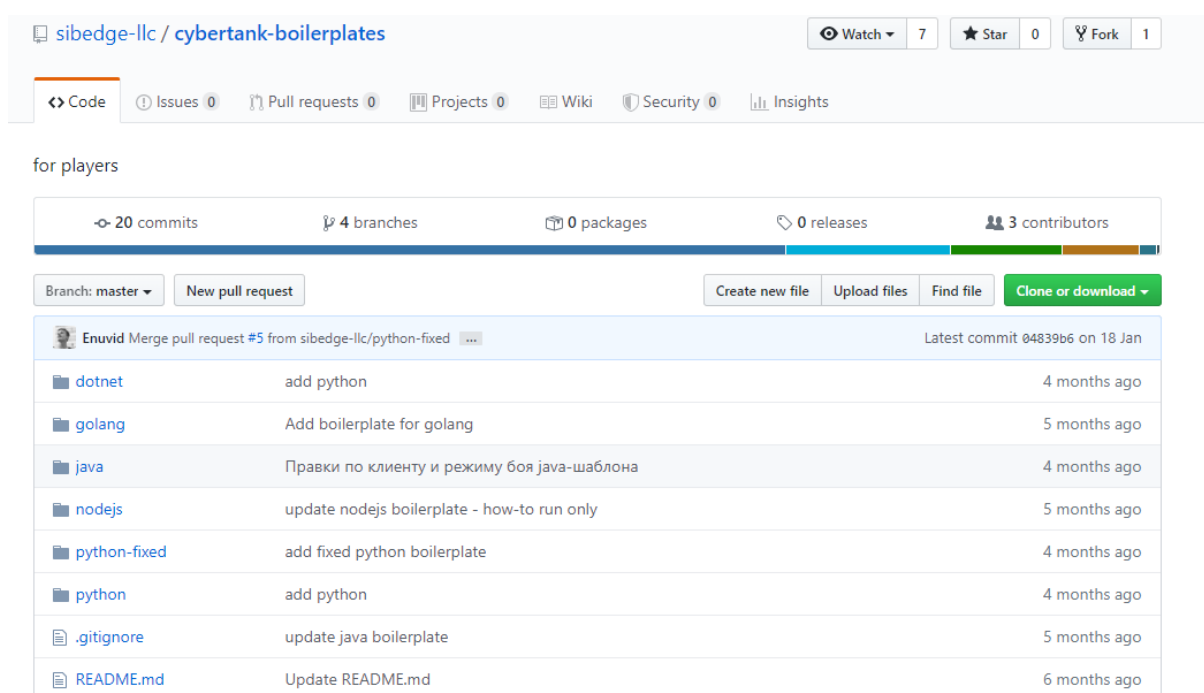


Рисунок 32. Структура проекта с шаблонами

В проекте реализованы основные методы для SignalR-коммуникации бота-клиента с игровым сервером.

В рамках данной выпускной квалификационной работы был разработан шаблон на языке Node.JS.

В шаблоне создается и запускается SignalR-соединение в одном из режимов: «Debug», «Test» или «Fight».

Ниже будут описаны конкретные методы для SignalR-соединения шаблона. Метода, названные с большой буквы реализованы на стороне игрового сервера, с маленькой – в самом шаблоне.

Отправка расстановки танков реализована с помощью методов «requestArrangement» и «ReceiveArrangement». С помощью метода «requestArrangement» шаблон «подписывается» на изменения сервера касательно отправленной расстановки, а метод «ReceiveArrangement» отправляет расстановку танков, тем самым инициируя выполнение текущего метода на стороне сервера.

Отправка событий о выстрелах реализована с помощью методов «requestArrangement» и «ReceiveArrangement». С помощью метода «requestArrangement» клиент «подписывается» на ответы сервера относительно каждого отправленного выстрела, а метод «ReceiveArrangement» вызывает серверный метод принятия этого выстрела.

Прослушивание клиентом сообщений от сервера о результатах выстрелов и итоге боя реализована с помощью метода «receiveMessage», который «подписывается» на обновления сервера и ожидает от него новых сообщений.

Шаблон предназначен для того, чтобы участники не погружались в особенности SignalR-коммуникации и настройки связи с сервером, а уделяли свое время на разработку эффективного алгоритма.

3.6 Тестирование веб-приложения

Тестирование программного модуля происходило двумя способами:

1. тестирование QA-инженером;
2. программное тестирование приложения с помощью Jest и Enzyme.

Покрытие программного кода тестами

Jest – это библиотека JavaScript для создания, запуска и структурирования тестов. Jest был специально создан для работы с React [12].

С помощью Jest и Enzyme написаны тесты в рамках frontend -части веб-приложения.

Написание тест-кейсов производилось для целей, описываемых далее.

- **Тестирование создания компонентов React при запуске приложения**

Чтобы протестировать компоненты React, создаётся помощник «setup», который пропускает «заглушки» как реквизиты для компонентов и обрабатывает компонент поверхностным рендерингом. Это позволяет отдельным тестам утверждать, были ли созданы данные компоненты или нет.

- **Тестирование работы Reducers**

«Reducer» – это инструмент для изменения состояния после вызова какого-либо «action», он должен вернуть новое состояние после применения действия к предыдущему состоянию. Тестирование функции «reducer» заключается в том, чтобы проверить корректность нового измененного состояния, которое определяется конкретным «reducer».

- **Тестирование работы Action Creators**

В Redux создатели действий (Action Creators) – это функции, которые возвращают простые объекты. При тестировании создателей действий мы хотим проверить, был ли вызван правильный создатель действий, а также было ли возвращено правильное действие.

- **Тестирование Middleware**

Функции промежуточного программного обеспечения являются надстройкой над поведением вызовов диспетчеризации в Redux, поэтому для проверки этого измененного поведения нам необходимо смоделировать поведение вызова диспетчеризации («dispatch»).

- **Тестирование функционала каждого компонента**

Тестирование функционала каждого компонента заключается в тестирование работоспособности и корректности всех его функций. По своей сути этот вид тестирования очень схож с Unit-тестированием в обычном его восприятии.

- **Отслеживание покрытие тестами программного кода**

Jest позволяет создавать отчеты, которые помогают в понимании тестовых покрытий всего программного кода в проекте.

Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

4.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

4.1.2 Потенциальные потребители результатов исследования

Данный программный модуль является HR-инструментом для компании, которая занималась разработкой этого модуля.

Потенциальными потребителями являются компании, заинтересованные в проведении хакатона для привлечения сотрудников в компанию, также сами сотрудники компании, которые занимаются организацией соревнований и участники соревнования – то есть разработчики, которые подбираются по определенным требованиям. Требования к разработчикам включают в себя владение определенным языком программирования, возраст и опыт разработки. Требования формируются исходя из текущей ситуации внутри компании, нехваткой специалистов определенного стека, а также требованиями рынка по квалификации специалистов.

В качестве потенциальных потребителей рассматриваются все IT-компании, которые занимаются коммерческой разработкой ПО. Поэтому для сегментации рынка будет учитываться охват рынка [13].

Таблица 2 – Карта сегментирования рынка

		Охват рынка		
		Компании г. Томска	Российские компании	Компании мирового рынка
Виды компаний	Крупные	–	+	+
	Средние	–	–	+
	Малые	–	–	–

Как показывает сегментация рынка в таблице 1, разработкой продуктов для внутренней инфраструктуры и для поиска новых сотрудников занимаются лишь крупные российские компании и большинство средних и крупных компаний мирового IT-рынка. Преимущественно для российских компаний, включая компании города Томска, такой вид разработки веб-приложений наиболее актуален, так как им занимается лишь совсем малое число бизнесов.

4.1.3 Анализ конкурентных технических решений

Текущий программный модуль, предназначенный для проведения хакатонов, был разработан для одной конкретной IT-компании города Томска. Подобные проекты для привлечения новых сотрудников IT-компании делают в основном на свою внутреннюю инфраструктуру без дальнейшего выдвижения проекта на коммерческий рынок.

Конкурентами разрабатываемого программного модуля являются приложения, которые помогают компаниям найти себе новых квалифицированных разработчиков в штат.

Примеры конкурентных решений:

1. Яндекс практикум
2. HeadHunter
3. SberGraduate

Яндекс практикум, как и SberGraduate, представляет собой онлайн-ресурс, где представлен набор курсов по разным направлениям. Курсы могут быть как платными, так и бесплатными. По итогу некоторых курсов или онлайн-олимпиад можно стать претендентом на стажировку в данной компании.

HeadHunter – система рекрутмента, где размещаются вакансии и резюме, и работодатель может найти своего сотрудника.

Обозначение разработок в таблице:

B_{ϕ} , K_{ϕ} – показатели для программного модуля, разрабатываемого в рамках ВКР.

B_{k1} , K_{k1} – показатели для Яндекс практикум

B_{k2} , K_{k2} – показатели для HeadHunter

B_{k3} , K_{k3} – показатели для SberGraduate

Для сравнения конкурентных решений максимальных бал возьмем равный 10.

Исходя из результатов, полученных в результате анализа конкурентных решений, согласно критериям, описанным в таблице 2, разрабатываемый программный модуль является конкурентноспособным, а также имеет определенные преимущества перед другими даже большими конкурентами.

Таблица 3 – Оценочная карта для сравнения конкурентных решений

Критерии оценки	Вес критерия	Баллы				Конкурентоспособность			
		Б _ф	Б _{к1}	Б _{к2}	Б _{к3}	К _ф	К _{к1}	К _{к2}	К _{к3}
Технические критерии оценки ресурсоэффективности									
1. Игровой формат	0,15	8	1	0	1	1,2	0,15	0	0,15
2. Короткие сроки поиска	0,1	8	5	4	5	0,8	0,5	0,4	0,5
3. Возможность оценить hard-skills кандидата	0,2	7	8	3	8	1,4	1,6	0,6	1,6
4. Возможность оценить soft-skills кандидата	0,2	5	0	3	0	1	0	0,6	0
5. Не требует вовлеченности большого числа людей	0,05	5	9	4	9	0,25	0,45	0,2	0,45
6. Отбор кандидатов по определенным критериям	0,1	7	4	7	4	0,7	0,4	0,7	0,4
Экономические критерии оценки эффективности									
7. Нестандартность метода	0,05	8	6	4	6	0,4	0,3	0,2	0,3
8. Перспективность программного решения	0,1	6	8	7	8	0,6	0,8	0,7	0,8
9. Законченность работы	0,05	5	7	7	7	0,25	0,35	0,35	0,35
Итого						6,6	4,55	3,75	4,55

4.1.4 Технология QuaD

С помощью технологии QuaD можно оценить качество программной разработка, а также ее перспективность на рынке [14].

Таблица 4 – Оценочная карта для сравнения конкурентных решений

Критерии оценки	Вес	Баллы	Макс. балл	Отн. знач.	Ср.-взвеш. знач.
Показатели оценки качества разработки					
1. Скорость проведения отбора специалистов	0,2	80	100	0,8	0,16
2. Требовательность системы к стабильной Интернет связи	0,05	50	100	0,5	0,025
3. Кроссплатформенность	0,1	90	100	0,9	0,09
4. Скорость обмена данными	0,15	70	100	0,7	0,105
5. Вовлеченности малого числа людей	0,15	70	100	0,7	0,105
6. Поддержка продукта после выпуска	0,05	75	100	0,75	0,0375
7. Скорость проведения отбора специалистов	0,2	80	100	0,8	0,16
Показатели оценки коммерческого потенциала разработки					
8. Нестандартность метода	0,1	80	100	0,8	0,08
9. Перспективность рынка	0,05	60	100	0,6	0,03
10. Законченность работы	0,05	65	100	0,65	0,0325
11. Конкурентоспособность программного модуля	0,1	75	100	0,75	0,075
Итого					0,74

Анализ, произведенный по технологии QuaD, показал, что разработка программного модуля имеет перспективность выше среднего, так как итоговое показательное значение попадает в диапазон от 60 до 79.

4.1.5 SWOT-анализ

SWOT-анализ является универсальным инструментом для исследования внутренних и внешних факторов, влияющих на продукт, а также его сильные и слабые стороны, угрозы и возможности [15].

Таблица 5 – Матрица SWOT-анализа

	<p>Сильные стороны:</p> <p>С1. Возможность оценки сразу нескольких кандидатов</p> <p>С2. Возможность оценить hard и soft skills кандидата</p> <p>С3. Короткие сроки (1 день)</p> <p>С4. Нестандартный подход к поиску сотрудников</p> <p>С5. Развлекательный, соревновательный и увлекательный формат для потенциальных участников</p> <p>С6. Поддержка проекта после выпуска</p>	<p>Слабые стороны:</p> <p>Сл1.Сложность организационного процесса</p> <p>Сл2. Ограниченный функционал</p> <p>Сл3. Выявление новых требований по ходу разработки</p>
<p>Возможности:</p> <p>В1. Расширение функционала.</p> <p>В2. Написание шаблонов на дополнительных языках программирования для расширения стека поиска кандидатов</p>	<p>С5.В1. В3 Внедрение нового функционала поможет программному модулю стать более адаптивным к внешним условиям, поддерживать свою актуальность и уникальность.</p>	<p>Сл2.В1. В3 Внедрение нескольких форматов для проведения соревнования поможет сохранить нестандартной подход к поиску</p>

<p>В3. Внедрение нескольких форматов игр и вариантов заданий</p> <p>В4. Использование олимпийской системы проведения игр для исключения случайных побед</p>	<p>С1.В2 Разнообразие шаблонов в плане языков программирования может помочь искать более широкий спектр разработчиков</p>	<p>сотрудников, не утратив актуальность</p>
<p>Угрозы:</p> <p>У1. Формат соревнования может быстро стать неактуальным</p> <p>У2. Потенциальные ошибки в ходе проведения соревнования</p> <p>У3. Ограниченный функционал может быстро устареть</p>	<p>С6.У2 Поддержка проекта после его выпуска поможет оперативно устранить появляющиеся неисправности</p> <p>С6.У3 Поддержка проекта также подразумевает функциональную доработку, что поможет расширить функционал программного модуля</p>	<p>Сл2.У1 У3</p> <p>Ограниченный функционал может привести к тому, что продукт быстро потеряет свою актуальность</p>

В результате проведения SWOT анализа было выявлено, что основные угрозы и слабости проекта связаны с ограниченным функционалом, но если у продукта будет поддержка после выпуска, которая включает в себя доработку функционала и устранение неисправностей, то потенциальных угроз и слабостей можно избежать.

4.2 Определение возможных альтернатив проведения научных исследований

В таблице 5 представлены альтернативные варианты развития программного модуля по критериям: тип заданий для соревнования, способ проведения боя между игроками и количество языков программирования, которое указывается в качестве критерия поиска для молодых специалистов.

Таблица 6 – Морфологическая таблица

	1	2	3
А. Вариант задания для соревнования	Упрощенный прототип игры «Морской бой»	Усложненный прототип игры «Морской бой»	Игровая реализация генетического алгоритма
Б. Способ проведения боя между игроками	1 бой с каждым из участников	2 боя с каждым из участников	Система боев по олимпийской системе
В. Количество языков программирования для поиска новых сотрудников	2	4	Больше 5

Оптимальными вариантами являются А1, Б2, В2. Выбирается упрощенный прототип игры «Морской бой» для того, что участники смогли за 6 часов реализовать и протестировать свой алгоритм, в таком случае организация всего мероприятия уложится в 8-9 часов. Способ проведения боев между участниками также обоснован целью минимизировать временные затраты, но при этом провести честные поединки участников. Количество языков программирования для формирования критерия поиска сотрудников обосновано потребностями компании на момент разработки программного модуля.

4.3 Планирование научно-исследовательских работ

4.3.1 Структура работ в рамках научного исследования

Важным аспектом планирования работ является определение и планирование основных этапов разработки на проекте, а также декомпозиция этих этапов и подробное описание работ.

Студент, исполнитель текущей ВКР, в рамках выполнения проекта исполнял должность frontend-разработчика.

Таблица 7 – Основные этапы выполнения ВКР

Основные этапы	№	Описание этапа	Исполнители
Аналитика	1	Составление и утверждение ТЗ	Системный аналитик
Организация процессов	2	Подбор команды участников проекта	Руководитель проекта
Планирование работ	3	Разработка календарного плана	Руководитель проекта, дизайнер, backend-разработчик, frontend-разработчик (студент)
	4	Выбор технологий разработки	Руководитель проекта, backend-разработчик, frontend-разработчик (студент)
Разработка программного модуля	5	Управление проектом	Руководитель проекта
	6	Проектирование дизайна программного модуля	Дизайнер
	7	Разработка игрового сервера	backend-разработчик, frontend-разработчик (студент)
	8	Разработка тестового бота-клиента на Node.JS	frontend-разработчик (студент)
	9	Разработка frontend части	frontend-разработчик
	10	Разработка шаблонов для участников	Привлечённые разработчики определенных языков

			программирования
Тестирование	11	Тестирование программного модуля	QA-специалист
Документирование и подведение итогов	12	Оформление основной части диплома	Студент
	13	Написание части «Социальная ответственность»	Студент
	14	Написание «Финансовый менеджмент»	Студент

4.3.2 Определение трудоемкости выполнения работ

Определим ожидаемое (среднее) значение трудоемкости $t_{ожи}$, используя следующую формулу:

$$t_{ожи} = \frac{3t_{мини} + 2t_{макс i}}{5}$$

где $t_{ожи}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{мини}$ – минимально возможная трудоемкость выполнения заданной i -ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;

$t_{макс i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

$$T_{pi} = \frac{t_{ожи}}{Ч_i}$$

4.3.3 Разработка графика проведения научного исследования

Линейный график строится на основании полученных значений $t_{\text{раб}}$, предварительно переведенных в календарные дни по формуле:

$$t_k = t_{\text{раб}} \cdot K_n, \quad (3)$$

где t_k – длительность этапов работ в календарных днях;

K_n – коэффициент календарности.

Произведем расчет коэффициента календарности за 2020 год

$T_k = 365$ – количество календарных дней,

$T_{\text{вд}} = 108$ – количество выходных дней,

$T_{\text{пд}} = 10$ – количество праздничных дней,

$$K_{\text{кал}} = \frac{366}{366 - 108 - 10} = 1,48$$

При округлении до целого $K_{\text{кал}} = 1$.

Таблица 8 демонстрирует временные показатели для всех работ на проекте.

Таблица 8 – Временные показатели проведенных работ

Название работы	Продолжительность работы в днях			Трудоемкость	
	t_{min}	t_{max}	$t_{\text{ож}}$	$t_{\text{раб}}$	t_k
Аналитика проекта	15	20	17	17	17
Подбор команды участников проекта	2	7	4	4	4
Разработка календарного плана	0,4	2	1	1	1
Выбор технологий разработки	0,3	1	0,58	0,29	0,29
Управление проектом	60	75	66	66	66
Проектирование дизайна программного модуля	20	30	24	24	24
Разработка игрового сервера	60	70	64	64	64

Название работы	Продолжительность работы в днях			Трудоемкость	
	t_{min}	t_{max}	$t_{ож}$	$t_{раб}$	t_k
Разработка тестового бота-клиента на 34	4	7	5,2	5,2	5,2
Разработка frontend части	40	55	46	46	46
Разработка шаблонов для участников	3	6	4,2	1	1
Тестирование программного модуля	30	40	34	34	34
Итого			268	265	265

Проект стартовал в начале 2020 года 13.01.2020. Разработка завершилась 20.04.2020, а окончание всего процесса совместно с документированием и подведением итогов 29.05.2020.

Таким образом, суммарное время исполнения проекта составило 4,5 месяца.

На рисунке 1 представлена диаграмма Ганта с планом-графиком проводимых работ. Планирование приведено с учетом выходных и праздничных дней.

Таблица 9 – Обозначение ролей по цветам на диаграмме Ганта

Системный аналитик	Зеленый
Руководитель проекта	Розовый
Дизайнер	Фиолетовый
Backend-разработчик	Синий
Frontend-разработчик (студент)	Оранжевый
Привлеченные разработчики для написания шаблонов	Коричневый
QA-специалист	Желтый

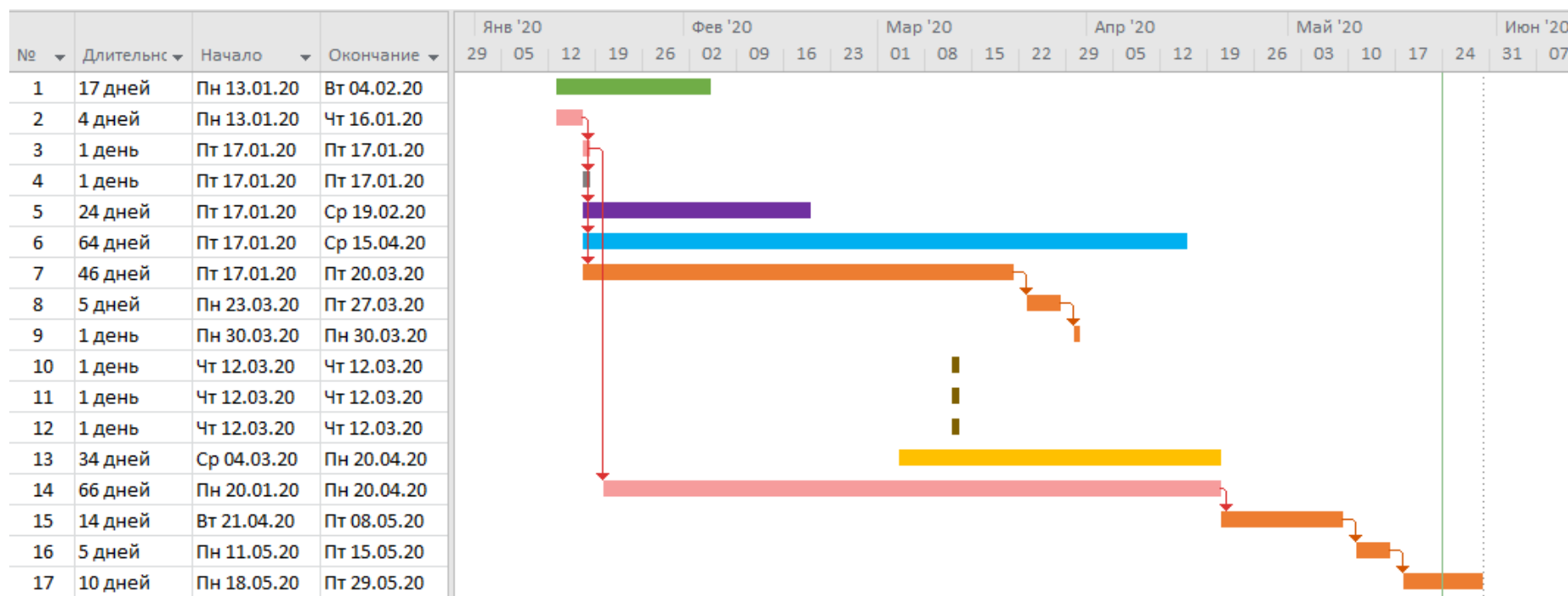


Рисунок 33. Диаграмма Ганта

4.4 Бюджет научно-технического исследования (НТИ)

4.4.1 Расчет затрат на специальное оборудование для научных работ

Поскольку разработка программного модуля не включала в себя затраты на приобретение дополнительного оборудования или иных ресурсов, а использовалось только компьютерное оборудование, предоставленное компанией.

Срок амортизации одного персонального компьютера составляет 30 месяцев. Первоначальная стоимость ПК составляет 70 000 рублей.

Норма амортизации: $\frac{1}{30} * 100\% = 3,33\%$

Ежемесячные отчисления составят: $70\,000 * 3,33\% = 2\,333,3$ рублей

Сумма отчислений за 6 ПК за период работы над проектом (4 месяца) составит: $2\,331 * 4 * 6 = 55\,944$ рублей

Таблица 10 – Затраты на амортизацию

Наименование оборудования	Первонач. стоимость, руб	Кол-во	Ежемесячная норма амортизации, %	Срок использования, мес	Аморт. отчисления в мес, руб.	Всего, руб
Персональный компьютер	70 000	6	3,33%	4	2 331	55 944

4.4.2 Основная заработная плата исполнителей

При наборе команды разработки компания наняла на позиции QA-специалиста, backend и frontend разработчиков в качестве стажеров. Условия работы стажеров: 3 месяца неоплачиваемой стажировки и дальнейшее трудоустройство на полную ставку. Текущий проект является некоммерческим, поэтому компании было выгодно нанять стажеров для того, чтобы минимизировать свои расходы.

Таблица 11 – Баланс рабочего времени

	Календарное число дней	Количество нерабочих дней	Потери рабочего времени	Действительный годовой фонд рабочего времени
Системный аналитик	365	118	24	223
Руководитель проекта	365	118	24	223
Дизайнер	365	118	27	220
Backend-разработчик	365	118	24	223
Frontend-разработчик	365	118	24	223
Привлеченные разработчики для написания шаблонов	365	118	24	223
QA-специалист	365	118	24	223

Дата выхода на стажировку frontend- и backend-разработчиков – с 13 января по 13 апреля, а QA-специалиста с 20 февраля по 20 мая. Количество неоплачиваемых рабочих дней у стажеров приведено в таблице 10:

Таблица 12 – Количество неоплачиваемых дней для стажеров

Месяц	Frontend- разработчик	Backend- разработчик	QA-специалист
Январь 2020	13	13	–
Февраль 2020	19	19	–
Март 2020	21	21	19
Апрель 2020	–	9	13
Итого	53	62	32

Исходя из таблицы 10, работа frontend разработчика и QA-специалиста на данном проекте будет полностью укладываться в стажировку и, следовательно, не будет оплачена. Backend разработчик имеет 2 оплачиваемых рабочих дня ($64 - 62 = 2$).

Следует отметить, что в проекте присутствуют участники, которые работают неполный рабочий день, как привлечены к нескольким проектам. А именно:

- Руководитель проекта – 3 часа в день
- QA-специалист – 4 часа в день
- Системный аналитик – 4 часа в день

Остальные участники: дизайнер, frontend-разработчик и backend-разработчик работают полный восьмичасовой рабочий день.

Поскольку некоторые участники работают неполный рабочий день, дневная заработная плата будет считаться по формуле:

$$Z_{\text{дн}} = \frac{Z_{\text{м}} * M}{F_{\text{д}}} * \frac{T_{\text{дн}}}{8}$$

где $Z_{\text{м}}$ – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года: при отпуске в 24 раб. дня берется значение = 11,2

$F_{\text{д}}$ – действительный годовой фонд рабочего времени персонала, раб. дн. (табл. 13)

$T_{\text{дн}}$ – количество часов в день, которое сотрудник уделяет данному проекту

Таблица 13 – Расчет основной заработной платы

Исполнитель	Тарифная ЗП	Прем. коэф-т	Районный коэф-т	Коз-т доплат	ЗП за месяц	ЗП за день	Кол-во дней	Часов в день	Общая ЗП за проект
Системный аналитик	20 500	1,1	1,3	1,1	32 247	810	17	4	13 766
Руководитель проекта	20 500	1,1	1,3	1,1	32 247	607	71	3	43 121
Дизайнер	19 000	1,1	1,3	1,1	29 887	1 522	24	8	36 516
Привлеченные разработчики для написания шаблонов	21 000	1,1	1,3	1,1	33 033	1 659	3	8	4 977
Backend-разработчик	20 000	1,1	1,3	1,1	31 460	1 580	2	8	3 160
Frontend-разработчик	20 000	1,1	1,3	1,1	31 460	1 580	0	8	0
QA-специалист	19 000	1,1	1,3	1,1	29 887	751	0	4	0
Общая заработная плата									101 541

4.4.3 Дополнительная заработная плата исполнителей темы

Оклад сотрудников компании не предусматривает наличие дополнительных выплат. Ежемесячные выплаты сотрудникам постоянны, это является частью политики компании.

4.4.4 Отчисления во внебюджетные фонды (страховые отчисления)

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{\text{внеб}} = k_{\text{внеб}} \cdot (З_{\text{осн}} + З_{\text{доп}}) \quad (7)$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Коэффициент отчислений во внебюджетные фонды ПФР – 22%.

Поскольку сотрудники в компании получают только основную заработную плату без надбавок, то в расчёте отчислений во внебюджетные фонды дополнительная заработная плата не участвует.

Таблица 14 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.
Системный аналитик	32 247
Руководитель проекта	32 247
Дизайнер	29 887
Привлеченные разработчики для написания шаблонов	33 033
backend-разработчик	31 460
frontend-разработчик	31 460
QA-специалист	29 887
Итого	48 449

4.4.5 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = (\text{сумма расходов статей}) \cdot k_{\text{нр}}$$

Возьмем $k_{\text{нр}} = 16\%$. $Z_{\text{накл}} = 205\,933 * 16\% = 32\,949$ рублей

4.4.6 Формирование бюджета затрат проекта

Ниже проведены финальные зачёты затрат на реализацию проекта. Средства на дополнительную заработную плату, командировки и контрагентские расходы не были потрачены.

Таблица 15 – Расчет бюджета затрат

Наименование	Сумма, руб.
Материальные затраты	0
Затраты на специальное оборудование	55 944
Затраты по основной заработной плате исполнителей темы	101 541
Затраты по дополнительной заработной плате исполнителей темы	0
Отчисления во внебюджетные фонды	48 449
Затраты на научные и производственные командировки	0
Контрагентские расходы	0
Накладные расходы	32 949
Бюджет затрат НИИ	238 883

Общий бюджет проекта составил 238 883 рублей, что ниже верхней границы бюджета (300 тысяч рублей).

4.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Ниже приведен анализ эффективности разрабатываемого программного модуля.

Таблица 16 – Оценка характеристик проекта

Критерии	Весовой коэффициент параметра	Оценка
Возможность повторного использования	0,4	5
Способность привлечения новых сотрудников	0,3	4
Расширяемость	0,2	4
Удобство в эксплуатации	0,1	4
	Ip	4,4

Для нахождения интегрального показателя эффективности вариантов исполнения необходимо найти частное ресурсоэффективности и интегрального финансового показателя.

Сравнение интегрального показателя эффективности вариантов исполнения позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Максимальная стоимость проекта составляет 300 тысяч рублей.

Таблица 17 – Сравнительная эффективность разработки

Показатели	Значение
Интегральный финансовый показатель разработки	0,8
Интегральный показатель ресурсоэффективности разработки	4,4
Интегральный показатель эффективности	5,5

На основе расчетов, приведенных в таблице 15, можно сделать вывод о том, что данный программный модуль является эффективным с точки зрения финансовой и ресурсной эффективности.

4.6 Выводы по главе

В ходе выполнения задания по текущему разделу был произведен анализ рынка и конкурентных решений, выявлены сильные и слабые стороны проекта. Проведено планирование работ, совместно в определении трудоёмкости проекта и общих трудозатрат работников. К тому же был проанализирован и просчитан бюджет выполнения данного проекта. Бюджет разработка программного модуля составит 238 883 рублей, а срок разработки – 101 календарный день.

В результате было выявлено, что проект имеет определенные конкурентные преимущества на рынке, а также является эффективным для исполнения.

Глава 5. Социальная ответственность

5.1 Введение

Темой данной выпускной квалификационной работы является проектирование и разработка программного модуля, предназначенного для проведения технических соревнований между разработчиками. Участники мероприятия должны написать алгоритм (бота), который должен уметь играть в игру по заданным правилам. Такие соревнования являются своеобразным HR-инструментом, которым позволяет искать новых квалифицированных сотрудников в компанию. Потенциальными пользователями являются сотрудники компании, а также участники соревнования.

Актуальность разрабатываемого программного модуля обусловлена тем, что на данный момент наблюдается сильная нехватка профессионально-подготовленных кадров в IT-сфере. Проблемой многих IT-компаний становится дефицит специалистов, обладающих компетенциями в области информационных технологий.

Разработка программного модуля производилась в офисе компании ООО «Сибэдж» по адресу город Томск, улица Пушкина 75. Офис, в котором проходила разработка программного модуля, имеет размеры 5 x 10м. Имеется 1 большое окно размером 2 x 1м, на окне установлены жалюзи, также присутствует 1 кондиционер. Для обеспечения пожарной безопасности в офисе имеется порошковый огнетушитель ОП-03, план эвакуации, датчики дыма и система пожаротушения тонкораспыленной водой. Освещение организуется с помощью 6 квадратных светильников ЕЛ ЛПО 620 x 620мм на потолке. В офисе размещены 10 рабочих мест для разработчиков.

5.2 Правовые и организационные вопросы обеспечения безопасности

Оплата труда разработчиков программного обеспечения происходит дважды в месяц: первая часть – аванс (30% от заработной платы), и основная окладная часть (70% от заработной платы). Нормой труда сотрудника является 8 часов дней 5 дней в неделю, также работник отдыхает в нерабочие праздничные дни, которые определены трудовым кодексом РФ [16]. Рабочий день строго не нормирован, разработчик сам занимается планированием своего рабочего времени. Обязательно лишь присутствовать на важных совещаниях и ежедневных митингах.

Учитывая то, что деятельность разработчика программного модуля для проведения технических соревнований между разработчиками предполагает работу с компьютером, то рабочее место должны быть обустроено исходя из норм, описанных в ГОСТ 12.2.032-78 «Рабочее место при выполнении работ сидя. Общие эргономические требования» и СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений».

Регулярная работа с компьютером оказывает негативное воздействие на зрение и длительную концентрацию внимания, а также создает физический дискомфорт, связанный с сидячим положением. Именно поэтому законодательством устанавливаются определенные правила организации рабочего места:

- оптимальное расположение клавиатуры – 0,1 – 0,3 м от края стола;
- расстояние между рабочими столами – не менее 2 м;
- между боковыми гранями мониторов – не менее 1,2 м;
- угол наклона спинки стула – около 30°;
- высота стола – не менее 0,6 м;

- площадь на одного сотрудника, который работает за компьютером, составляет 4,5 – 6 м².

5.3 Производственная безопасность

Как при разработке программного модуля, так и при проведении технических соревнований основным фактором безопасности является грамотно обустроенная рабочая среда в офисе или помещении для проведения технических соревнований.

Основными опасными и вредными факторами, которые оказывают негативное влияние на здоровье разработчиков являются показатели, приведенные в таблице 1.

Таблица 18 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разработка	Внедрение	Эксплуатация	
1.Отклонение показателей микроклимата	+	+	+	СанПиН 2.2.4.3359-16 «Санитарно-эпидемиологические требования к физическим факторам на рабочих местах» [17] СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений» [18]
2. Превышение уровня шума	+	+	+	
3.Отсутствие или недостаток естественного света	+	+	+	Санитарно-эпидемиологические правила и нормативы СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» [19] СанПиН 2.2.2.1332 - 03. Гигиенические требования к организации работы на копировально-множительной технике [20] СП 44.13330.2011 Административные и бытовые здания. Актуализированная редакция СНиП 2.09.04-87 [21]
4.Недостаточная освещенность рабочей зоны	+	+	+	

				ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя [22]
--	--	--	--	---------------------------------------------------------------------

Анализ опасных и вредных производственных факторов

Микроклимат помещения

Причиной нарушения микроклимата в офисных помещениях может быть неправильная теплоизоляция помещения, недостаточная или избыточная вентиляция, нарушение норм по отоплению помещений, а также слишком частое или слишком редкое проветривание помещения.

Повышенная температура оказывает негативное влияние на сердечно-сосудистую систему, происходит повышенная утомляемость. Пониженная температура способствует ослаблению иммунитета, возрастает риск заражения заболеваниями.

Пониженная влажность оказывает негативное влияние на слизистые оболочки и кожу, может привести к таким последствиям, как бронхиальная астма, слезливость глаз, заложенность в носу, обострение аллергических реакций.

Условия микроклимата делятся на 2 периода:

1. Холодный – со среднесуточной температурой воздуха ниже + 10°C
2. Теплый – со среднесуточной температурой воздуха выше + 10°C

Нормы для микроклимата в офисе должны соответствовать следующим требованиям, представленным на рисунке 1 согласно ГОСТ 12.1.005 – 88 и СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений».

Категория работ	Температура, °С			Относительная влажность, %		Скорость движения, м/с	
	оптим.	допустимая		оптим.	допуст.	оптим.,	допуст.
		верхняя граница	нижняя граница				
Легкая - Ia	23-25	28	22	40-60	55 (при 28°С)	0,1	0,1-0,2
Легкая - Ib	22-24	28	21	40-60	60 (при 27°С)	0,2	0,1-0,3
Средней тяжести - Па	21-23	27	18	40-60	65 (при 26°С)	0,3	0,2-0,4
Средней тяжести - Пб	20-22	27	16	40-60	70 (при 25°С)	0,3	0,2-0,5
Тяжелая - Пв	18-20	26	15	40-60	75 (при 24°С)	0,4	0,2-0,6

Рисунок 34. Нормы для микроклимата

Оптимальная температура воздуха для разработчиков программного обеспечения и участников соревнований составляет 23 - 25°С при относительной влажности воздуха 40-60%.

Уровень шума

Источником возникновения шума могут быть условия за окном офиса, например, строительные работы, а также работа офисного технического оборудования.

Воздействие шума может спровоцировать проблемы нервной системы и органов слуха. К тому же ухудшается длительная концентрация внимания и работоспособность сотрудников в целом.

Оптимальный уровень шума описан в СН 2.2.4/2.1.8.562-96 и для офисного помещения применительно к разработчикам программного продукта, а также к участникам технического соревнования, при незначительной физической нагрузке не должен превышать 80 дБ.

Освещенность

Причиной неправильной организации освещения в офисе может быть недостаток источников естественного света, неправильное искусственное освещение, а также неправильно подобранные типы поверхностей в помещении, такие как стены, потолок, пол, рабочие поверхности, которые создают неблагоприятный для зрения коэффициент отражения.

Последствиями неправильной освещенности рабочей среды может стать ухудшение зрения сотрудников, болезненные ощущения в глазах, а также повышенная утомляемость из-за чрезмерной нагрузки на глаза работников.

Оптимальный уровень освещенности рабочего места установлен в СП 52.13330.2016 «Естественное и искусственное освещение» и должен составлять 300-500 люкс, коэффициент пульсации не должен превышать 20%

Для разных поверхностей оптимальными считают следующие коэффициенты отражения:

- стены – 0,3-0,5
- пол – 0,1-0,4
- потолок – 0,6-0,8
- рабочие поверхности – 0,2-0,7

Уровень освещенности в офисе, где проводилась разработка программного модуля составляет около 250 люкс в дневное время и 200 люкс в вечернее, что является ниже установленной нормы.

Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов на исследователя (работающего)

Микроклимат

Для снижения температуры может помочь кондиционер или вентилятор, если повышение температуры случилось в холодное время года,

можно уменьшить уровень подаваемого отопления в помещение с помощью регулятора.

Для повышения температуры можно также урегулировать уровень подаваемого отопления в помещении в случае, если отопление включено. Для обогрева помещения также можно использовать тепловые пушки.

Для регулирования влажности воздуха следует регулярно проветривать помещение и использовать увлажнители воздуха и низком уровне влажности воздуха.

Условия микроклимата на рабочем месте разработчиков программного модуля для проведения технических соревнований между разработчиками удовлетворяет всем требованиям. В офисе имеется кондиционер, тепловая пушка, а также батареи с возможностью регулировки уровня отопления.

Освещение

Уровень освещенности в офисе, где проводилась разработка программного модуля составляет около 250 люкс в дневное время и 200 люкс в вечернее. Причиной пониженного уровня освещенности является то, что часть ламп для искусственного освещения на потолке не функционирует. Данную проблему можно устранить заменой старых ламп на новые, а также обеспечить сотрудников офиса дополнительным освещением, например настольными лампами.

Разработчикам с пониженным зрением, работающим постоянно за компьютером, рекомендовано носить очки или контактные линзы, чтобы минимизировать нагрузку на глаза, уменьшить степень утомляемости и увеличить продуктивность.

Уровень шума

Устранить высокий уровень шума можно путем улучшения изоляции помещения, например усиленная шумоизоляция окон, а также заменой техники, которая производит превышающее количество шума, на более новую, которая будет удовлетворять нормам. Уровень шума на рабочем

месте разработчиков программного модуля не превышает допустимых значений.

Работа с электрооборудованием

Поскольку разработчики участники соревнования работают с компьютерами, следует учесть правила безопасности, которые уберегут от возможности поражения организма электрическим током.

Перед тем, как начать работу с ПК необходимо удостовериться в том, что техника исправна, а также исправны выключатели и электропроводка.

Запрещается делать следующее:

- развешивать какие-либо вещи на проводах, пережимать их и подвергать постоянному воздействию
- прикасаться к соединительным кабелям и их разъемам
- проводить ремонтные работы компьютерной техники без специальных знаний и разрешения
- работать во влажной одежде и влажными руками

В офисе, где проводилась разработка программного модуля для проведения технических соревнований между разработчиками, разработчики соблюдают меры предосторожности, чтобы не навредить своему здоровью и не причинить ущерб техническому оборудованию.

Проанализировав все вышеперечисленные факторы, делаем вывод о том, что рабочее место соответствует принятым нормам.

5.4 Экологическая безопасность

Разрабатываемый программный модуль не оказывает прямого влияния на атмосферу и гидросферу.

Как и при любой человеческой деятельности, существуют бытовые отходы, которые остаются в результате функционирования офиса для разработки программных продуктов. Бытовые отходы должны быть утилизированы, согласно нормам, СанПиН 2.1.2.2645-10 «Санитарно-

эпидемиологические требования к условиям проживания в жилых зданиях и помещениях».

Также при разработке и проведении технических соревнований остается достаточное количество бумажных отходов, такие как распечатки, брошюры, приглашения, постеры и тому подобное. Для утилизации таких отходов компания сдает всю накопленную макулатуру в пункты приема и переработки макулатуры в городе Томске.

Таким образом программный модуль имеет лишь косвенное отношение к загрязнению литосферы посредством бытовых и офисных отходов.

5.5 Безопасность в чрезвычайных ситуациях

Разрабатываемый программный модуль для проведения технических соревнований между разработчиками не может напрямую стать причиной чрезвычайной ситуации. Однако, как и любая программная система, он размещен на физическом сервере, который представляет собой компьютерный кластер. Как и любая техника, данный кластер может перегреться и воспламениться, что может стать причиной пожара.

Каждый работник должен пройти противопожарный инструктаж и знать нормы поведения при этой чрезвычайной ситуации.

Как только обнаружится задымление запах гари или непосредственно огонь, следует незамедлительно звонить по номеру 001 (или 101 для мобильных операторов). Всем сотрудникам эвакуироваться в случае серьезной опасности.

В случае проведения соревнований следует провести краткий противопожарный инструктаж для участников и организаторов мероприятия, показать способы эвакуации, запасные выходы и план эвакуации в текущем здании.

Чтобы предотвратить серьезные последствия пожара, а также минимизировать их, необходимо наличие в каждом офисе плана эвакуации, огнетушителя, а также наличие датчиков дыма и систем пожаротушения.

Все меры пожарной безопасности определены согласно нормам в СНиП 21-01-97 от 1998-01-01 «Пожарная безопасность зданий и сооружений» [23].

5.6 Выводы по главе

В ходе разработки раздела «Социальная ответственность» для данной выпускной квалификационной работы были выявлены основные правовые и организационные вопросы обеспечения безопасности, а также обеспечения комфортных условий труда для офисных помещений.

Был произведен анализ вредных факторов, влияющих на работника в процессе разработки программного модуля для проведения технических соревнований между разработчиками, потенциальные угрозы жизни и здоровью сотрудников, а также пути исправления нарушений и пути оптимизирования рабочего места сотрудников в соответствие с государственными правилами и нормами.

Помимо того, были выявлены проблемы экологической безопасности, основное воздействие процесса разработки программных продуктов на окружающую среду и способы уменьшения вредного воздействия.

Также определены потенциально-возможные чрезвычайные ситуации в офисных помещениях, способы защиты от них и пути ликвидации последствий.

Заключение

В рамках текущей выпускной квалификационной работы было выполнено следующее:

1. реализация внутренней коммуникации между игровым сервером и ботом-клиентом, а также между сервером и frontend-частью программного комплекса;
2. разработка frontend-части веб-приложения;
3. полное покрытие кода тестами с помощью Jest;
4. разработка тестового клиент-бота на Node.JS;
5. разработка шаблона для участников хакатона на Node.JS.

После окончания разработки было проведено внутреннее соревнование между сотрудниками компании, а после проведено полноценное мероприятие, в котором принимали участие молодые специалисты. Хакатон был рассчитан на использование языков Java, Python, Node.JS и C#. Разработчиков давалось на разработку 6 часов. По итогу проведения хакатона были выбраны победители, и 2 разработчика были приглашены на работу в компанию.

Проведение хакатонов для набора новых сотрудников оказалось очень интересным и увлекательным мероприятием не только для самих участников, но и для сотрудников компании, которые занимались разработкой. Данный метод в целом занял 1 день, позволил оценить hard- и soft skills кандидатов, повысить интерес к компании у участников, а также по итогу набрать новых сотрудников в штат компании.

Список литературы

1. The complete guide to organizing a successful hackathon. [Электронный ресурс]. – URL: <https://www.hackerearth.com/> (дата обращения 01.02.2020).
2. Single-page application vs. multiple-page application. [Электронный ресурс]. – URL: <https://medium.com/> (дата обращения 03.03.2020).
3. React JS. [Электронный ресурс]. – URL: <https://ru.reactjs.org/> (дата обращения 03.02.2020).
4. Redux. [Электронный ресурс]. – URL: <https://redux.js.org/> (дата обращения 03.02.2020).
5. Realtime Technology Glossary. [Электронный ресурс]. – URL: <https://www.pubnub.com/learn/glossary/what-is-»SignalR»/> (дата обращения 09.01.2020).
6. Gustavo Armenta Valdez (December 12, 2012). ««SignalR»: Building real time web applications». Microsoft MSDN.
7. TypeScript – JavaScript that scales [Электронный ресурс]. – URL: <https://www.TypeScriptlang.org/> (дата обращения 01.02.2020).
8. Styled-components [Электронный ресурс]. – URL: <https://styled-components.com/> (дата обращения 03.02.2020).
9. Real-time ASP.NET with «SignalR». [Электронный ресурс]. – URL: <https://dotnet.microsoft.com/apps/aspnet/»SignalR»/> (дата обращения 11.01.2020).
10. Дидыч Я.О., Малинецкий Г.Г. Поиск оптимальных алгоритмов действий противников в игре «Морской бой» // Препринты ИПМ им. М.В.Келдыша. 2015. № 98. 16 с.
11. Дж. Д. Вильямс. Совершенный стратег, или Букварь по теории стратегических игр; пер. с англ. М.: Советское радио, 1960. 270 с.
12. Jest · Delightful JavaScript Testing [Электронный ресурс]. – URL: <https://jestjs.io/> (дата обращения 22.03.2020).

13. Видяев И.Г., Серикова Г.Н., Гаврикова Н.А. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение: учебно-методическое 89 пособие / И.Г. Видяев, Г.Н. Серикова, Н.А. Гаврикова, Н.В. Шаповалова, Л.Р. Тухватулина, З.В. Криницына; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2014. – 36 с
14. Технология оценки бизнеса QuaD [Электронный ресурс] / Центр Креативных Технологий. Режим доступа: <https://www.inventech.ru/> – Загл. с экрана 13. Цулая И. Н.
15. SWOT-анализ в системе стратегического управления: особенности применения и пути адаптации к объектам мезоуровня // Вестник ВолГУ. Серия 3: Экономика. Экология. 2010. №2.
16. Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ [Электронный ресурс] / Консультант Плюс. URL: https://www.consultant.ru/document/cons_doc_LAW_34683/, свободный (дата обращения: 05.05.2018)
17. СанПиН 2.2.4.3359-16 Санитарно-эпидемиологические требования к физическим факторам на рабочих местах [Электронный ресурс] URL: <http://docs.cntd.ru/document/420362948/> (дата обращения: 04.04.2020).
18. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений [Электронный ресурс] URL: <http://docs.cntd.ru/document/901704046> (дата обращения: 04.04.2020).
19. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы [Электронный ресурс] URL: <http://docs.cntd.ru/document/901865498> (дата обращения: 03.04.2020).
20. Санитарно-эпидемиологические правила и нормативы СанПиН 2.2.2.1332 - 03. Гигиенические требования к организации работы на копировально-множительной технике [Электронный ресурс] URL: <http://docs.cntd.ru/document/901865553> (дата обращения: 03.04.2020).

21. СП 44.13330.2011 Административные и бытовые здания. Актуализированная редакция СНиП 2.09.04-87 (с Поправкой, с Изменениями N 1, 2) [Электронный ресурс] URL: <http://docs.cntd.ru/document/1200084087> (дата обращения: 03.04.2020).
22. ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования [Электронный ресурс] URL: <http://docs.cntd.ru/document/1200003913> (дата обращения: 03.04.2020).
23. СНиП 21-01-97* Пожарная безопасность зданий и сооружений (с Изменениями N 1, 2) [Электронный ресурс] URL: <http://docs.cntd.ru/document/871001022> (дата обращения: 03.04.2020).