

Инженерная школа информационных технологий и робототехники
 09.03.02 Информационные системы и технологии
 Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка серверного приложения для асинхронного симметричного коммуницирования сетевых клиентов

УДК 004.455.:004.383.2

Студент

Группа	ФИО	Подпись	Дата
8И6Б	Бедер Константин Борисович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Токарева О.С.	К.Т.Н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШПИБ	Рыжакина Т. Г.	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Белоеенко Е. В.	К.Т.Н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Цапко И.В.	К.Т.Н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результатов	Результат обучения (выпускник должен быть готов)
Профессиональные и общепрофессиональные компетенции	
P1	Применять базовые и специальные естественнонаучные и математические знания для комплексной инженерной деятельности по созданию, внедрению и эксплуатации геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
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 геоинформационных систем и технологий, информационных систем в бизнесе, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Выполнять комплексные инженерные проекты по созданию информационных систем и технологий, а также средств их реализации (информационных, методических, математических, алгоритмических, технических и программных).
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 геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P6	Внедрять, эксплуатировать и обслуживать современные геоинформационные системы и технологии, информационные системы и технологии в бизнесе, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
Универсальные (общекультурные) компетенции	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом. Владеть иностранным языком (углублённый английский язык), позволяющим работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена команды, состоящей из специалистов различных направлений и квалификаций.
P10	Демонстрировать личную ответственность за результаты работы и готовность следовать профессиональной этике и нормам ведения комплексной инженерной деятельности.
P11	Демонстрировать знания правовых, социальных, экологических и культурных аспектов комплексной инженерной деятельности, а также готовность к достижению должного уровня физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности.

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

Инженерная школа информационных технологий и робототехники
 09.03.02 Информационные системы и технологии
 Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Цапко И.В.
 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8И6Б	Бедер Константин Борисович

Тема работы:

Разработка серверного приложения для асинхронного симметричного коммуницирования сетевых клиентов	
Утверждена приказом директора (дата, номер)	28.02.2020, № 59-61/с

Срок сдачи студентом выполненной работы:	05.06.2020
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе</p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Объектом проектирования в исследовательской работе является веб-приложение для отдела HR. Режим работы: клиент-сервер. Особые требования к продукту: независимость развертывания веб-приложения от конкретного хостинга; поддержка всех современных браузеров; модульная архитектура.</p>
---	--

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> 1. Исследование предметной области; 2. Проектирование веб-приложения; 3. Программная реализация; 4. Анализ результатов разработки веб-приложения; 5. Финансовый менеджмент; 6. Социальная ответственность.
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<p>Архитектура системы; Граф переходов конечного автомата.</p>
<p>Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i></p>	
<p>Раздел</p>	<p>Консультант</p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Рыжакина Т. Г.</p>
<p>Социальная ответственность</p>	<p>Белоенко Е. В.</p>
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	
<p>Все разделы должны быть написаны на русском языке</p>	

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	<p>27.01.2020</p>
--	-------------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Токарева О.С.	к.т.н.		27.01.2020

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И6Б	Бедер Константин Борисович		27.01.2020

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

Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.02 «Информационные системы и технологии»
 Уровень образования бакалавриат
 Отделение информационных технологий
 Период выполнения осенний/весенний семестр 2019/2020 учебного года

Форма представления работы:

бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	5.06.2020
--	-----------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
14.02.2020	1. Анализ предметной области	20
02.03.2020	2. Проектирование системы	20
15.04.2020	3. Программная реализация	20
25.05.2020	4. Тестирование	15
01.06.2020	5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
01.06.2020	6. Социальная ответственность	10
05.06.2020	7. Оформление пояснительной записки	5

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Токарева О.С.	к.т.н.		27.01.2020

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Цапко И.В.	к.т.н.		27.01.2020

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

Студенту:

Группа	ФИО
8И6Б	Бедеру Константину Борисовичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.02 Информационные системы и технологии

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 затрат НИ: 389404,8 руб.
2. <i>Нормы и нормативы расходования ресурсов</i>	Районный коэффициент: 1,3; Коэффициент дополнительной заработной платы: 0,12; Коэффициент накладных расходов: 0,16.
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Коэффициент отчислений на уплату во внебюджетные фонды: 0,3.

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; Определение возможных альтернатив проведения научных исследований.
2. <i>Планирование и формирование бюджета научных исследований</i>	Формирование плана и графика разработки. Формирование бюджета затрат на разработку.
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	Расчет интегрального финансового показателя. Расчет интегрального показателя ресурсоэффективности. Проведение сравнения финансовой эффективности вариантов разработки

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	
---	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШПИБ	Рыжакина Т. Г.	К.Э.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И6Б	Бедер К. Б.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8И6Б	Бедеру Константину Борисовичу

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.04.02 Информационные системы и технологии

Тема ВКР:

Разработка серверного приложения для асинхронного симметричного коммуницирования сетевых клиентов	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объект исследования – веб приложение, автоматизирующее процессы интернет-рекрутинга. Область применения – отделы HR.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	– Организация рабочих мест с электронно-вычислительными машинами регулируется СанПиНом 2.2.2/2.4.1340 – 03 – Рабочее место при выполнении работ сидя регулируется ГОСТом 12.2.032 –78
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	– Отклонение показателей микроклимата – Отсутствие или недостаток естественного света – Недостаточная освещенность рабочей зоны – Повышенный уровень электромагнитных излучений
3. Экологическая безопасность:	Анализ негативного воздействия на окружающую природную среду: утилизация компьютеров, смартфонов, оргтехники
4. Безопасность в чрезвычайных ситуациях:	Возможные чрезвычайные ситуации: Пожар, Землетрясение
Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

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

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И6Б	Бедер Константин Борисович		

РЕФЕРАТ

Выпускная квалификационная работа 77 с., 10 рис., 21 табл., 13 источников, 1 прил.

Ключевые слова: АДМИНИСТРАТОР, УЧАСТНИК, БОТ, БЭКЭНД, ФРОНТЭНД, КЛИЕНТ.

Целью данной работы является разработка программного обеспечения, позволяющего сетевым клиентам симметрично и асинхронно взаимодействовать с сервером, реализация игры «Cybertank» на его основе для проведения хакатона.

Объектом исследования является симметричное взаимодействие в клиент-серверной архитектуре.

Практическая значимость разработки программного проекта для компании «Сибэдж» состоит в том, что его можно использовать для осуществления мероприятий по привлечению новых сотрудников, таких как хакатоны.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных;

ИС – информационная система;

ИТ – информационные технологии;

ПО – программное обеспечение;

API – application programming interface (англ.) – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой;

JSON – JavaScript Object Notation (англ.) – текстовый формат обмена данными, основанный на JavaScript;

XML – расширяемый язык разметки. Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров

HTTP – протокол прикладного уровня передачи данных;

Веб-сервер – сервер, который использует определенный протокол для передачи статического и динамического контента по запросу пользователей;

Аутентификация – процедура сравнения введенных данных с записанными на внешнее хранилище с целью идентификации пользователя;

URL – унифицированный указатель ресурса, определяющий его конкретное местоположение;

HARD-скилы – профессиональные навыки, которым можно научиться и которые можно измерить;

Фреймворк – программное обеспечение, представляющее шаблон для программной платформы и облегчающее процесс разработки;

SOLID – это аббревиатура пяти основных принципов проектирования в объектно-ориентированном программировании – Single responsibility, Open-closed, Liskov substitution, Interface segregation и Dependency inversion;

Тест-кейс – тестовый артефакт.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	12
1 Анализ предметной области	14
1.1 Актуальность разработки ПО, облегчающего отбор персонала	14
1.2 Анализ задачи	14
1.2.1 Игра «Cybertank»	14
1.2.2 Правила игры	15
1.2.3 Режимы игры	17
1.2.4 Правила чемпионатов	18
1.2.5 Начало игры	18
1.2.6 Расстановка	19
1.2.7 Выстрелы.....	19
1.2.8 Результаты.....	20
1.2.9 Описание элементов системы	20
2 Проектирование системы	22
2.1 Требования к разрабатываемой системе	22
2.2 Архитектура системы.....	22
2.3 Конечный автомат	24
2.4 Игровой интерфейс.....	25
3 Программная реализация	28
3.1 Выбор программного обеспечения.....	28
3.2 Формат обмена сообщениями	31
3.3 Реализация контрактов.....	32
3.4 Реализация бизнес логики	33
3.5 Реализация контроллеров	36
3.6 Тестирование приложения.....	38
4 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение ...	43
4.1 Оценка перспективности проведения исследований.....	43
4.1.1 Потенциальные потребители результатов исследования	43
4.1.2 Анализ конкурентных технических решений	43

4.1.3	Технология QuaD	45
4.1.4	SWOT-анализ	46
4.2	Определение возможных альтернатив проведения научных исследований	48
4.2.1	Структура работ в рамках научного исследования	49
4.2.2	Разработка графика проведения научного исследования	50
4.3	Бюджет научно-технического исследования.....	54
4.3.1	Расчет материальных затрат научно-технического исследования....	54
4.3.3	Расчет затрат на специальное оборудование для научных работ	55
4.3.4	Основная заработная плата исполнителей	56
4.3.5	Расчет дополнительной заработной платы.....	57
4.3.6	Отчисления во внебюджетные фонды	58
4.3.7	Накладные расходы.....	58
4.3.8	Формирование бюджета затрат НТИ	59
4.4	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	59
5	Социальная ответственность	63
5.1	Правовые и организационные вопросы обеспечения безопасности.....	64
5.1.1	Организационные мероприятия обеспечения безопасности	64
5.2	Производственная безопасность	67
5.2.1	Анализ опасных и вредных производственных факторов	69
5.2.2	Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов	70
5.3	Экологическая безопасность	71
5.4	Безопасность в чрезвычайных ситуациях	71
5.5	Пожар.....	71
	ЗАКЛЮЧЕНИЕ	74
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	75
	ПРИЛОЖЕНИЕ А Акт о внедрении	77

ВВЕДЕНИЕ

Для многих предприятий проблемой является отбор соискателей. В ИТ-компаниях на каждую вакансию разработчика претендует минимум 15 соискателей, желающих занять это место.

Одним из способов отбора сотрудников является хакатон. Хакатон – это форум разработчиков ПО, посвященный решению какой-либо конкретной проблемы из области ИТ или разработке конкретного продукта. Кроме того, его задачей часто является поиск и выявление перспективных кадров – организаторы и наблюдатели заинтересованы нанимать людей, имевших хорошие способности к программированию и хорошо показавших себя на мероприятии.

У компании «Сибэдж» появилась необходимость в проведении хакатона. Руководством компании было принято решение сделать его в виде онлайн-игры с использованием написанных ботов и их алгоритмов. Игру требовалось сделать максимально гибкой с возможностью модификаций, совершенствования и подмены контекста. Для осуществления данного требования было принято решение разрабатывать игровую логику с использованием конечных автоматов. Так как игра должна была проходить в режиме “онлайн”, то для работы требовалось полноценное веб-приложение, имеющее возможность асинхронного симметричного обмена сообщениями с клиентами, а также с веб интерфейсом для отображения процесса игры и некоторых настроек.

Таким образом, для решения данной задачи требовалось создать платформу, которая позволяла бы провести хакатон с помощью игры «Cybertank».

Целью данной работы является разработка программного обеспечения, позволяющего сетевым клиентам симметрично и асинхронно взаимодействовать с сервером, реализация игры «Cybertank» на его основе для проведения хакатона.

В первом разделе приведены результаты анализа предметной области, задачи и правила реализованной игры.

Во втором разделе описывается процесс проектирования системы, разработку игрового интерфейса и конечного автомата.

В третьем разделе описаны результаты реализации каждого компонента приложения, а также тестирования.

В разделе 4 изложены вопросы, связанные с финансовым менеджментом и ресурсоэффективностью разработки.

В разделе 5 рассмотрены экологическая безопасность и техника безопасности работников на рабочем месте.

Объектом исследования является симметричное взаимодействие в клиент-серверной архитектуре.

Практическая ценность разрабатываемой системы состоит в ее использовании в качестве платформы для проведения хакатонов, что подтверждается актом о внедрении (Приложение А).

Для выполнения работы использовались: операционная система Windows, платформа .NET Core 3 SDK, среда разработки JetBrains Rider, редактор исходного кода Visual Studio Code, браузер Google Chrome, Word для оформления пояснительной записки, Figma для разработки дизайна.

Также для реализации клиентов использовались следующие языки: JavaScript, C#, Java, TypeScript, Python, GoLang.

1 Анализ предметной области

1.1 Актуальность разработки ПО, облегчающего отбор персонала

Отбирая сотрудников, один HR-специалист компании «Сибэдж» тратит на каждого соискателя не менее 2 часов рабочего времени. В первую очередь соискатели оцениваются по так называемым HARD-скилам – профессиональным навыкам.

Для отбора самых компетентных и талантливых разработчиков было решено провести серию тематических хакатонов в виде командной игры, где участникам предстояло решить какую-либо задачу или проблему. Данный формат способен помочь выбрать среди всей массы претендентов самых талантливых разработчиков.

Для проведения хакатона, была необходима такая платформа, в которую была бы возможность дешево и легко вносить изменения, менять контекст и модифицировать под тематику очередного хакатона. Кроме стандартного RestAPI решения, в котором сервер отвечает на HTTP запросы клиентов, сервер должен иметь возможность синхронно обращаться к подключенным клиентам, вызывая процедуры на их стороне и передавать данные.

Для проведения первого хакатона на основе этой платформы предстояло создать игру «Cybertank», победа в которой являлась бы целью для участников хакатона.

1.2 Анализ задачи

1.2.1 Игра «Cybertank»

Командная игра, где задачей является написать такого автоматического клиента (бота) для игры в «Cybertank», чтобы он умел играть в «Cybertank» и как можно чаще побеждал ботов других команд по

правилам «Cybertank». Игра клиент-серверная, поэтому каждая команда должна получить уникальный идентификатор, являющийся ее именем, для возможности подключения к игровому серверу.

Задача каждой команды – разработать алгоритм, который был бы способен с помощью получаемых от сервера команд, рассчитать наилучшую точку для обстрела или составить карту танков. Для реализации алгоритма разрешалось использовать любые вычислительные библиотеки.

1.2.2 Правила игры

Для организации игрового процесса были составлены следующие правила:

- Игра ведется на поле размером 10x10 клеток.
- В одной игре принимают участие всегда два игрока.
- В режиме отладки одновременно может происходить любое количество игр, но один игрок всегда может играть только в одну игру. В режиме боя одновременно происходит только одна игра, которая отображается на экране.
- Каждому игроку дается 10 танков.
- Танки представляют собой отрезки длиной 2 смежные клетки, т.е. танк обязательно линейный.
- Танки не могут касаться друг друга, в том числе по диагонали.
- Игра делится на фазы: размещение танков, бой, подведение итогов.
- В фазе размещения танков необходимо указать координаты всех клеток, занятых танками. Сервер считает, что размещение задано после пересылки определенного правилами количества клеток (20). 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.

- Закончив расстановку сил, игроки начинают «обстрел» друг друга. Первый ход разыгрывается между игроками случайно.
- В фазе выстрелов игроки получают команду от сервера, на которую они отвечают координатами ячейки, в которую они хотели бы выстрелить. Сервер сообщает результаты.

Если выстрел попал в целый танк, то он считается «подбитым» и для его уничтожения нужен еще один точный выстрел по смежной ячейке танка. Игрок может продолжать свой ход.



Рисунок 1 – Схема подбитого танка

Если выстрел попал в «подбитый» танк, то он считается уничтоженным. Игрок может продолжать свой ход.

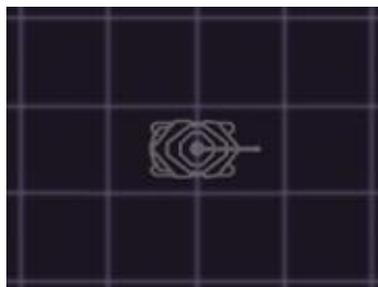


Рисунок 2 – Схема уничтоженного танка

Если выстрел не попал в танк, ход передается другому игроку.



Рисунок 3 – Схема промаха

- Игра завершается, если у одного из игроков не осталось танков. Выигрывает тот игрок, у которого они остались.
- Игра завершается при достижении хода 100 для каждого из участников. Победителем становится игрок с большим количеством клеток с танками. Если клеток поровну, результатом игры является ничья.
- Игра завершается, если один из игроков превысил таймаут (3 сек, действует только для режима игры бой). В этом случае победителем становится другой игрок.

1.2.3 Режимы игры

Предусмотрены следующие варианты игр:

- FIGHT (Бой) – игра, в которой сервер подбирает настоящего противника. В таком режиме действует таймаут (3 секунды) на ответ от игрока, поэтому в него нужно укладываться. В этом режиме присутствует визуализация боя, поэтому игра будет замедлена на этапе выбора сервером очередности хода;
- DEBUG (Отладка) – отладочный режим, когда сервер выдает противника, не являющегося настоящим игроком. В данном режиме таймаут не действует. Виртуальный оппонент будет случайно обстреливать всю площадь поля без повторных попаданий в одни и те же ячейки;

- TEST (Тестовый) – тестовый режим, аналогичен режиму «FIGHT», однако не имеет таймаута и после завершения игры, запись результатов не происходит. Визуализация игры также присутствует.

1.2.4 Правила чемпионатов

Для выявления победителя были составлены следующие правила:

- Чемпионат объявляется организаторами.
- Чемпионаты проводятся по схеме, когда каждый участник играет с каждым другим участником одну игру.
- Участвуют в чемпионате все, кто подключится к серверу и попытается начать игру в режиме «Бой».
- Сервер случайно выбирает соперника.
- За победу начисляется 3 балла. Ничья – 1 балл. Проигрыш – 0 баллов.
- Если в конце всех игр, несколько участников набрали для победы равное количество баллов, объявляется дополнительный бой. Количество дополнительных боев ограничено и равно трем. Если по истечению трех дополнительных игр, счет остается равным, победитель определяется случайно.

1.2.5 Начало игры

Для начала игры нужно игроку послать «start: <режим игры>». Как только сервер подберет вам противника (зависит от режима игры), в ответ придет сообщение «prepare!», которое означает, что необходимо переходить к фазе расстановки.

1.2.6 Расстановка

Начинается после получения от сервера события «prepare!» и продолжается до тех пор, пока на сервер не уйдут координаты всех ячеек с танками (20 ячеек).

Необходимо послать координаты всех клеток с танками в виде «<x>,<y>;<x>,<y>;...», то есть, например, «1,7;1,8;2,3;...».

Порядок пересылки координат не имеет значения.

Если расстановка по каким-то причинам не является верной, например:

- количество танков не соответствует правилам;
- танки касаются друг друга;
- часть координат недопустима;
- нарушен формат сообщения.

В этом случае сервер сам расставит танки за игрока случайным образом.

При этом будет выслано сообщение: «Info: Your tank setup would be random».

Фаза заканчивается получением команды «fire!», которая означает, что нужно выполнить свой ход и выстрел.

1.2.7 Выстрелы

После получения команды «FIRE», нужно сделать выстрел.

Необходимо послать координаты на сервер в виде «<x>,<y>», то есть, например, «1,7». В ответ придет сообщение вида: «fire result: <result>: <x>,<y>». Результаты выстрела:

- MISS – промах. В случае промаха «fire result: MISS: 1,7», ход переходит к следующему игроку. Если выстрел произведен по ранее обстрелянной клетке, результатом является промах;

- WOUNDED – ранил. В случае попадания по целому танку «fire result: HIT: 1,7», ход продолжается;
 - KILL – убил. В случае попадания по «подбитому» танку «fire result: KILL: 1,7», ход продолжается.
- Ошибочные результаты выстрела:
- ERROR_COORD – неправильные координаты (вне границ игрового поля);
 - ERROR_NOT_YOUR_TURN – у вас нет выстрелов (не ваш ход);
 - ERROR_TIMEOUT – превышен таймаут.

1.2.8 Результаты

После окончания игры каждый игрок получает сообщение «winner: <player>», где указывается победитель игры.

Чтобы начать новую игру, нужно послать новое сообщение «start: ...», иначе игрок больше не будет участвовать в играх.

Если игра закончилась по тайм-ауту, то придет сообщение «Timeout detected: your session <id> expired». Здесь <id> соответствует идентификатору игры. После этого сообщения все равно оба игрока получают сообщение «winner: <player>», поэтому обычно нет нужды как-то особо обрабатывать сообщение о тайм-ауте.

1.2.9 Описание элементов системы

В таблице 1 представлено описание элементов системы.

Таблица 1 – Описание элементов схемы

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

3	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 по заданному в ней алгоритму.
5	Фронтэнд с визуальной частью игрового поля	Клиентская часть приложения для визуализации игр бот-клиентов.

Также, по правилам игры, к серверу имеют возможность подключаться клиенты основных технологических стеков:

- JavaScript;
- C#;
- Java;
- TypeScript;
- Python;
- GoLang.

2 Проектирование системы

2.1 Требования к разрабатываемой системе

После анализа задачи к разрабатываемой системе были представлены следующие требования:

- система должна быть слабосвязанной и понятной для программиста;
- система должна позволять программисту легко подменять контекст на новый или модифицировать имеющийся;
- система должна уметь взаимодействовать с подключенными клиентами, самостоятельно отправляя им сообщения, не дожидаясь запроса;
- система должна уметь идентифицировать клиента и хранить информацию о подключившихся;
- система должна иметь базу данных для хранения и записи игровых результатов;
- система должна иметь панель администрирования;
- система должна иметь возможность обнуления всех пользовательских результатов;
- система должна иметь возможность разрешать и запрещать игры.

2.2 Архитектура системы

Система состоит из 3 элементов: сервер, веб-сайт и клиент. Самой главной частью является сервер, он должен содержать в себе всю игровую логику, аутентифицировать клиентов, поддерживать подключение с БД и иметь API для подключения веб-сайта и клиентов.

Задачами веб-сайта являются:

- Подключение к серверу;
- Получение статистики;
- Отображение статистики;

- Отображение панели администрирования;
- Отправка команд для разрешения или запрета проведения игры;
- Отправка команд для сброса статистики.

Бот-клиентом является приложение, разработанное участниками хакатона, способное участвовать в хакатоне по правилам «Cybertank».

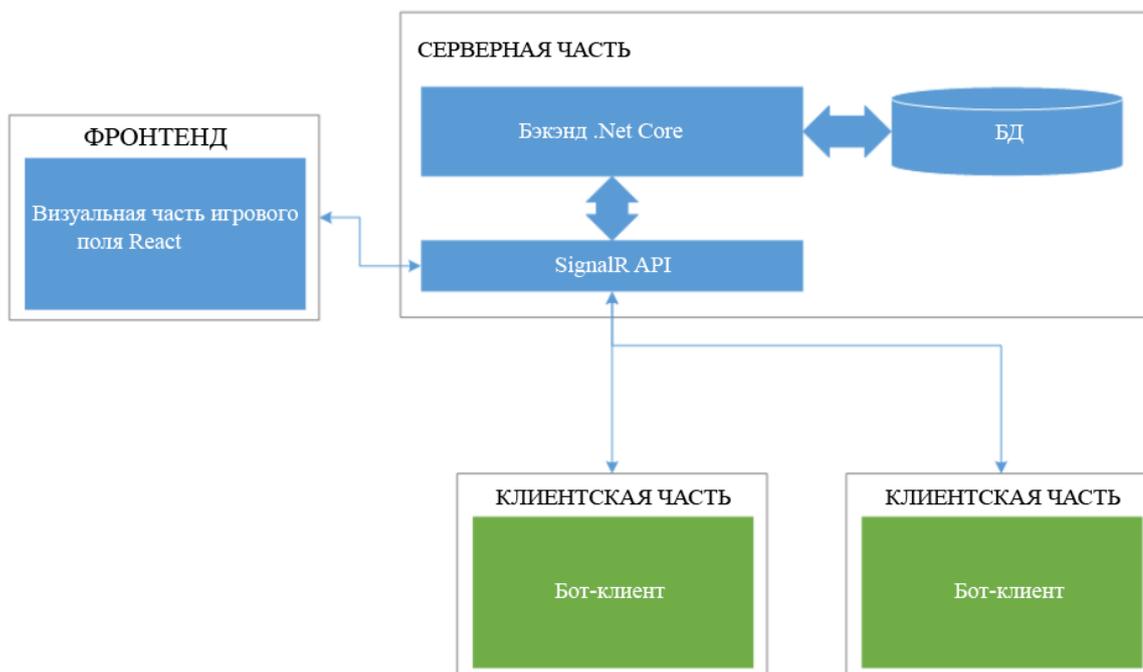


Рисунок 4 – Архитектура описанной системы

Кроме архитектуры системы, требовалось определиться с архитектурой приложения: используемые компоненты, библиотеки, модели, сервисы и уровни абстракций.

Так, для работы приложения требовалось создать необходимые контракты для обмена сообщениями, сервисы, которые бы манипулировали этими данными, контроллеры и методы контроллеров для взаимодействия с клиентами.

Для визуального отображения хода игры был выбрана JS-фреймворк ReactJS [1] так как данный фреймворк отлично подходит для создания веб-приложений, где данные могут меняться на регулярной основе, а также имеет низкий порог вхождения, высокий уровень гибкости, максимальную отзывчивость и является самым популярным фреймворком на рынке ПО.

Архитектура включала базу данных. Для данной работы была выбрана база данных Microsoft SQL Server [2], так как ORM EntityFramework из стандартного пакета .NET Core 3 [3] имеет встроенную поддержку Microsoft SQL Server без подключения дополнительных библиотек, а также начиная с версии 2017 у Microsoft SQL Server появилась возможность развертывания базы в Docker-контейнере [4] под управлением Linux.

База данных для хранения результата была развернута в отдельном Docker-контейнере на случай непредвиденных ситуаций, когда основное приложение заикливается или перестает отвечать. В этой аварийной ситуации, бэкэнд мог быть жестко перезагружен с помощью портейнера [5], а БД могла продолжить свою работу, храня все записи независимо от основного приложения.

2.3 Конечный автомат

Для работы игрового движка требовалось реализовать алгоритм с помощью стейт машины, алгоритм функционирования которой представлен на рисунке 5. Алгоритм должен был изменять состояние автомата основываясь на трех командах: «prepare», «next», «finish». Так, после того как бот подключился к системе, система создает новый объект «NetworkPlayer» в пуле игроков, задает объекту свойства в виде имени, идентификатора и пустого игрового поля и добавляет в потокозащищенный список игроков «ConcurrentBag<IPlayer> Players» [6]. После того как к игре подключается второй игрок, состояние системы переходит в режим «arrangement» и клиентам отправляется запрос на начало игры, при этом с помощью команды «next» система переходит в новое состояние.

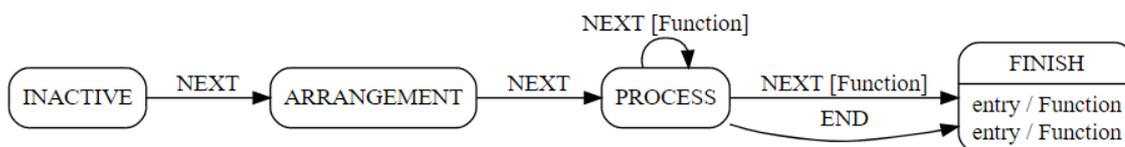


Рисунок 5 – Алгоритм работы конечного автомата

Таким образом с помощью библиотеки Stateless был разработан конечный автомат для шаговой игры, который будет взят за основу игрового движка.

2.4 Игровой интерфейс

Так как игра не требовала наличия элементов управления из UI-интерфейса, процесс проектирования интерфейса был достаточно простым. Согласно правилам игры, каждый игрок имеет собственное поле 10x10 клеток, шкалу здоровья и идентификатор в виде имени игрока или команды.

Игрок 1	0	1	2	3	4	5	6	7	8	9
0	■	■		■						■
1			■		■	■				
2										
3										
4	■	■				■				■
5						■				
6										
7										
8	■									
9	■			■	■					■

Рисунок 6 – Эскиз игрового поля

На рисунке 6 изображен эскиз игрового поля, в котором каждая клетка является элементом матрицы 10x10. В случаях, когда две соседние клетки закрашены, игра считает их заполненными.



Рисунок 7 – Процесс создания дизайна игры

На рисунке 7 изображена одна из первых версий дизайна игры. За время разработки проекта было рассмотрено несколько схожих вариантов дизайна (рис. 8). После выбора, отработки и прорисовки дизайна был утвержден единый, который и был использован в конечном продукте.



Рисунок 8 – Варианты дизайна игровых объектов

На рисунке 9 представлен конечный вариант исполнения дизайна интерфейса.



Рисунок 9 – Конечный вариант исполнения дизайна игры

Таким образом был разработан игровой интерфейс. В него включены поля для обоих противников, шкалу их здоровья и идентификатор. При очередном ходе игрока его поле визуально подсвечивается. Точка, куда игрок делает ход также подсвечивается на поле противника.

3 Программная реализация

3.1 Выбор программного обеспечения

Для реализации сервера необходима платформа с фреймворками, поддерживающий современные протоколы и библиотеки для создания и конфигурирования веб-серверов.

Для создания платформы были рассмотрены следующие языки программирования: Java, C#, JavaScript, Python и Go.

Java – это объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems. Программы на Java транслируются в байт-код, который затем выполняется виртуальной машиной Java (JVM). JVM – это программа, которая обрабатывает байтовый код и передает инструкции оборудованию как интерпретатор. Достоинством подобной реализации является независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует JVM.

Другой важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

C# является языком со статической типизацией, поддерживающий полиморфизм, перегрузку операторов и имеющих огромное количество библиотек для реализации веб-серверов. Этот объектно-ориентированный язык программирования, разработан в 1998-2001 годах группой инженеров компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку

операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Стандартная библиотека включает большой объём полезных функций. Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Go – компилируемый многопоточный язык программирования, разработанный внутри компании Google. Язык Go разрабатывался как язык программирования для создания высокоэффективных программ, работающих на современных распределённых системах и многоядерных процессорах. Он может рассматриваться как попытка создать замену языкам C и C++ с учётом изменившихся компьютерных технологий и накопленного

опыта разработки крупных систем. Go был разработан для решения реальных проблем, возникающих при разработке программного обеспечения в Google. Язык создавался в расчёте на то, что программы на нём будут транслироваться в объектный код и исполняться непосредственно, не требуя виртуальной машины, поэтому одним из критериев выбора архитектурных решений была возможность обеспечить быструю компиляцию в эффективный объектный код и отсутствие чрезмерных требований к динамической поддержке.

В качестве языка для веб-сервера был выбран язык C#, потому что платформа .NET Core имеет встроенную библиотеку для создания симметричного соединения в серверах – SignalR [7], а также загружаемую библиотеку Stateless [8], позволяющую сконфигурировать стейт-машину.

SignalR представляет библиотеку от компании Microsoft, которая предназначена для создания приложений, работающих в режиме реального времени. В частности, ее можно использовать вместе с ASP.NET Core. SignalR использует двунаправленную связь для обмена сообщениями между клиентом и сервером, благодаря чему сервер может отправлять в режиме реального времени всем клиентам некоторые данные. Исходя из возможностей клиента и сервера инфраструктура SignalR выбирает наилучший механизм для взаимодействия. В частности, наиболее оптимальным является WebSockets, соответственно если и клиент, и сервер позволяют использовать этот механизм, то взаимодействие идет через WebSockets. Однако если WebSockets не поддерживается, то применяется Server-Side Events. И если SSE не поддерживается, то применяется Long Polling.

«Stateless» – это библиотека для создания конечных автоматов на C#. Stateless позволяет описывать высокоуровневые машины состояния для бизнеса, а также предоставляет инструменты для легкого моделирования состояний устройства IoT, пользовательских интерфейсов и т.д. Более того, Stateless также сериализует разработанный конечный автомат в стандартный

текстовый формат «DOT Graph», который затем можно сгенерировать в SVG или PNG для отображения разработанного графа создаваемой машины состояний.

3.2 Формат обмена сообщениями

Наиболее распространенными и поддерживаемыми программными средствами форматами данных являются JSON и XML. Несмотря на гибкость разрабатываемого решения и независимость работы серверной части от конкретного формата данных, необходимо выбрать формат, поддержку которого следует организовать в первую очередь. В таблице 2 указаны критерии, по которым проводилось сравнение JSON и XML.

Таблица 2 – Критериальное оценивание различных форматов передачи и хранения данных

Критерий/Формат данных	JSON	XML
Избыточная информация (7)	6 – необходимость использования кавычек	2 – название каждого атрибута дублируется
Поддерживаемые архитектурные стили (5)	3 – только REST	5 – REST, SOA
Степень поддержки JavaScript'ом (6)	6 – полностью поддерживается	3 – требуется использовать сторонние библиотеки
Удобство представления для пользователя (3)	2 – легко читается, но требуется знание JavaScript	1 – множество тегов, требуется знание HTML
ИТОГО	17	11

JSON обладает лучшей поддержкой и не требует установки дополнительных программных средств или какой-либо настройки, это формат по умолчанию для многих сервисов, поэтому было принято решение использовать JSON в качестве основного формата данных.

3.3 Реализация контрактов

Модель осуществляет соединение с источником данных, предоставляя необходимый интерфейс другим компонентам архитектуры. Характер связи зависит от типа источника данных и способа реализации модели.

После того, как архитектура была описана, встал вопрос о создании контрактов игры. Так, для обеспечения взаимодействия с клиентами-игроками были созданы следующие модели:

- ArrangementResponse – ответ игроку на расстановку;
- FinishResponse – ответ игроку с результатами игры;
- ShotResponse – ответ игроку на выстрел;
- Response – базовый ответ, от которого были унаследованы остальные ответы.

Для взаимодействия сервера с веб-клиентом, отображающим ход игры, были созданы следующие модели:

- OnStartMessage – сообщение, отправляемое веб-клиенту (браузеру), сообщающее о начале игры;
- OnStepMessage – сообщение, отправляемое веб-клиенту, сообщающее о результатах сделанного хода;
- OnFinishMessage – сообщение, отправляемое веб-клиенту, сообщающее о результатах сыгранного матча (имя победителя, тип победы).

После того, как контракты были созданы, было необходимо создать необходимые модели для игрового взаимодействия. Так, в папке «Models» (рис. 3) основного проекта расположились следующие модели:

- Administrator – модель администратора игры, для управления игрой (разрешить, отменить);
- CheckOfStep – модель проверенного хода, содержит статус хода и Response-ответ из контрактов;

- IPlayer – интерфейс игрока;
- NetworkPlayer – модель игрока, подключенного по SignalR;
- NpcPlayer – модель бота, реализованного на сервере (искусственный бот-противник);
- Score – модель набранных очков для хранения в БД (содержит поля «Name» и «Points»);
- ScoresContext – модель контекста для подключения к БД;
- TypeOfGame – enum-класс типа игры (Debug, Fight, Test);
- Watcher – модель подключаемого веб-клиента, для просмотра хода игры.

3.4 Реализация бизнес логики

Так как у приложения уже появились модели для манипуляции, следующим логичным шагом стало создание инструментов (сервисов) для манипуляции объектами-моделями. В папке Services были созданы следующие классы:

- Manager – сервис для управления всеми играми, обрабатывающий и распределяющий все входящие от игроков сообщения. Класс содержит методы по управлению всеми игроками и всеми играми. Каждая начатая игра инициализируется в классе Manager и удаляется, при завершении, в нем же. Также Manager хранит объекты IPlayer, инициализируя настоящего сетевого противника или искусственного противника при создании новой игры;
- MatrixChecker – сервис проверки матриц-полей, присылаемых игроками. Проверяет валидность и правильность заполнения согласно правилам игры;

Для проверки матрицы класс использует обход графа в глубину, а также проверяет размерность, расстановку, артефакты и, в случае если матрица прошла все проверки, конвертирует ее в понятный игровому

- движку формат и возвращает туда, где проверенная матрица будет применена как игровое поле;
- StepChecker – сервис для проверки ходов, присылаемых игроками. Содержит методы для проверки валидности и результатов хода игрока. Процесс проверки проходит в несколько этапов: сначала входящие данные проверяются на выход точки за пределы поля, затем при помощи поля противника и входящей точки выстрела сервис определяет результат выстрела. Обработав точку выстрела, сервис высчитывает результат выстрела в зависимости от заполненности координаты на поле и генерирует ответ для каждого из состояний, заменяя статус точки выстрела на игровом поле и возвращая ответ для асинхронной отправки ходившему игроку;
 - PlayerFactory – сервис-фабрика для инициализации игроков по протоколу SignalR, в случае подключения по сети, и инициализации объекта-бота в случае необходимости в боте. Сервис имеет 3 перегруженных метода «GetPlayer», которые инициализируют нового игрока в виде интерфейса «IPlayer» в зависимости от передаваемых параметров. Так, метод «GetPlayer» без параметров инициализирует серверного бота для режима отладки. «GetPlayer» с передаваемыми параметрами “string name” и “ string id” инициализирует нового сетевого клиента, присвоив ему имя и идентификатор подключения соответственно. Кроме имени и идентификатора, передавая “int timeout” в качестве третьего параметра «GetPlayer» инициализирует нового сетевого игрока с именем, идентификатором и фиксированным таймаутом соответственно;
 - PullSender – сервис для отправки пула (набора данных) игры. Представляет собой класс, который позволяет на протяжении всего жизненного цикла каждой игры отправлять всю игровую статистику подключаемым к веб-сайту клиентам. Клиенты, в свою очередь, получают возможность просмотра всей игры независимо от того, в

- какой момент игры они подключились, присылаемый игровой пулл содержит историю каждого хода для обоих игроков;
- PullSenderFactory – сервис-фабрика для инициализации сервиса «PullSender» в зависимости от типа игры. Для обеспечения меньшей связанности приложения, бизнес-логика разрабатывалась так, чтобы основная логика была максимально независима от реализации всех входящих и исходящих сообщений, следовательно для основной логики нет разницы, в каком сейчас режиме проходит игра, основная логика манипулирует только лишь абстракциями, а логика в абстракциях зависит от реализации, которую определяют Factory-методы. Таким образом PullSenderFactory, принимая в своем методе параметр типа игры, создает игровую комнату с возможностью отображения в браузере или возвращает null в тестовом режиме;
 - RatingService – сервис для чтения и записи рекордов. Данный сервис работает непосредственной с базой данных. После того как игра в режиме Fight подошла к концу, сервис автоматически обращается к базе, загружая ранние результаты соперников и, в случае их присутствия, добавляет выигранные очки, или создает новые сущности и сохраняет изменения. Также, в сервис заложен функционал по просмотру или очистке базы данных от результатов по запросу от администратора игры. При отображении турнирной доски фронтенд-клиент делает запрос для получения результатов в эндпоинт «ManagementHub», который возвращает результат в виде ответа именно этого сервиса и метода «GetRating»;
 - TankPlacer – сервис для генерирования случайного размещения танков для модели «NpcPlayer», также используется в тех случаях, когда присланная матрица игрока невалидна или не соответствует правилам игры, «TankPlacer» генерирует случайную, но валидную матрицу. TankPlacer имеет единственный метод Place, который считывая правила игры «Cybertank» генерирует определенное

- количество танков, установленное правилами игры, определяет поворот каждого танка и случайно расставляет, используя проверки валидности матрицы согласно правилам игры;
- FSM (Commands, Game, States) – State-машина, содержащая в себе основную логику игры. Где класс «Commands» содержит все необходимые команды для State-машины «Game», а «States», соответственно, содержит состояния этой машины;
 - Extensions – вспомогательный класс, содержащий методы-расширения для некоторых моделей, для упрощения разработки.

3.5 Реализация контроллеров

Для взаимодействия с подключаемыми клиентами использовалась библиотека SignalR, которая работает по протоколу WebSocket, что позволяет серверу вызывать процедуры как на подключенном клиенте, так и на группе клиентов. Для подключения клиента к серверу, на сервере должен быть создан один или несколько SignalR-контроллеров, которые содержат вызываемые клиентом методы. Каждый контроллер получает уникальный адрес, обращаясь к которому, клиент имеет возможность вызвать содержащуюся в нем процедуру, указав название метода и передав нужное количество аргументов определенного типа, определенного в сигнатуре метода. При первом подключении клиента, SignalR генерирует для нового клиента уникальный идентификатор, по которому система получает возможность обращаться к клиенту, вызывая процедуры уже на его стороне. Таким образом было создано 2 контроллера: GameHub для соединения сервера с игроками и ManagementHub для соединения сервера с веб-сайтом для визуализации и управления играми.

Контроллер GameHub является точкой коммуницирования сетевых игроков. Для того чтобы подключиться к игре, программа игрока должна выполнить SignalR запрос по URL:

«<https://cybertank.sibedge.com:5001/gameHub>», а в качестве параметров в теле запроса передать название метода и свое опознавательное имя. Так, для режима отладки должен быть вызван метод «Debug», «Test» – для тестового режима и «Fight» для режима чемпионата.

Кроме методов для соединения с игрой контроллер имеет дополнительные методы «ReceiveArrangement» и «ReceiveStep» для взаимодействия с игроком, которые вызываются одинаково, независимо от режима.

ReceiveArrangement – метод, принимающий поле игрока в виде заполненной матрицы 10*10 точек, сериализованной в строку формата string по стандарту JSON.

ReceiveStep – метод, предназначенный для передачи хода игрока, принимающий 2 переменные формата «integer», которые являются «x» и «y» координатами соответственно.

Методы расстановки и хода не требуют повторной передачи имени игрока, т.к. системе уже известен уникальный идентификатор подключения каждого клиента в рамках текущей сессии, с помощью чего система однозначно определяет кем был сделан ход.

Контроллер «ManagementHub» является точкой коммунцирования веб-сайта с системой для просмотра и управления играми. ManagementHub имеет несколько методов для взаимодействия с клиентом:

- «GetRating» – метод для получения рейтинга всех игравших игроков в режиме турнира;
- «ResetRating» – метод для обнуления рейтинга всех игравших игроков в режиме турнира;
- «Watch» – метод для подключения к визуализации игр в режиме турнира;
- «WatchTest» – метод для подключения к визуализации игр в режиме тест;

- «Control» – метод для подключения веб-клиента к панели администрирования играми;
- «AcceptFightGame» – метод администрирования для подтверждения предстоящей игры.
- «DeclineFightGame» – метод администрирования для отклонения предстоящей игры.

Методы управления «AcceptFightGame» и «DeclineFightGame» необходимы для контроля администраторами проведения несанкционированных игр в режиме турнира, так администратор имеет возможность подтвердить игру или отклонить, если, например, имя участника не соответствует требованиям.

3.6 Тестирование приложения

Тестирование является неотъемлемой частью любой разработки. Любой условный оператор в программе может содержать баг и привести к фатальной ошибке приложения. После того как основная часть была написана, параллельно с разработкой началось активное тестирование запросов и логики разрабатываемого приложения.

Так как игра является состязательной, требовалось найти и устранить все возможные баги и отследить все возможные варианты поведения приложения при всех возможных условиях.

Для полного и точного тестирования функционала приложения применялся подход с использованием регрессионного тестирования.

Регрессионное тестирование – это набор тестов, направленных на обнаружение дефектов в уже протестированных участках приложения. Делается это совсем не для того, чтобы окончательно убедиться в отсутствии багов, а для поиска и исправления регрессионных ошибок. Регрессионные ошибки – те же баги, но появляются они не при написании программы, а при добавлении в существующую сборку нового функционала или исправлении

других багов, что и стало причиной возникновения новых дефектов в уже протестированном продукте.

Таким образом, цель регрессионного тестирования – убедиться, что исправление одних багов не стало причиной возникновения других и что обновление сборки не создало новых дефектов в уже проверенном коде.

Тест-кейс – это тестовый артефакт, суть которого заключается в выполнении некоторого количества действий и/или условий, необходимых для проверки определенной функциональности разрабатываемой программной системы. Для проведения процесса тестирования было разработано 33 тест-кейса, в таблицах 3-4 представлены некоторые из них.

Таблица 3 – Тест-кейс на проверку интерфейса

Описание	Проверка интерфейса (проверка появления экрана победы при уничтожении всех танков противника)
Сценарий теста	От лица двух игроков присоединиться к игре в режиме fight; передать серверу данные о расстановке танков; от лица активного игрока расстрелять все танки противника; удостовериться, что после этого на фронт-энд будет выведен экран победы активного игрока
#	Тестовые данные
1	Запрос, содержащий "start: fight"
2	Запрос на расстановку танков для игрока №1 с координатами в формате <x,y> (1,1;1,5;1,9;1,10;2,1;2,5;4,1;4,2;4,10;5,10;6,2;6,5;6,6;7,2;9,10;10,1;10,2;10,5;10,6;10,10)
3	Запрос на расстановку танков для игрока №2 с координатами в формате <x,y> (1,1;1,2;1,5;1,6;1,10;2,10;4,1;5,1;5,6;5,7;5,10;6,4;6,10;7,4;9,1;10,1;10,4;10,5;10,9;10,10)
4	Координаты пустой клетки в поле игрока №1 в формате <x,y> (5,5)
5	Координаты пустой клетки в поле игрока №2 в формате <x,y> (5,5)
Шаг	Ожидаемый результат
1 Отправить серверу запрос "start: fight" или "start: test" от лица игрока №1	Сервер принял запрос; Сервер ожидает данные от второго игрока
2 Отправить серверу запрос "start: fight" или	Сервер принял запрос; Сервер отправил сообщение "prepare!" обоим игрокам.

"start: test" от лица игрока №2	
3 Отправить серверу данные о расстановке танков игрока №1	Сервер принял запрос; Сервер ожидает данные от второго игрока
4 Отправить серверу данные о расстановке танков игрока №2	Сервер принял запрос; Сервер отправил сообщение "Fire!" игроку, чей ход будет первым.
5 Если система дала первый ход игроку №1, см.п.5.1, 6.1	
5 Если игроку №2, см.п.5.2, 6.2	
6 От имени игрока №1 отправить серверу координаты неповреждённого танка игрока №2	Сервер принял запрос; Получено сообщение "Fire result: HIT: <x,y>"; Здоровье противника уменьшилось на 1; Ход продолжился.
7 Повторить шаг 5.1 со всеми координатами расстановки танков игрока №2	Сервер принял запрос; Получено сообщение "Fire result: HIT: <x,y>" или "Fire result: KILL: <x,y>" (в зависимости от того, ранен или уничтожен танк противника); После передачи последних координат здоровье противника уменьшилось до 0; Затем, на экране фронт-энда появилось оповещение о победе игрока №1.
8 От имени игрока №2 отправить серверу координаты неповреждённого танка игрока №1	Сервер принял запрос; Получено сообщение "Fire result: HIT: <x,y>"; Здоровье противника уменьшилось на 1; Ход продолжился.

9 Повторить шаг 5.2 со всеми координатами расстановки танков игрока №1	Сервер принял запрос; Получено сообщение "Fire result: HIT: <x,y>" или "Fire result: KILL: <x,y>" (в зависимости от того, ранен или уничтожен танк противника); После передачи последних координат здоровье противника уменьшилось до 0; Затем, на экране фронт-энда появилось оповещение о победе игрока №2.
--	--

Таблица 4 – Тест-кейс на проверку механизма стрельбы

Описание	Проверка работы механизма стрельбы (некорректные координаты)
Сценарий теста	Послать серверу некорректные координаты клетки, получить сообщение об ошибке; ход переходит противнику
#	Тестовые данные
1	Запрос, содержащий "start: fight"
2	Запрос на расстановку танков для игрока №1 с координатами в формате <x,y> [1, 1, 0, 1, 0, 0, 0, 0, 0, 1], [0, 0, 0, 1, 0, 1, 1, 0, 0, 1], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 1, 0, 0, 0, 1, 0, 0, 0, 1], [0, 0, 0, 0, 0, 1, 0, 0, 0, 1], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0], [1, 0, 0, 1, 1, 0, 0, 0, 1, 1]
3	Запрос на расстановку танков для игрока №2 с координатами в формате <x,y> [1, 0, 0, 1, 1, 0, 0, 0, 1, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 1, 1, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 1, 0, 0, 1, 1, 0, 0, 0, 1]
4	Координаты для выстрела с одним из значений больше 10 (10,2)
Шаг	Ожидаемый результат
1 Отправить серверу запрос "start: fight" или "start: test" от лица игрока №1	Сервер принял запрос; Сервер ожидает данные от второго игрока

2 Отправить серверу запрос "start: fight" или "start: test" от лица игрока №2	Сервер принял запрос; Сервер запросил расстановку от обоих игроков.
3 Отправить серверу данные о расстановке танков игрока №1	Сервер принял запрос; Сервер ожидает данные от второго игрока
4 Отправить серверу данные о расстановке танков игрока №2	Сервер принял запрос; Сервер получил расстановку.
5 От лица игрока активного игрока отправить некорректные координаты	Получено сообщение "Over_border" (неправильные координаты); Выстрел не производится; Ход переходит противнику.

Тестирование приложения с помощью метода регрессионного тестирования позволило создать приложение с минимальным количеством багов. За время проведения мероприятия не было выявлено ни одного бага, что свидетельствует о качестве разработанного приложения и тестирования.

4 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

4.1 Оценка перспективности проведения исследований

4.1.1 Потенциальные потребители результатов исследования

В современном мире информационных технологий многие предприятия сталкиваются с проблемой отбора соискателей. В ИТ-компаниях на каждую вакансию разработчика существует минимум 15 соискателей, желающих занять это место.

Одним из способов отбора сотрудников является хакатон. Хакатон – это форум разработчиков ПО, посвященный решению какой-либо конкретной проблемы из области ИТ или разработке конкретного продукта. Кроме того, его задачей часто является поиск и выявление перспективных кадров - организаторы и наблюдатели часто нанимают людей, имевших хорошие способности к программированию и хорошо показавших себя на мероприятии.

У компании Сибэжд появилась необходимость в проведении хакатона. Руководством компании было принято решение сделать его в виде онлайн-игры с использованием написанных ботов и их алгоритмов. Игру требовалось сделать максимально гибкой с возможностью модификаций, совершенствования и подмены контекста. Для осуществления данного требования было принято решение разрабатывать игровую логику с использованием конечных автоматов.

4.1.2 Анализ конкурентных технических решений

Процесс трудоустройства затрагивает большинство жителей всех стран мира, поэтому на момент написания работы существует множество уже реализованных и функционирующих интернет-приложений, позволяющих проводить такого рода хакатоны. В виду того, что разрабатываемую систему

предлагается внедрить в HR программы компании Сибэдж, разработка должна была быть собственной. Из имеющихся разработок можно выделить «Custis» (к1), «Business Game Clash» (к2) и «DevTime» (к3).

Таблица 5 – Оценочная карта для сравнения конкурентных разработок

Критерии оценки	Вес критерия	Баллы				Конкурентоспособность			
		Бф	Бк1	Бк2	Бк3	Кф	Кк1	Кк2	Кк3
1	2	3	4	5	6	7	8	9	10
Технические критерии оценки ресурсоэффективности									
1. Простота использования системы	0,10	5	5	5	3	0,50	0,50	0,50	0,30
2. Удобный интерфейс пользователя	0,10	4	4	4	4	0,40	0,40	0,40	0,40
1	2	3	4	5	6	7	8	9	10
3. Функциональные возможности	0,30	4	1	2	5	1,20	0,30	0,60	1,50
4. Скорость обработки данных	0,08	4	4	4	4	0,32	0,32	0,32	0,32
5. Потребность в ресурсах памяти	0,08	4	5	5	4	0,32	0,40	0,40	0,32
Экономические критерии оценки эффективности									
6. Цена	0,20	5	5	5	3	1,00	1,00	1,00	0,60
7. Послепродажное обслуживание	0,05	4	4	4	5	0,20	0,20	0,20	0,25
8. Предполагаемый срок эксплуатации	0,09	5	5	5	5	0,45	0,45	0,45	0,45
Итого	1					4,74	3,57	3,87	4,14

Исходя из полученных результатов, оценка основных технических и экономических характеристик конкурентных программных решений показывает, что разрабатываемое веб-приложение является конкурентоспособным по сравнению с рассмотренными аналогами.

Основным недостатками конкурентных программных продуктов являются ограниченность функционала или сложность использования из-за большой универсальности функционала в случае с DevTime.

4.1.3 Технология QuaD

Для оценки качества разработки и ее перспективности на рынке была построена оценочная карта конкурентных программных решений по технологии QuaD с учетом технических и экономических особенностей этой разработки. Данная оценочная карта представлена в таблице 6.

Таблица 6 – Оценочная карта для сравнения конкурентных разработок по технологии QuaD

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение	Средневзвешенное значение
1	2	3	4	5	6
Показатели оценки качества разработки					
1. Простота использования системы	0,10	90	100	0,9	0,090
2. Удобный интерфейс пользователя	0,10	70	100	0,7	0,070
3. Функциональные возможности	0,30	80	100	0,8	0,240
4. Скорость обработки данных	0,08	70	100	0,7	0,056
5. Потребность в ресурсах памяти	0,08	70	100	0,7	0,056
Показатели оценки коммерческого потенциала разработки					
6. Цена	0,20	100	100	1,0	0,200
7. Послепродажное обслуживание	0,05	80	100	0,8	0,040
8. Предполагаемый срок эксплуатации	0,09	80	100	0,8	0,072
Итого	1				0,824

Значение средневзвешенного показателя качества и перспективности научной разработки равно 82,4, а это значит, что разработка является перспективной.

4.1.4 SWOT-анализ

Чтобы выявить достоинства и недостатки продукта с целью дальнейшего улучшения качества необходимо провести SWOT анализ. SWOT-анализ (показан в таблице 7) – один из методов стратегического планирования, является простым и качественным инструментом оценивания конкурентоспособности. Суть метода заключается в выявлении ключевых факторов внутренней и внешней среды.

В названии заложены первые буквы четырех категорий, из которых складываются факторы – Strength (сильные стороны продукта), Weaknesses (слабые стороны), Opportunities (возможности) и 122 Threats (угрозы). Дополнительно все факторы следует проанализировать в парах, например слабые стороны и угрозы.

Таблица 7 – SWOT-анализ разрабатываемого продукта

		Внутренние факторы	
Внешние факторы		Сильные стороны 1. Отказоустойчивость системы из-за микросервисной архитектуры. 2. Балансировка нагрузки для оптимизации работы сервера. 3. Модульность клиентского интерфейса. 4. Технология надежного хранения и логирования всех событий системы. 5. Дополнительные возможности для молодых специалистов и учащихся	Слабые стороны 1. Система молодая, находится на этапе становления и развития. 2. Использование серверов начального уровня производительности. 3. Медленный старт из-за ограниченности рекламного бюджета.
	Возможности 1. Рост производительности вычислительной техники. 2. Попадание продукта на первые строчки выдачи поисковых систем. 3. Создание релевантных	Сильные стороны и возможности показывают, что исходной производительности сервера хватит с запасом на начальный этап, модульность системы позволит предлагать	Исходя из слабых сторон и возможностей, можно предположить экспоненциальный рост числа пользователей, наблюдается сильная зависимость развития от первоначальной активности, приложение

	систем, интегрируемых с продуктом.	пользователям самый современный функционал.	может завязнуть на начальном этапе.
	Угрозы 1. Подъем пользовательского спроса на приложения, запускаемых в ОС, а не в браузере. 2. Массированные атаки, направленные на отказ в обслуживании. 3. Появление более совершенного конкурентного продукта.	Полный отказ системы почти невозможен, но намеренный вывод из строя критичных частей может вызвать серьезное недовольство пользователей, также тенденции к использованию выделенных приложений могут снизить их активность.	Сервер может зависнуть, в таком случае приходится перезагружать контейнеры и восстанавливать все соединения.

Самой слабой стороной продукта является отсутствие выделенного приложения и нативной панели управления, программная система запускается в браузере с помощью веб-клиента. Для создания таких приложений в проект необходимо привлечь несколько разработчиков, однако это потребует пересмотра бюджета проекта.

Если популярность, а вместе с ней и сетевая нагрузка, продукта в некоторый момент времени стремительно возрастет, потребуется в срочном порядке улучшить серверный парк. На программном уровне архитектура системы позволяет это легко реализовать, но на настройку новых серверов все равно требуется значительное количество времени, которое невозможно сократить. Потребуется внимательно следить за средним количеством ежедневных активных пользователей и предсказывать всплески активности, чтобы избежать репутационных потерь из-за медленной работы серверной части.

Система ориентирована преимущественно на использование молодыми специалистами и предлагает им дополнительные возможности, что является отличительной чертой продукта. Однако всегда остается вероятность появления конкурентного приложения, предлагающего схожие функциональные возможности. Если их бюджет будет значительно превышать бюджет продукта, часть пользователей может уйти к

конкурентам, а дальнейшие темпы прироста аудитории пользователей продукта будут существенно снижены.

Наконец, даже выделенной группе разработчиков понадобится несколько месяцев для разработки отдельного приложения для мобильных ОС. Если же пользовательские предпочтения неожиданно сместятся в сторону таких приложений, то на период их разработки интерес к продукту может значительно снизиться.

Стоит учесть, что вероятность возникновения таких событий остается низкой, и никаких серьёзных преград для реализации продукта в результате проведенного SWOT-анализа обнаружено не было.

4.2 Определение возможных альтернатив проведения научных исследований

Для определения возможных альтернативных путей проведения научных исследований использовался морфологический подход.

Морфологическими характеристиками являются: вид приложения, язык программирования, метод хранения данных системы, интегрированная среда разработки, исполнитель.

В таблице 8 представлена морфологическая матрица проекта.

Таблица 8 – Морфологическая матрица проекта

	1	2	3
А. Вид приложения	Веб-приложение	Клиентское приложение	Frontend - приложение
Б. Язык программирования	C#	Java	JavaScript
В. Метод хранения данных системы	Хранение в кэше системы	Хранение данных в базе данных	Хранение данных в облачном хранилище
Г. Интегрированная среда разработки	JetBrains Rider 2020	Microsoft Visual Studio 2019	Webstorm
Д. Исполнитель	Один программист	Два программиста	Один программист

Из данной морфологической матрицы проекта было выделено три варианта решения технической задачи для дальнейших расчетов:

И1. А1Б1В2Г1Д2;

И2. А2Б2В2Г3Д1;

И3. А3Б3В3Г2Д2.

4.2.1 Структура работ в рамках научного исследования

В нижеприведенной таблице под словом «Все» в графе исполнители работ следует понимает следующих участников работы: Бедер Константин Борисович, Соловьева Анастасия Дмитриевна, Рубанова Алевтина Игоревна, Куртуков Виктор Викторович.

Таблица 9 – Структура работ в рамках проектирования и разработки веб-приложения

Наименование работы	Исполнители работы
Выбор научного руководителя бакалаврской работы	Бедер Константин Борисович
Подбор и изучение литературы по теме бакалаврской работы	Бедер Константин Борисович
Анализ предметной области	Бедер Константин Борисович, Соловьева Анастасия Дмитриевна, Рубанова Алевтина Игоревна
Выбор программных решений для разработки клиентской части приложения	Бедер Константин Борисович, Рубанова Алевтина Игоревна
Выбор программных решений для разработки серверной части приложения	Бедер Константин Борисович
Проработка архитектуры приложения	Бедер Константин Борисович
Реализация основных микросервисов приложения	Бедер Константин Борисович
Реализация дополнительных микросервисов	Бедер Константин Борисович, Соловьева Анастасия Дмитриевна
Реализация словарей данных	Бедер Константин
Реализация средства коммуникации	Бедер Константин Борисович, Соловьева Анастасия Дмитриевна
Интегрирование бота	Соловьева Анастасия Дмитриевна
Конфигурация Docker-файлов	Бедер Константин Борисович
Выработка концептуального дизайна приложения	Соловьева Анастасия Дмитриевна

Реализация компонентов управления	Соловьева Анастасия Дмитриевна
Настройка внутренних режимов	Бедер Константин Борисович
Подключение библиотеки управления состоянием к серверной части	Бедер Константин Борисович
Настройка политики кросс-доменных запросов	Бедер Константин Борисович
Написание тестов	Куртуков Виктор Викторович
Тестирование серверной части	Куртуков Виктор Викторович, Бедер Константин Борисович
Тестирование клиентской части	Бедер Константин Борисович
Пользовательское тестирование приложения	Куртуков Виктор Викторович
Исправление ошибок приложения	Бедер Константин Борисович
Составление календарного плана-графика выполнения бакалаврской работы	Рубанова Алевтина Игоревна

4.2.2 Разработка графика проведения научного исследования

Согласно производственному календарю при расчете на 6 рабочих дней в неделю в 2019 году 365 календарных дней, 299 рабочих дней, 66 выходных или праздничных дней. Таким образом, коэффициент календарности на 2019 год вычисляется следующим образом:

$$T_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}} = 1,22$$

Для расчета временных показателей проведения научного исследования, необходимо для каждой задачи определить минимальную и максимальную ожидаемую трудоемкость, выраженную в человеко-днях. Зная эти показатели, ожидаемая трудоемкость может быть вычислена по следующей формуле:

$$t_{\text{ож}} = \frac{3t_{\text{mini}} + 2t_{\text{maxi}}}{5}$$

Для нахождения длительности этапа работы, выраженного в днях, следует найти отношение ожидаемой трудоемкости к количеству участников проектной команды, задействованной для решения этапа. Умножив полученное значение на коэффициент календарности, можно получить

ожидаемую календарную длительность каждого этапа работы, выраженную в днях. Длительность работ при этом следует округлять до целых чисел согласно математическим правилам. Таблица 10 содержит рассчитанные временные показатели научного исследования.

Таблица 10 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ									Исполнители			Длительность работ в рабочих днях T_{pi}			Длительность работ в календарных днях T_{ki}		
	t_{mini} , чел-дни			t_{maxi} , чел-дни			$t_{ожи}$, чел-дни											
	И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3
Выбор научного руководителя	4	4	4	7	7	7	5,2	5,2	5,2	2	1	2	2,6	5,2	2,6	3	6	3
Подбор и изучение литературы по теме бакалаврской работы	3	3	3	5	5	5	3,8	3,8	3,8	1	1	1	3,8	3,8	3,8	5	5	5
Анализ предметной области	4	4	4	7	7	7	5,2	5,2	5,2	3	2	3	1,7	2,6	1,7	2	3	2
Программные решения клиентской части	1	1	1	3	3	3	1,8	1,8	1,8	1	1	1	1,8	1,8	1,8	2	2	2
Программные решения серверной части	7	7	7	14	14	14	9,8	9,8	9,8	2	1	2	4,9	9,8	4,9	6	12	6
Проработка архитектуры приложения	1	1	1	2	2	2	1,4	1,4	1,4	1	1	1	1,4	1,4	1,4	2	2	2
Реализация основных микросервисов	1	2	2	2	3	3	1,4	2,4	2,4	1	1	1	1,4	2,4	1,4	2	3	2
Реализация дополнительных микросервисов	2	2	2	3	3	3	2,4	2,4	2,4	1	1	1	2,4	2,4	2,4	3	3	3
Реализация словарей данных	1	1	1	2	2	2	1,4	1,4	1,4	1	1	1	1,4	1,4	1,4	2	2	2
Реализация средства коммуникации	1	1	2	2	2	3	1,4	1,4	2,4	1	1	1	1,4	1,4	1,4	2	2	2
Эскизное проектирование интерфейса	2	2	3	3	3	4	2,4	2,4	3,4	1	1	1	2,4	2,4	2,4	3	3	3
Пользовательское тестирование приложения	2	2	2	3	3	3	2,4	2,4	2,4	1	1	1	2,4	2,4	2,4	3	3	3
Реализация api-сервиса	7	-	7	14	-	14	9,8	-	9,8	1	-	1	9,8	-	9,8	12	-	12
Реализация клиента	7	12	7	14	20	14	9,8	15,2	9,8	1	1	1	9,8	15,2	9,8	12	19	12
Авторизация	3	3	3	6	6	6	4,2	4,2	4,2	1	1	1	4,2	4,2	4,2	5	5	5
Кросс-доменные запросы	10	8	12	15	12	16	12,0	9,6	13,6	1	1	1	12,0	9,6	13,6	15	12	17
Методы визуализации	16	20	16	20	24	20	17,6	21,6	17,6	1	1	1	17,6	21,6	17,6	21	26	21
Окончательная верстка клиента	3	4	3	6	7	6	4,2	5,2	4,2	1	1	1	4,2	5,2	4,2	5	6	5
Программное-тестирование	1	2	3	3	4	5	1,8	2,8	3,8	1	1	1	1,8	2,8	3,8	2	3	5
Ручное тестирование	1	1	1	2	2	2	1,4	1,4	1,4	1	1	1	1,4	1,4	1,4	2	2	2
Исправление ошибок	1	1	1	6	5	7	3,0	2,6	3,4	2	1	2	1,5	2,6	1,7	2	3	2
Оценка соответствия ПО соответствию ТЗ	2	2	2	4	4	4	2,8	2,8	2,8	3	2	3	0,9	1,4	0,9	1	2	1
Оформление пояснительной записки	3	3	3	5	5	5	3,8	3,8	3,8	3	2	3	1,3	1,9	1,3	2	2	2

Данные из таблицы можно использовать для построения диаграммы Ганта, при этом следует учитывать, что некоторые этапы исследовательской работы могут выполняться параллельно. Видно, что количество работ, в которые вовлечен каждый участник проектной группы, соответствует его обязанностям, что стало возможным благодаря задействованию современных средств коммуникации, системе учета времени и контроля версионности исходного кода.

Построенная диаграмма Ганта представлена на рисунке 10.

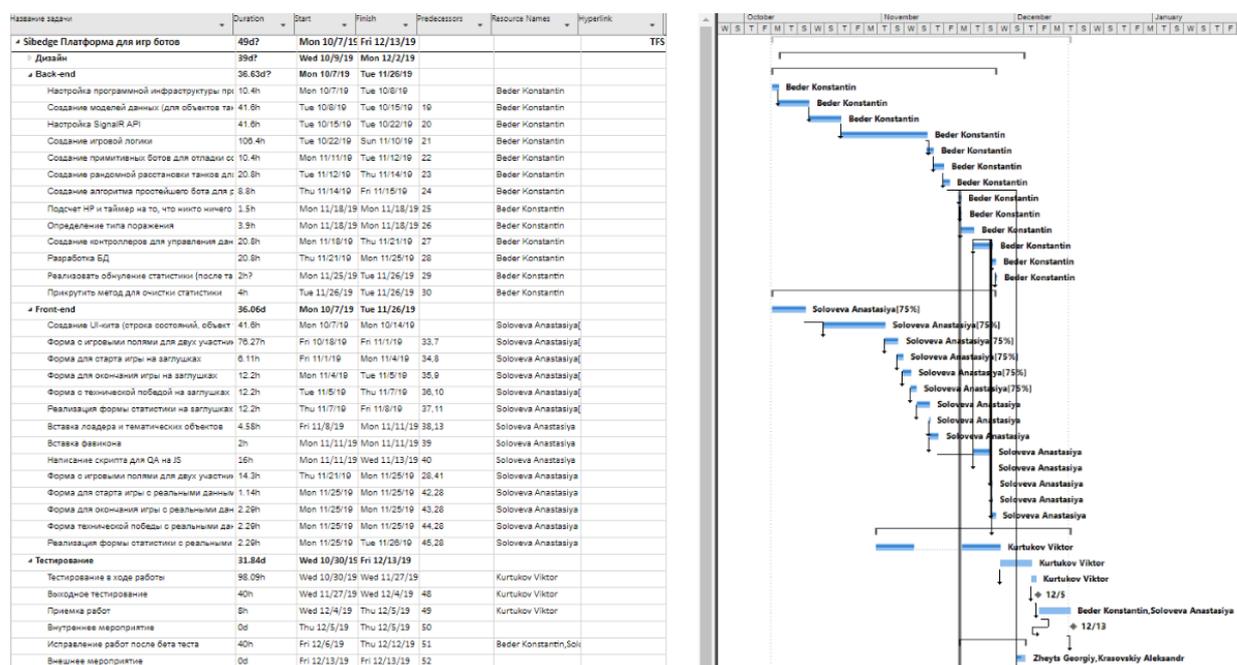


Рисунок 10 – Диаграмма Ганта, охватывающая все задачи исследовательской работы

Как видно из диаграммы, время полного выполнения исследовательской работы составило 93 календарных дня. Многие этапы выполнялись параллельно, так как затрагивали независимые компоненты веб-приложения: клиентская и серверная часть. При этом, определенные этапы требовали преждевременного завершения всех предыдущих, например, пользовательское тестирование, которое можно провести только после завершения тестирования клиентской и серверной части. Наиболее долгим этапом, занявшим 48 календарных дней (51.6 % от всего времени

исследования) оказалась разработка серверной части ввиду архитектурных сложностей и начальных требований.

4.3 Бюджет научно-технического исследования

В состав бюджета выполнения работ по научно-технической работе входит стоимость всех расходов, необходимых для их выполнения.

При формировании бюджета используется группировка затрат по следующим статьям:

- Материальные затраты НТИ;
- Затраты на специальное оборудование для научных работ;
- Основная заработная плата исполнителей темы;
- Дополнительная заработная плата исполнителей темы;
- Отчисления во внебюджетные фонды;
- Затраты на научные и производственные командировки (равны нулю для данного проекта);
- Контрагентные расходы (равны нулю для данного проекта);
- Накладные расходы.

4.3.1 Расчет материальных затрат научно-технического исследования

Расчет материальных затрат осуществляется по следующей формуле:

$$Z_m = (1 + k_T) \cdot \sum_{i=1}^m C_i \cdot N_{расхi}$$

где m – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расхi}$ – количество материальных ресурсов i -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м² и т.д.);

C_i – цена приобретения единицы i -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м² и т.д.);

k_T – коэффициент, учитывающий транспортно-заготовительные расходы.

При расчете количества электроэнергии учитывалось, что ноутбук потребляет около 60 Вт в час. Расчет материальных затрат приведен в таблице 11.

Таблица 11 – Материальные затраты

Наименование	Единица измерения	Количество			Цена за ед., руб.			Затраты на материалы, (Z_M), руб.		
		Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Оплата электроэнергии	КВт·ч	60,5	59,0	62,9	2,45			148,2	144,6	154,1
Итого								148,2	144,6	154,1

4.3.3 Расчет затрат на специальное оборудование для научных работ

В таблице 12 приведены значения затрат на специальное оборудование для трех альтернативных вариантов с учетом того, что новое оборудование не покупалось, затраты были вычислены в виде амортизационных отчислений (с учетом срока службы оборудования 5 лет, и использования его 0,5 года).

Сумма амортизации была вычислена по формуле:

$$A = \frac{C}{\text{СПИ} \cdot 12} \cdot \text{СИ}, \quad \text{где}$$

A – итоговая сумма амортизации основных средств, рублей;

C – первоначальная стоимость, рублей;

СПИ – срок полезного использования, лет;

СИ – срок использования для выполнения данной темы, месяцев.

Таблица 12 – Расчет бюджета затрат на приобретение спецоборудования для научных работ

№ п/п	Наименование оборудования			Кол-во единиц оборудования			Цена единицы оборудования, тыс. руб.			Общая стоимость оборудования, тыс. руб.		
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
1.	Ноутбук			2	1	2	22,5			45,0	22,5	45,0
2.	Компьютерная мышь			2	1	2	0,2			0,4	0,2	0,4
Итого:										45,4	22,7	45,4

4.3.4 Основная заработная плата исполнителей

Ниже представлена формула для расчета затрат на заработную платы

$$З_{п} = З_{осн} + З_{доп}, \quad \text{где}$$

$З_{осн}$ – выраженная в рублях основная заработная плата;

$З_{доп}$ – выраженная в рублях дополнительная заработная плата.

Основную заработную плату можно вычислить по формуле:

$$З_{осн} = З_{дн} * Т_{р} * (1 + К_{пр} + К_{д}) * К_{р}, \quad \text{где}$$

$З_{дн}$ – среднедневная заработная плата, руб.;

$Т_{р}$ – продолжительность работ, выполняемых работником (рабочие дни);

$К_{пр}$ – премиальный коэффициент (0,3);

$К_{д}$ – коэффициент доплат и надбавок (0,3-0,5);

$К_{р}$ – районный коэффициент, для Гомска составляет 1,3.

Среднедневная заработная плата вычисляется по формуле:

$$З_{дн} = \frac{З_{м} * М}{F_{д}}$$

$З_{м}$ – месячный оклад работника, руб.;

$М$ – количество месяцев работы без отпуска в течение года (для 6-дневной рабочей недели $М=10,4$);

Фд – действительный годовой фонд рабочего времени персонала, выражен в рабочих днях. Расчет баланса приведен в таблице 13.

Таблица 13. Баланс рабочего времени (для 6-дневной недели)

Показатели рабочего времени	Дни
Календарные дни	365
Нерабочие дни (праздники/выходные)	66
Потери рабочего времени (отпуск/невыходы по болезни)	56
Действительный годовой фонд рабочего времени	243

Таким образом, в 2019 году действительный годовой фонд рабочего времени составляет 243 дня. Найденных показателей достаточно, чтобы составить таблицу расчета основной заработной платы, при этом средняя зарплата младшего разработчика или тестировщика по Томской области в среднем равняется 21760, а руководителя проекта – 33664 рублей.

Таблица 14 – Расчет основной заработной платы

Исполнители	З _{окл} , руб.	k _р	З _м , руб.	З _{дн} , руб.	T _р , раб. дн.			З _{осн} , руб.		
					Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Руководитель	33664,0	1,3	43763,2	1440,7	4	4	4	5760	5760	5760
Программист1	21760,0	1,3	28288,0	931,29	85	127	90	79159	118237	83816
Программист2	21760,0	1,3	28288,0	931,29	85	127	85	79159	118237	79159
Тестировщик	21760,0	1,3	28288,0	931,29	85	0	85	79159	0	79159
Итого:								243237	242234	247894

4.3.5 Расчет дополнительной заработной платы

Для получения расчета дополнительной заработной платы используется следующая формула:

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}}, \quad \text{где}$$

$k_{\text{доп}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15).

В данной работе коэффициент дополнительной заработной платы принят равным 0,12.

4.3.6 Отчисления во внебюджетные фонды

Базовый размер страховых взносов в 2020 году составляет 30 процентов. Данная ставка будет использоваться при расчете отчислений во внебюджетные фонды. В таблице 15 представлены результаты расчетов.

Таблица 15 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Руководитель	5760	5760	5760	691,2	691,2	691,2
Программист1	79159	118237	83816	9499,08	14188,4	10057,9
Программист2	79159	118237	79159	9499,08	14188,4	9499,08
Тестировщик	79159	0	79159	9499,08	0	9499,08
Итого						
Исполнение 1	81727,632					
Исполнение 2	81390,564					
Исполнение 3	83292,378					

4.3.7 Накладные расходы

Помимо затрат на амортизацию и выплату заработной платы, компания также несет издержки за оплату электроэнергии, услуг связи, ксерокопии материалов и пр., эти издержки объединяются в накладные расходы, которые могут быть вычислены по следующей формуле:

$$Z_{\text{накл}} = (\text{сумма статей } 1 \div 7) \cdot k_{\text{нр}}, \quad \text{где}$$

$k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы.

В данной работе величина коэффициента накладных расходов принимается в размере 16%.

4.3.8 Формирование бюджета затрат НИИ

После того, как была рассмотрена каждая из статей расходов, можно приступить к формированию бюджета затрат проекта. Результаты вычисления итогового бюджета по каждому из исполнений проекта представлены в таблице 16.

Таблица 16 – Расчет бюджета затрат НИИ

Наименование статьи	Сумма, руб.		
	Исп. 1	Исп. 2	Исп. 3
1. Материальные затраты НИИ	148,2	144,6	154,1
2. Затраты на специальное оборудование для научных работ	45400,0	22700,0	45400,0
3. Затраты по основной заработной плате исполнителей темы	243237	242234	247894
4. Затраты по дополнительной заработной плате исполнителей темы	10190,28	18802,9	19923,6
5. Отчисления во внебюджетные фонды	81727,632	81390,564	83292,378
6. Затраты на научные и производственные командировки	0	0	0
7. Контрагентские расходы	0	0	0
8. Накладные расходы	8701,7	8349,0	9066,607
9. Бюджет затрат НИИ	389404,8	373621,0	405730,6

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

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad \text{где}$$

$I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта.

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования определяется следующей формулой:

$$I_{pi} = \sum a_i \cdot b_i, \quad \text{где}$$

где I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i – бальная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания.

Критерии, их весовые коэффициенты и оценки вариантов исполнения разработки, необходимые для расчета, представлены в таблице 17.

Таблица 17 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
1. Простота использования системы	0,2	5	4	4
2. Скорость разработки	0,2	5	2	4
3. Функциональные возможности	0,5	4	4	3
4. Скорость обработки данных	0,1	4	4	3
5. Потребность в ресурсах памяти	0,1	4	3	5
ИТОГО	1			

$$I_{p-исп1} = 5 * 0,2 + 5 * 0,2 + 4 * 0,5 + 4 * 0,1 + 4 * 0,1 = 4,8$$

$$I_{p-исп2} = 4 * 0,2 + 2 * 0,2 + 4 * 0,5 + 4 * 0,1 + 3 * 0,1 = 3,9$$

$$I_{p-исп3} = 4 * 0,2 + 4 * 0,2 + 3 * 0,5 + 3 * 0,1 + 5 * 0,1 = 3,9$$

Интегральный показатель эффективности вариантов исполнения разработки $I_{исп.i}$ определяется по формуле:

$$I_{исп.i} = \frac{I_{p-исп.i}}{I_{финр.i}}$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволяет определить сравнительную эффективность проекта, представленную в таблице 18, и выбрать лучший вариант из предложенных. Сравнительная эффективность проекта $\mathcal{E}_{ср}$ рассчитывается по следующей формуле:

$$\mathcal{E}_{ср} = \frac{I_{исп.i}}{I_{исп.max}}$$

Таблица 18 – Сравнительная эффективность разработки

№ п/п	Показатели	Исп. 1	Исп. 2	Исп. 3
1	Интегральный финансовый показатель разработки	0,97	0,87	1,00
2	Интегральный показатель ресурсоэффективности разработки	4,80	3,90	3,90
3	Интегральный показатель эффективности	4,95	4,48	3,90
4	Сравнительная эффективность вариантов исполнения	1,00	0,91	0,79

Согласно значениям интегрального показателя эффективности, наиболее эффективным является вариант исполнения 1.

Первый вариант разработки имеет наивысший интегральный показатель ресурсоэффективности и второй по величине интегральный финансовый показатель. Данный вариант не является самым недорогим, но обеспечивает максимальное удобство разработки и использования, производительности и скорость разработки.

5 Социальная ответственность

Исследовательская работа заключалась в проектировании и разработке веб-приложения, помогающего рекрутерам отбирать новых потенциальных работников среди всех кандидатов.

Используя приложение, работодатели будут способны быстро и точно понять профессиональные уровни отбираемых кандидатов. Проведение такого рода мероприятий позволяет не только экономить время рекрутеров и соискателей, а также не требует обязательного личного присутствия, весь процесс устроен так, что его проведение в режиме онлайн, гарантирует практически тот же результат.

Согласно требованиям, предъявленным к разрабатываемому веб-приложению, клиентами могут являться пользователи, использующие как стационарные персональные компьютеры, так и ноутбуки.

По этой причине интерфейс веб-приложения необходимо было адаптировать под различные пользовательские устройства, проверка корректности разметки содержимого осуществлялась лично участниками исследовательской работы. Сама разработка при этом велась с использованием ноутбуков и персонального компьютера, в связи с чем участники проектной группы подвергались таким вредным факторам, как:

Некачественные TN-матрицы ноутбуков под острым углом сильно искажают цвета, вплоть до полного инвертирования при угле, близком к 180 градусов. Неправильная цветопередача может привести к легкой дезориентации и головным болям;

Частота вертикальной развертки мониторов ограничена 60 Гц, мерцание, возникающее при этом из-за особенностей работы ШИМ-контроллера, быстро утомляет глаза;

В непосредственной близости от электронно-вычислительных устройств существует область повышенного электромагнитного излучения, длительное нахождение в которой вредно для внутренних органов;

Специфичность трудовой деятельности, связанной с разработкой веб приложений, состоит в необходимости выполнения работ преимущественно в сидячем положении, что может привести к нарушению кровообращения и образованию зловредных новообразований;

Сбои в электропитании могут нарушить равномерность подсветки матриц жидкокристаллических мониторов, в результате утомляемость глаз может дополнительно усилиться;

В вечернее и ночное время разработчики вынуждены использовать искусственное освещение.

5.1 Правовые и организационные вопросы обеспечения безопасности

5.1.1 Организационные мероприятия обеспечения безопасности

Разработка веб-приложения велась в помещении офиса компании Сибэдж, находящийся по адресу г.Томск ул.Пушкина 75, общей площадью 26 квадратных метров, в качестве искусственного источника освещения использовались 3 лампы накаливания, общей мощностью 85 Вт. Помещение оборудовано одним крупным столом, позволяющим работать одновременно до двух человек, и одним компьютерным столом стандартного размера с выдвижной клавиатурой. Участники научно-исследовательской работы взаимодействовали с электронно-вычислительными устройствами, находясь в операторских креслах, выполненных в виде компьютерного кресла с регулируемыми подлокотниками и углом наклона спинки. Перемещение кресла внутри помещения обеспечивают 5 пластиковых колес диаметром 50 мм. Каждый участник исследования постоянно имел доступ к свободному операторскому креслу.

С целью минимизации воздействия вредных факторов на группу разработчиков при выполнении научно-исследовательской работы, рабочее место должно быть организовано с учетом требований ГОСТ 12.2.032-78

«ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования» и СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

Ниже приведены наиболее важные для соблюдения фрагменты стандарта (используется оригинальная нумерация пунктов соответствие с ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования»):

Подвижность кресла относительно пола или другой поверхности, на которой оно установлено, может не ограничиваться. В случае необходимости обеспечения строго определенного положения человека-оператора по отношению к средствам отображения информации и органам управления, а также в случае, если трудовая деятельность человека-оператора сопряжена с силовыми и резкими движениями, кресло должно быть фиксировано. При этом, в зависимости от характера трудовой деятельности оператора, должна быть обеспечена возможность изменения положения кресла или сиденья в горизонтальной плоскости с фиксацией его в нужном положении. При необходимости подвижность кресла должна задаваться также вращением кресла на 180-360° вокруг вертикальной оси опорной конструкции кресла с фиксацией в нужном положении.

Подставка для ног должна быть регулируемой по высоте. Ширина должна быть не менее 300 мм, длина - не менее 400 мм. Поверхность подставки должна быть рифленой. По переднему краю следует предусматривать бортик высотой 10 мм.

При работе двумя руками органы управления размещают с таким расчетом, чтобы не было перекрещивания рук.

Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

Расстояние между внутренними гранями подлокотников определяется межлоктевым диаметром, измеренным в положении сидя, для 95 перцентиля данного антропометрического признака, с поправкой на специальную одежду и снаряжение, а при наличии регулировки параметра 50-95 перцентилями данного антропометрического признака.

В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.

Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). При устройстве отраженного освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп. В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенные.

Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5-2,0 м.

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Проектной группой научно-исследовательской работы были соблюдены в допустимой мере все требования, предусматриваемые государственным стандартом 12.2.032-78. Во время выполнения выпускной квалификационной работы не происходило случаев, несущих в себе угрозы

для здоровья и жизни сотрудников и участников проектной группы, а также представляли опасности для окружающей среды.

5.2 Производственная безопасность

В данном пункте производится анализ вредных и опасных факторов, которые могут возникнуть на одном из этапов выполнения работы. Химические факторы не рассмотрены в связи с использованием высококачественных материалов и отсутствием необходимости взаимодействия с химикатами в процессе выполнения научно-исследовательской работы. Помещение, где выполнялись работы по созданию серверного приложения для асинхронного симметричного коммуницирования сетевых клиентов, располагалось вдали от крупных городских улиц на 6 этаже, в связи с чем шумовые факторы были неспособны оказать существенное влияние на ход проведения работ.

Рассмотрены следующие факторы: отклонение показателей микроклимата и естественного, недостаточная освещенность и повышенный уровень электромагнитного излучения. Таблица 19 содержит информацию о воздействии факторов в зависимости от этапов научного исследования.

Таблица 19 – Возможные опасные и вредные факторы

Факторы (по ГОСТ 12.0.003-2015) [13]	Этапы работ			Нормативные документы
	Проектирование	Разработка	Формирование отчетности	
Отклонение показателей микроклимата	+	+	+	СанПиН 2.2.4.548-96 [9]
Отсутствие или недостаток естественного света	+	+	+	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95* [10]
Недостаточная освещенность рабочей зоны	+	+	+	СанПиН 2.2.1/2.1.1.1278–03 [11]
Повышенный уровень электромагнитных излучений	+	+	+	СанПиН 2.2.4.1191-03 [12]

Как видно из таблицы 19, опасные и вредные факторы воздействовали на участников разработки приложения для асинхронного симметричного коммуницирования сетевых клиентов на всем протяжении работ, так как все этапы научно-исследовательской работы были произведены в одном и том же помещении.

5.2.1 Анализ опасных и вредных производственных факторов

Таблица 20 – Влияние опасных и вредных факторов

Фактор	Источник	Воздействие	Допустимые нормы
Отклонение показателей микроклимата	Отсутствие кондиционеров	Вялость, усталость, сниженная концентрация	Таблица 3
Отсутствие или недостаток естественного света	Периодическая необходимость работы за ЭВМ в ночное время	Ухудшение зрения, усталость глаз	КЕО не ниже 1,2 %-1,5 %
Недостаточная освещенность рабочей зоны	Недостаточная мощность осветительных приборов	Ухудшение зрения, усталость глаз	Освещенность на рабочей поверхности от системы общего искусственного освещения 200-300 лк.
Повышенный уровень электромагнитных излучений	Компоненты персональных компьютеров и ноутбуков	Возможно возникновение рака	Напряженность электростатического поля не более 20 кВ/м

Норма микроклимата является плавающим параметром и зависит от температуры помещения, поверхностей, влажности и скорости воздуха. Допустимые величины показателей микроклимата продемонстрированы в таблице 21.

Таблица 21 – Допустимые величины показателей микроклимата

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	20,0-21,9	19,0-26,0	15-75	0,1
Теплый	21,0-22,9	20,0-29,0		

5.2.2 Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов

Для восстановления и поддержания допустимого микроклимата необходимо придерживаться следующих правил:

- Оборудование помещения системами обогрева, вентилирования и увлажнения;
- Оборудование помещения современными пластиковыми окнами, поддерживающими возможность микровентилирования;
- Защита фасада здания от солнца: шторы, жалюзи, навесы и т.д;
- Рационально размещать рабочие места;
- Ежедневная влажная уборка рабочего помещения.

Для решения проблемы отсутствия или недостатка естественного света и плохой освещенности рабочего места подходят следующие пункты:

- Сокращение времени работы;
- Своевременная чистка стекол в светопроемах;
- Снос деревьев, препятствующих проникновению света в помещение;
- Ремонт помещения в светлых тонах;
- Установка более мощных ламп или в большем количестве;
- Установка ламп в правильном положении.

Повышенный уровень электромагнитных излучений можно избежать, если следовать следующим пунктам:

- Прекратить использование мониторов с электронно-лучевой трубкой;
- Использовать высокоэффективные блоки питания и прочие преобразователи напряжения;
- Располагать монитор в углу помещения для того, чтобы стены поглощали излучение;
- Выключать компьютер при его неиспользовании;
- Сокращать время, проводимое за компьютером.

5.3 Экологическая безопасность

Для выполнения научно-исследовательской работы по разработке серверного приложения для асинхронного симметричного коммуницирования сетевых клиентов использовались компьютеры, ноутбуки и смартфоны средней мощности. Современные электронно-вычислительные устройства не выбрасывают в окружающую среду каких-либо вредных веществ, однако используют для работы электроэнергию и создают электромагнитные поля.

Также при производстве смартфонов и персональных компьютеров используются тяжелые, щелочноземельные металлы, ртуть, пластик и стекло, что без должной утилизации по окончании службы попадает в природу и остается в не переработанном виде.

Мероприятия, позволяющие сохранять экологическую безопасность находясь на своем рабочем месте:

- Правильная утилизация персональных компьютеров и смартфонов, а также их комплектующих;
- Использование энергосберегающих ламп;
- Использование аккумуляторов вместо солевых батареек.

5.4 Безопасность в чрезвычайных ситуациях

Одной из наиболее возможной чрезвычайной ситуацией, которая может возникнуть при работе в помещениях такого типа – пожар.

5.5 Пожар

Научно-исследовательская работа по разработке серверного приложения для асинхронного симметричного коммуницирования сетевых клиентов проходила в помещении офиса компании Сибэдж, находящийся по адресу г.Томск ул.Пушкина 75, подходящем под определение офис. К пожару

могут привести неисправности в технических средствах, оргтехнике, а также действия самих сотрудников. Главное во время пожара – не поддаваться панике и действовать согласно правилам поведения при пожаре. Для сотрудника существует порядок действий и правила поведения в подобной чрезвычайной ситуации:

- Заметив пожар или загорание, необходимо немедленно организовать оповещение об этом всех находящихся в здании людей, независимо от размеров и места пожара или загорания, равно как и при обнаружении хотя бы малейших признаков горения (дыма, запаха гари) и немедленно вызвать пожарную охрану по телефону «01». Очевидно, что быстрота прибытия пожарной помощи, позволит успешнее ликвидировать пожар и быстрее помочь людям, находящимся в опасности;
- Сообщения о пожаре, как правило, передаются по телефону. Поэтому каждый человек должен хорошо знать места расположения телефонных аппаратов, особенно тех, которые доступны в любое время суток;
- Каждый работник образовательного учреждения, обнаруживший пожар или его признаки (задымление, запах горения или тления различных материалов, повышение температуры и т.п.) обязан:
- Немедленно сообщить об этом по телефону в пожарную часть (при этом необходимо четко назвать адрес учреждения, место возникновения пожара, а также сообщить свою должность, фамилию и номер своего телефона);
- Задействовать систему оповещения людей о пожаре, приступить самому и привлечь других лиц к эвакуации детей из здания в безопасное место согласно плану эвакуации;
- Известить о пожаре руководителя образовательного учреждения или заменяющего его работника;

- Организовать встречу пожарных подразделений, принять меры по тушению пожара имеющимися в учреждении средствами пожаротушения.

В результате изучения и анализа стандартов и правил, касающихся работы в помещениях с электронно-вычислительными устройствами, можно сделать вывод, что осуществление проекта для асинхронного симметричного коммуницирования сетевых клиентов соответствовало всем заявленным нормам безопасности жизнедеятельности. Участники проектной группы не подвергались серьезному воздействию опасных факторов. Рабочие места и помещение в целом во время проведения исследовательской работы соответствовало региональным стандартам, а также санитарно-эпидемиологическим правилам и нормам. Федеральный закон «О персональных данных» внес изменения в архитектуру серверной части и визуальному изменению некоторых компонентов приложения, члены проектной группы готовы к поведению в условиях чрезвычайных ситуаций.

ЗАКЛЮЧЕНИЕ

Результатом работы является разработанное, протестированное и внедренное приложение для асинхронной симметричной коммуникации сетевых клиентов, а также игра «Cybertank» на его основе. Благодаря строгому соответствию принципам «SOLID» и использованию проверенных паттернов проектирования, удалось достичь высокой поддерживаемости кода, что означает высокую степень готовности веб-приложения к дальнейшим изменениям, поддержке и масштабированию.

Использование средства конфигурирования машин конечных автоматов Stateless позволило создать игровую платформу для реализации любой шаговой игры, основываясь на теории конечных автоматов.

Разработанная платформа и игра «Cybertank» позволили компании Сибэдж провести хакатон и упростить отбор соискателей. Компания планирует дальше развивать данную платформу для создания новых игр и проведения новых тематических хакатонов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация React [Электронный ресурс] – URL: <https://ru.reactjs.org/docs/getting-started.html> (дата обращения: 05.05.20)
2. Microsoft SQL Server 2017 [Электронный ресурс]. – URL: <http://www.microsoft.com/sql-server/sql-server-2017> (дата обращения: 02.06.2020).
3. Документация по .NET Core [Электронный ресурс]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/core/> (дата обращения: 27.05.2020)
4. Docker Documentation [Электронный ресурс]. – URL: <https://docs.docker.com/> (дата обращения: 22.05.2020)
5. Portainer Documentation [Электронный ресурс] – URL: <https://www.portainer.io/documentation/> (дата обращения: 22.05.2020)
6. Как устроен ConcurrentBag в .Net [Электронный ресурс] – URL: <https://habr.com/ru/post/241706/> (дата обращения: 22.05.2020)
7. Real-time ASP.NET with SignalR [Электронный ресурс] – URL: <https://dotnet.microsoft.com/apps/aspnet/signalr> (дата обращения: 22.05.2020)
8. Stateless [Электронный ресурс] – URL: <https://github.com/dotnet-state-machine/stateless> (дата обращения: 22.05.2020)
9. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений [Электронный ресурс] – URL: <http://docs.cntd.ru/document/901704046> (дата обращения: 01.05.2020)
10. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95 [Электронный ресурс] – URL: <http://docs.cntd.ru/document/456054197>
11. СанПиН 2.2.1/2.1.1.1278-03 Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий [Электронный ресурс] – URL: <http://docs.cntd.ru/document/901859404> (дата обращения: 01.05.2020)

12. СанПиН 2.2.4.1191-03 Электромагнитные поля в производственных условиях [Электронный ресурс] – URL: <http://docs.cntd.ru/document/901853847> (дата обращения: 01.05.2020)
13. ГОСТ 12.0.003-2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация [Электронный ресурс] – URL: <http://docs.cntd.ru/document/1200136071> (дата обращения: 01.05.2020)

ПРИЛОЖЕНИЕ А

Акт о внедрении



www.sibedge.com
contacts@sibedge.com

ООО «Сибэдж», LLC «SibEDGE»
634006, Томск, ул. Пушкина, 75
тел: +7 (38-22) 701-841

ОГРН 1127017015179
ИНН/КПП 7017307138 / 701701001
РЕГ.НОМЕР 7000037783

АКТ

о внедрении результатов бакалаврской работы
«Серверное приложение для асинхронного симметричного коммуницирования сетевых клиентов»
Бедера Константина Борисовича

Комиссия в составе:

- Руководитель отдела разработки Алеутдинов Олег Владимирович,

Другие члены комиссии:

- Рубанова Алевтина Игоревна, руководитель проектов,
- Круковский Виктор Александрович, стек-менеджер Frontend-разработчиков,

составила настоящий акт о том, что результаты бакалаврской работы студента Национального исследовательского Томского политехнического университета Бедера Константина Борисовича внедрены в отдел управления персоналом компании «Sibedge» в виде проекта для автоматизации процессов по отбору персонала и рекрутинга.

Работа выполнялась в соответствии с согласованным техническим заданием и предоставленными «Sibedge» документами.

Использование данной программы помогает отделу рекрутинга с проведением мероприятий по привлечению новых сотрудников, таких как хакатоны.

Руководитель отдела разработки

Члены комиссии

Печать организации


Алеутдинов О.В.

Рубанова А.И.

Круковский В.А.

