

Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.04 «Программная инженерия»
 Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Проектирование и реализация чат-бота стримингового сервиса для повышения вовлеченности аудитории

УДК 004.455.2:004.744

Студент

Группа	ФИО	Подпись	Дата
8К61	Щукин Василий Алексеевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Савельев Алексей Олегович	к.т.н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

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

По разделу «Социальная ответственность»

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

ДОПУСТИТЬ К ЗАЩИТЕ:

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

**Планируемые результаты обучения по направлению 09.03.04
«Программная инженерия»**

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

Инженерная школа информационных технологий и робототехники

Направление подготовки: 09.03.04 «Программная инженерия»

Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы

Студенту:

Группа	ФИО
8К61	Щукину Василию Алексеевичу

Тема работы:

Проектирование и реализация чат-бота стримингового сервиса для повышения вовлеченности аудитории	
Утверждена приказом директора (дата, номер)	№59-51/с от 28.02.2020

Срок сдачи студентом выполненной работы:

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Исходные данные к работе <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>	Объектом проектирования в исследовательской работе является чат бот стримингового сервиса. Режим работы: клиент-сервер. Особые требования к продукту: масштабируемость; поддержка всех современных браузеров; высокий уровень доступа.
Перечень подлежащих исследованию, проектированию и разработке вопросов <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>	<ol style="list-style-type: none">1. Исследование предметной области и проектирование программной системы2. Разработка программной системы3. Анализ результатов разработки программной системы4. Финансовый менеджмент5. Социальная ответственность
Перечень графического материала <i>(с точным указанием обязательных чертежей)</i>	<ol style="list-style-type: none">1. Модель работы программной системы (диаграммы в нотациях IDEF0, IDEF3)2. Диаграмма EPC3. Диаграмма BPMN4. Рисунки, демонстрирующие результаты5. Диаграмма Ганта

Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i>	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Спицына Любовь Юрьевна
Социальная ответственность	Белоенко Елена Владимировна
Названия разделов, которые должны быть написаны на русском и иностранном языках:	

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

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Савельев Алексей Олегович	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К61	Щукин Василий Алексеевич		

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники

Направление подготовки: 09.03.04 «Программная инженерия»

Уровень образования бакалавриат

Отделение информационных технологий

Период выполнения осенний/весенний семестр 2019/2020 учебного года

Форма представления работы:

бакалаврская работа

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Глава 1. Описание предметной области	15
	Глава 2. Проектирование чат бота для платформы Twitch.tv	20
	Глава 3. Обоснование выбора технологий	15
	Глава 4. Разработка приложения	20
	Глава 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
	Глава 6. Социальная ответственность	15

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Савельев Алексей Олегович	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата

Доцент ИШИТР ТПУ	ОИТ	Чердынцев Евгений Сергеевич	к.т.н.		
---------------------	-----	-----------------------------------	--------	--	--

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

Студенту:

Группа	ФИО
8К61	Щукину Василию Алексеевичу

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

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

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Бюджет проекта – не более 500 тыс. руб., В т.ч. на затраты на оплату труда не более 250 тыс. руб.
2. <i>Нормы и нормативы расходования ресурсов</i>	Значение показателя интегральной ресурсоэффективности не менее 4 баллов из 5; Значение интегрального финансового показателя не более 1;
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Коэффициент отчислений на уплату во внебюджетные фонды - 30%; Коэффициент накладных расходов - 16%;

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
2. <i>Планирование и формирование бюджета научных исследований</i>	Формирование плана и графика разработки: - определение структуры работ; - определение трудоемкости работ; - разработка графика Ганта. Формирование бюджета затрат на научное исследование: - материальные затраты; - заработная плата (основная и дополнительная); - отчисления на социальные цели; - накладные расходы.
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	- Определение потенциального эффекта исследования

Перечень графического материала (с точным указанием обязательных чертежей):

1. <i>Оценка конкурентоспособности технических решений</i>
2. <i>Матрица SWOT</i>
3. <i>График Ганта</i>
4. <i>Расчет бюджета затрат</i>

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

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Спицына Л. Ю.	Кандидат экономических наук		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К61	Щукин Василий Алексеевич		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8К61	Щукину Василию Алексеевичу

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

Тема ВКР:

Проектирование и реализация чат-бота стримингового сервиса для повышения вовлеченности аудитории	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объект исследования - алгоритм обработки текстовых команд в чате. Область применения – онлайн платформы с возможностью обмена информацией через текстовый чат.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	Трудовой кодекс РФ СанПиН 2.2.2/2.4.1340-03 ТОИ Р-45-084-01 ГОСТ 12.2.032-78
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	<ul style="list-style-type: none"> – отклонение показателей микроклимата в помещении – недостаточная освещенность рабочей зоны – отсутствие или недостаток естественного света – повышенный уровень электромагнитных излучений
3. Экологическая безопасность:	– анализ негативного воздействия на окружающую природную среду: утилизация компьютеров, смартфонов, оргтехники.
4. Безопасность в чрезвычайных ситуациях:	<ul style="list-style-type: none"> – пожар – землетрясение.

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

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП ТПУ	Белоенко Елена Владимировна	Кандидат технических наук		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К61	Щукин Василий Алексеевич		

Реферат

Выпускная квалификационная работа содержит 91 страницу, 20 рисунков, 20 таблиц, 24 источника и 3 приложения.

Ключевые слова: стриминговая платформа, чат бот, трансляция, вовлеченность, эфир, React, NodeJS.

Объектом исследования является программная система чат бот стримингового сервиса для повышения вовлеченности аудитории.

Цель работы – Повышение лояльности и вовлеченности аудитории посредством использования программной системы чат бота с целью повышения конкурентоспособности трансляции.

В процессе исследования проводилось исследование текстовых ассистентов, их классификации и возможность приложения механизмов привлечения и удержания аудитории к площадкам потокового вещания посредством программной системы на языке программирования TypeScript

В результате исследования была спроектирована и разработана программная система, позволяющая автоматизировать часть рутинных задач ведущего трансляции, привлечь новую аудиторию и удержать существующую.

Экономическая эффективность работы заключается в актуальности поставленной задачи, применению современных программных средств, отсутствии прямых конкурентных решений.

В дальнейшем планируется расширение функциональности системы, в том числе списка обрабатываемых команд и объема контента, поставляемого через текстовый чат трансляции, снятие статистических метрик в условиях реального пользования.

Оглавление

Перечень терминов и условных обозначений.....	15
Введение	18
Глава 1. Описание предметной области	20
Глава 2. Проектирование чат-бота для платформы Twitch.tv	22
2.1. Требования к услугам обеспечения качества	27
2.1.1. Общие требования к составу, содержанию и порядку оказания услуг.....	27
2.1.2. Функциональные требования.....	27
2.1.3. Нефункциональные требования.....	29
Глава 3. Обоснование выбора технологий	41
3.1 Обоснование выбора программных средств разработки.....	41
3.2 Фреймворк серверной части	42
3.3 СУБД	43
3.4 Клиентское приложение.....	44
3.5 Облачный сервис.....	46
Глава 4. Разработка приложения	48
4.1. Сервер.....	48
4.2. Клиентская часть.....	50
4.3. Деплой в Heroku	53
Глава 5. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения	55
5.1. Оценка перспективности проведения исследований	55
5.1.1 Потенциальные потребители результатов исследования	55
5.1.2. Анализ конкурентных технических решений	56
5.1.3. Технология QuaD	58
5.1.4. SWOT-анализ.....	59
5.2. Планирование научно-исследовательских работ.	61

5.2.1. Структура работ в рамках научного исследования.	61
5.2.2. Разработка графика проведения научного исследования.	62
5.3. Бюджет научно-технического исследования	66
5.3.1. Расчет материальных затрат научно-технического исследования	66
5.3.2. Основная заработная плата исполнителей исследовательской работы.....	66
5.3.3. Расчет дополнительной заработной платы.....	68
5.3.4. Отчисления во внебюджетные фонды	68
5.3.5. Накладные расходы.....	69
5.3.6. Формирование бюджета затрат научно-исследовательского проекта.....	69
5.4. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	70
5.5. Выводы по главе	72
Глава 6. Социальная ответственность	73
6.1. Введение	73
6.2. Правовые и организационные вопросы обеспечения безопасности 74	
6.2.1. Организационные мероприятия обеспечения безопасности	74
6.3. Производственная безопасность	76
6.3.1. Анализ опасных и вредных производственных факторов	77
6.3.2. Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов.....	78
6.3.2.1. Мероприятия по снижению воздействия недопустимого микроклимата.....	79
6.3.2.2. Мероприятия по снижению воздействия недостатка освещенности рабочего места и естественного света	79
6.3.2.3. Мероприятия по снижению воздействия недопустимого уровня электромагнитного излучения	80

6.4 Экологическая безопасность	80
6.5 Безопасность в чрезвычайных ситуациях	81
6.5.1 Пожар.....	81
6.5.2 Землетрясение.....	82
6.6 Выводы по главе	83
Заключение	84
Список использованных источников.....	85
Приложение А. Диаграмма событий бизнес-процесса	88
Приложение Б. Клиентский интерфейс программной системы	90
Приложение В. Рабочая среда облачного сервиса Heroku	91

Перечень терминов и условных обозначений

1. **БД (База Данных)** – представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ)
2. **Веб-сайт** – одна или несколько логически связанных между собой веб-страниц.
3. **Веб-приложение** — клиент-серверное приложение, в котором клиент взаимодействует с сервером при помощи браузера, а за сервер отвечает веб-сервер.
4. **Веб-сервер** — сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными.
5. **HTTP (HyperText Transfer Protocol)** – протокол прикладного уровня передачи данных (изначально — в виде гипертекстовых документов в формате «HTML», в настоящий момент используется для передачи произвольных данных).
6. **HTML (HyperText Markup Language)** – стандартизированный язык разметки документов во Всемирной паутине.
7. **API (Application Programming Interface)** – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой. Позволяет отвечать на Get- и Post- запросы (получение и отправка данных соответственно).
8. **REST API** - архитектурный стиль взаимодействия компонентов распределённого приложения в сети, представляет собой

согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа-системы.

9. **Get** - метод, который используется для запроса содержимого указанного ресурса.
10. **Post** - метод, который применяется для передачи пользовательских данных заданному ресурсу.
11. **Git** – распределенная система управления версиями.
12. **SPA (Single Page Application)** – это веб-приложение или веб-сайт, использующий единственный HTML-документ как оболочку для всех веб-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript.
13. **Node.js** - программная платформа, основанная на движке V8, которая транслирует JavaScript в машинный код.
14. **NPM (Node Package Manager)** – менеджер пакетов, входящий в состав Node.js.
15. **React** – JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов.
16. **WebStorm** – интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы IntelliJ IDEA.
17. **Visual Studio Code (VS Code)** - редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS.
18. **Atom** - бесплатный текстовый редактор с открытым исходным кодом и поддержкой плагинов, написанных на Node.js, и встраиваемых под управлением Git.
19. **JavaScript (JS)** – мультипарадигменный язык программирования, поддерживающий объектно-ориентированный, императивный и функциональный стили.

20. **CSS (Cascading Style Sheets – каскадные таблицы стилей)** – формальный язык для описания разметки веб-документов, хранения изображений, а также компилированного программного кода.
21. **SASS** - метаязык на основе CSS, предназначенный для увеличения уровня абстракции CSS кода и упрощения файлов каскадных таблиц стилей.
22. **SCSS** – разновидность синтаксиса SASS, использующая фигурные скобки.
23. **СУБД (система управления базами данных)** – комплекс программ, позволяющих создать базу данных (БД) и манипулировать данными (вставлять, обновлять, удалять и выбирать).
24. **MS SQL Server** – система управления реляционными базами данных, разработанная корпорацией Microsoft.
25. **Фреймворк** – заготовки, шаблоны для программной платформы, определяющие структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных модулей программного проекта.
26. **ORM** – программное решение, позволяющее связать схему базы данных с представлениями объектов в объектно-ориентированных языках программирования;
27. **Транзакция** – атомарная с точки зрения СУБД последовательность операций, переводит базу из одного устойчивого состояния в другое;
28. **JSON** – формат обмена данными, запись которых идентична представлению объектов в JavaScript;
29. **DOM** – представление содержимого веб-страницы в виде иерархической структуры;
30. **Webpack** – современная система сборки клиентского кода, поддерживающая возможности гибкой настройки параметров;
31. **NPM** – распространенный менеджер пакетов, входящий в состав Node.js.

Введение

Стоимость глобального рынка потоковых мультимедиа-платформ к 2019 году превысила 42 миллиарда долларов с предполагаемым среднегодовым темпом роста в 20,4% в период с 2020 по 2027 гг.[1] Развитие данного сегмента подкрепляется инновациями в области криптовалют и искусственного интеллекта, что приводит к повышению качества изображения – главного фактора развития указанной индустрии. Бурный темп развития потокового вещания также обусловлен активной адаптацией рынка для мобильных устройств, так как подавляющая часть пользователей платформ используют портативные средства коммуникации.

Сервисы телевизионного потокового вещания, такие как Netflix, являются самым быстро растущим сегментом индустрии ТВ.[2] Также потоковый видео контент может быть использован для улучшения учебного процесса в сфере образования. Примерами такой модернизации изучения материала могут служить онлайн курсы, лекции и вебинары. Такой способ изложения информации положительно влияет на усвоение материала учащимися. Таким образом, многие школы, университеты и другие образовательные учреждения стали частью рынка потокового мультимедиа. Быстрый доступ к материалу, повышенный спрос, удобные модели монетизации, растущий процент пользователей мобильных устройств и повсеместный доступ к сети Интернет положительно влияет на целевой рынок. Основываясь на типе контента, рынок можно разделить на два блока:

- Потоковое онлайн вещание;
- Нелинейный видео поток;

В 2019 году сегмент потокового вещания являлся лидером по показателю суммарной прибыли. Это объясняется растущим спросом на цифровые мультимедийные устройства в сочетании с быстрым Интернет-

соединением, позволяющим получать удаленный доступ к медиа контенту. Существует несколько других факторов, которые повышают прибыль данного сегмента: контент без рекламы, услуги повышенного качества видео, большая целевая аудитория. События, происходящие в реальном времени, такие как спортивные события и запуск ракет, подтвердили спрос общества на подобные сервисы.

Объектом исследования является узкий сегмент рынка, ориентированный на потоковые онлайн трансляции по тематике видеоигр. Пользователи подобного рода сервисов потратили суммарно 17,9 миллионов часов на просмотр видео контента за первый квартал 2018 года [3]. В данном сегменте лидирующим игроком является платформа онлайн мультимедиа Twitch.tv. За трансляцией может наблюдать неограниченное число зрителей (во время масштабных спортивных событий данные цифры достигают сотен тысяч). Действующими лицами данного бизнес процесса являются ведущий и зрители.

Глава 1. Описание предметной области

Онлайн эфиры в большей части можно отнести к сфере развлечений, поэтому при выборе из десятков и сотен альтернатив решающим фактором для зрителя является контент трансляции: он может быть разнообразным – от спортивного матча в видеоигре до эфира с уроками по кулинарии. Влияние на события в эфире исключено – это процесс, полностью управляемый автором. Из этого следует, что главным вектором развития информационных систем в данной сфере должно быть расширение инструментария автора канала. Так как основным источником контента на трансляции является экран автора, а предоставляемый платформой набор инструментов ограничен, в условиях высокой конкуренции требуется искать новые источники контента. Таким источником может являться чат трансляции, так как им может воспользоваться каждый зритель, и это инструмент, предоставляемый любым стриминговым сервисом.

Автоматизация общения с людьми в чате и внедрение ботов – один из векторов развития данной платформы и сообщества, т.к. благодаря данным программам человек, ведущий эфир, не вынужден отвлекаться и многократно отвечать на похожие запросы пользователей. Наиболее популярные функции чат-ботов: модерация сообщений, предоставление дополнительной информации (время эфира, краткая информация о ведущем). На рисунке 1 приведен снимок интерфейса платформы.

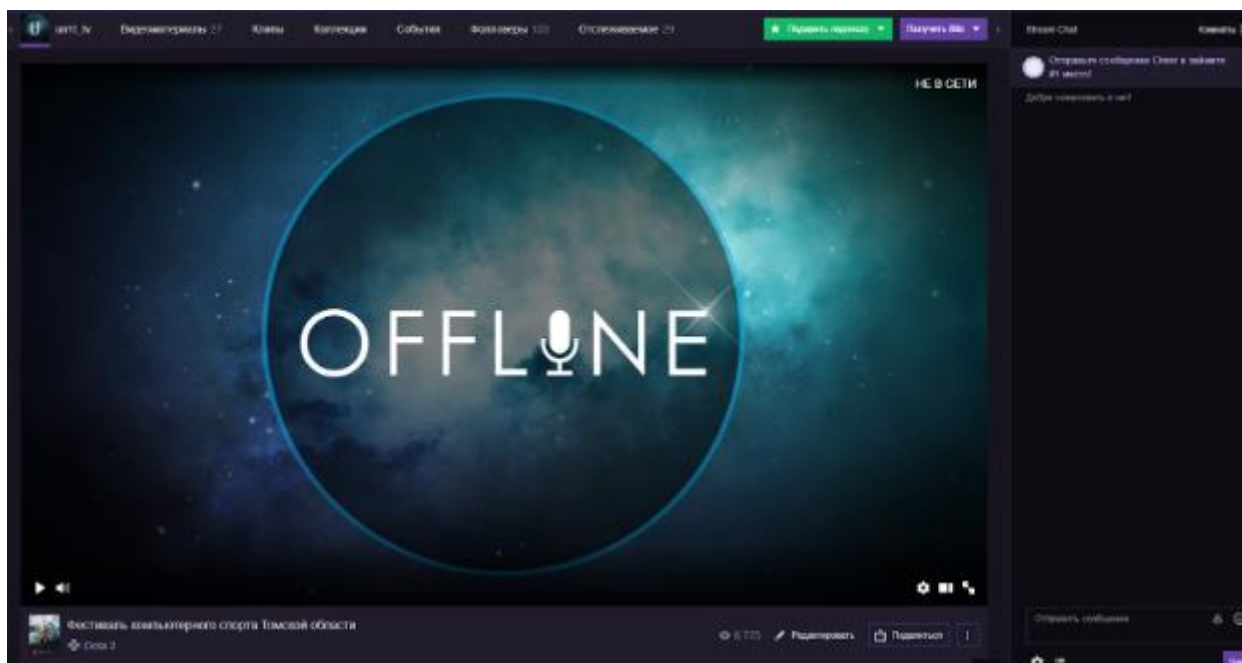


Рисунок 1. Интерфейс платформы Twitch.tv

Таким образом, чат бот – система, которая может создать дополнительные источники контента. Выбранная модель подачи – список автоматически обрабатываемых команд. Список будет предоставляться в описании трансляции, так любой зритель будет иметь доступ к информации о функциональности бота.

Глава 2. Проектирование чат-бота для платформы Twitch.tv

В общем случае, включая контекст социальных сетей, боты классифицируются следующим образом:

- **Технические.** В их структуре присутствуют специально прописанные программы. В их основные задачи входит накопление односложных комментариев под нужной записью, создание большой массы контактов для увеличения доверия к боту и распространение публикаций посредством копирования и повторной публикации. Является наиболее популярной разновидностью.
- **Боевые.** Требуются в первую очередь для снижения репутации или блокировки определенной страницы в социальной сети путем отправления большого количества жалоб и негативных комментариев.
- **Сливные.** Для распространения той или иной информации используются специальные боты, первоначальная модель поведения такого бота соответствует реальному пользователю, но в определенный момент начинают распространять ложную приватную информацию. Впоследствии многие интернет-издания и СМИ будут ссылаться на этот фальшивый источник.
- **Гиперболизирующие.** Это вид социальных ботов, который предназначен для вхождения в доверие к клиентам конкурента заказчика с дальнейшим созданием антирекламы.
- **Интеллектуальные.** Эти боты интересны тем, что используют собственный вложенный интеллектуальный ресурс. Программа, с необходимыми заложенными данными отправляется на информационную войну для пропаганды стороннего мнения.

Главная задача такого бота - общение на узкоспециализированную тематику. Также бот переходит на оскорбления или провокационные высказывания в отношении других пользователей, тем самым отвлекая внимания от основной темы беседы. Этот вид бота наиболее популярный при обсуждении политических и социальных явлений.

- **Ассистенты.** Программная система с заложенными алгоритмами обработки определенного набора команд. Обращение происходит по ключевому слову команды, список которых заранее предоставляется пользователю.

Исходя из предметной области, целевой моделью программной системы является бот-ассистент. Модель обработки команд в чате эфира представлена в виде контекстной диаграммы IDEF0 (Рисунок 2).

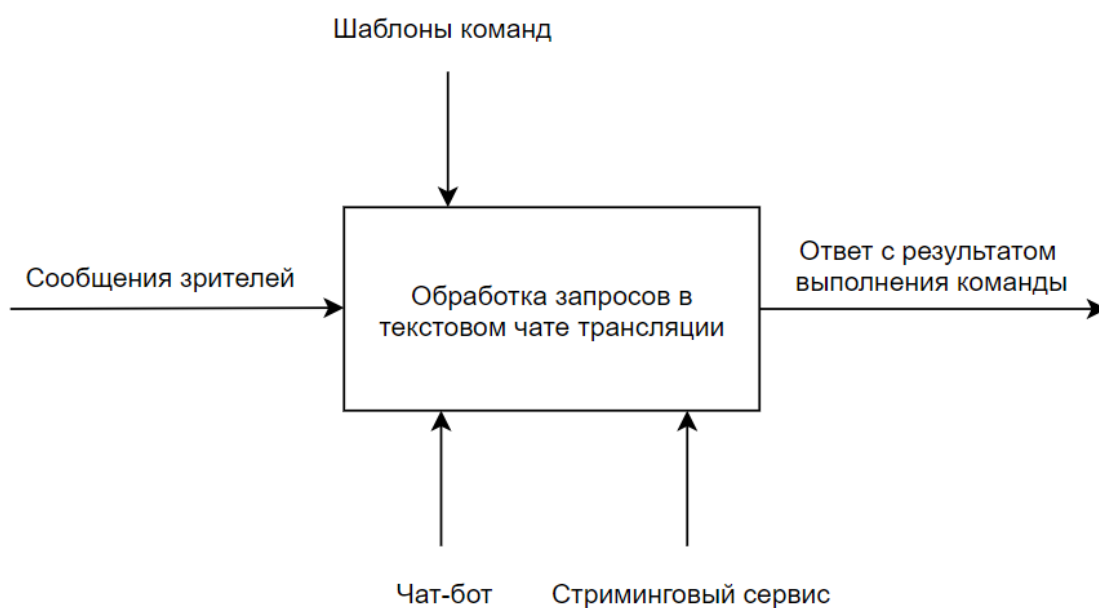


Рисунок 2. Модель обработки команд чат ботом

Для более детального определения решаемых проблем был проведен семантический анализ причин, изображенный на диаграмме Fishbone (Рисунок 3).



Рисунок 3. Семантический анализ причин

Далее необходимо провести ранжирование факторов, влияющих на бизнес-процесс, по следующим критериям:

- Влияние на показатель монетизации канала (Критерий 1)
- Решаемость с точки зрения программной системы (Критерий 2)

Оценка производится по шкале от 1 до 10, данные представлены в таблице 1. Стоит отметить, что краеугольный камень успеха канала – ведущий, это закладывается при оценке, из-за чего ни один из факторов не может получить максимальную оценку по какому-либо критерию.

Таблица 1 – Оценка факторов, влияющих на бизнес процесс

Фактор	Критерий 1	Критерий 2	Сумма
Большое количество трансляций на платформе	4	1	5
Низкое разнообразие контента	7	7	14
Большие массы зрителей следуют трендам	6	0	6
Ограниченная функциональность трансляции	6	9	15

Исходя из ранжирования можно утверждать, что использование программной системы на потоковой онлайн-трансляции наиболее актуально для решения проблем низкого разнообразия контента и ограниченной функциональности платформы.

Взаимодействие с платформой должно быть организовано через открытый API, позволяющий обрабатывать поток данных в чате, получать информацию о авторах сообщений, а также название канала, в эфире которого это сообщение пришло. Таким образом, проектируемая программная система создает новый процесс, представленный на диаграмме IDEF0 (Рисунок 2). Данный процесс создает новые действующие лица в процессе: администратор системы, новый пользователь системы.

Также подразумевается формирование количественных показателей, зависящих от общего времени просмотра зрителем эфиров данного канала. Таким образом, создается дополнительный источник мотивации для длительного просмотра одного канала длительное время, вследствие чего повышается лояльность зрителя. Показатели будут рассчитываться во время

обработки команды с использованием неизменяющихся данных о пользователе, таким образом избегается устаревание информации, хранящейся в базе данных. Ключевой количественной информацией является общее количество часов, которые зритель потратил на просмотр эфиров данного канала, это позволит создать глобальный «прогресс» пользователя в рамках канала, это повышает общую мотивацию и лояльность зрителя.

Целями оказания услуг являются:

- повышение вовлеченности аудитории;
- повышение лояльности аудитории;
- повышение показателей монетизации онлайн-трансляции;
- повышение конкурентоспособности трансляции;

Для достижения указанных целей в рамках оказания услуг должны быть решены следующие основные задачи:

- Повышение разнообразия контента трансляции с помощью различных команд в чате трансляции;
- Обработка данных по общему времени просмотра пользователем конкретной трансляции;
- Возможность использования бота любой трансляцией на платформе.

2.1. Требования к услугам обеспечения качества

2.1.1. Общие требования к составу, содержанию и порядку оказания услуг

В результате анализа предметной области и доступных аналогов, были сформулированы следующие основные требования:

- экспертный анализ документации;
- функциональное тестирование чат-бота;
- API-тестирование чат-бота;
- UX-исследование чат-бота;
- Обеспечение открытого доступа к системе в сети Интернет;

2.1.2. Функциональные требования

1. Пользовательский интерфейс должен обладать возможностью ввода названия канала трансляций для подключения к нему чат-бота;
2. Чат-бот должен авторизоваться в API стримингового сервиса и быть готовым к работе не позднее, чем через 7 секунд после запуска программы на сервере;
3. Чат-бот должен поддерживать одновременное соединение с 3000 каналов трансляций;
4. В пользовательском интерфейсе должна предоставляться информация о текущем количестве каналов, использующих чат-бота;
5. При разрыве соединения с сервером стриминговой платформы, приложение автоматически должно попытаться присоединиться к серверу каждые 10 секунд, начиная с 5ой секунды после разрыва;

6. При трех неудачных попытках восстановления соединения приложение должно оповестить администратора системы о недоступности API;
7. Приложение должно работать с командами в чате, начинающимися с символа «!». Пример: !uptime – по данной команде приложение предоставляет ответ с информацией о времени трансляции (эфира);
8. Приложение должно сохранять список каналов для подключения в базе данных для возможности восстановления состояния после непредвиденного отключения;
9. Чат-бот должен предоставлять администратору возможность изменения адреса ресурса, с которого берется RSS-лента новостей.
10. Приложение должно отправлять в чат случайную новость с указанной в настройках RSS ленты по команде «!feed»;
11. Ответное сообщение с новостью из RSS ленты должно содержать сокращенный URL-адрес новости в исходном ресурсе;
12. Любое сообщение чат-бота должно содержать менее 400 символов;
13. При первичном запуске чат-бота должна проверяться доступность используемых удаленных ресурсов, в случае неудачного соединения – логирование ошибки в консоли и прекращение работы.

14. Чат бот должен обрабатывать команду «!echo» - ответом на данную команду должно быть сообщение «Tiberius here» - команда позволяет проверить наличие чат-бота на трансляции.

2.1.3. Нефункциональные требования

1. Чат-бот должен обладать клиентской частью – SPA приложение на основе ReactJS;
2. Интерфейс клиентской части должен корректно отображаться при масштабировании окна браузера;
3. Интерфейс клиентской части должен корректно отображаться при масштабе окна браузера 100%, 75%, 50%;
4. Клиентская часть должна поддерживаться браузерами Chrome (11.0.696 и выше), Opera (6 и выше), Firefox (53.0 и выше), Safari (5.1 и выше);
5. Сообщения в чате должно отображаться с указанием имени пользователя, который сделал запрос, с префиксом «@» (пример – «@username, ваша команда обработана»);
6. Новости из RSS ленты должны отправляться отформатированными по тегам;
7. Новость из RSS ленты должно делиться красной строкой на 2 части – заголовок и краткое содержание;
8. В конце краткого содержания статьи должна прилагаться ссылка на её полный вариант;
9. Запросы от одного пользователя в рамках одного канала должны обрабатываться с интервалом 5 секунд, команды, отправленные в этот промежуток, игнорируются.

- 10.Серверная часть чат-бота должна использовать не более 2 Гбайт оперативной памяти;
- 11.Клиентская часть чат-бота должна использовать не более 80 Мбайт оперативной памяти;
- 12.Поле ввода для названия канала в пользовательском интерфейсе не должно предлагать ранее введенные варианты (autocomplete - off).
- 13.Пользовательский интерфейс должен оповестить пользователя о результате запроса на подключение чат-бота к введенному каналу. Положительный результат – зеленое всплывающее уведомление в верхнем правом углу экрана с текстом «Подключение успешно». Отрицательный результат – красное всплывающее окно в верхнем правом углу экрана с текстом «Не удалось подключиться к каналу»;

В условиях интеграции приложения с интерфейсом платформы потокового вещания, необходимо спроектировать процесс обработки команд пользователей. Данный процесс иллюстрируется в нотации IDEF0 (Рисунок 4), также приведена частная диаграмма процесса обработки команды в соответствии с хранимой информацией об общем времени просмотра контента текущего канала (Рисунок 5).

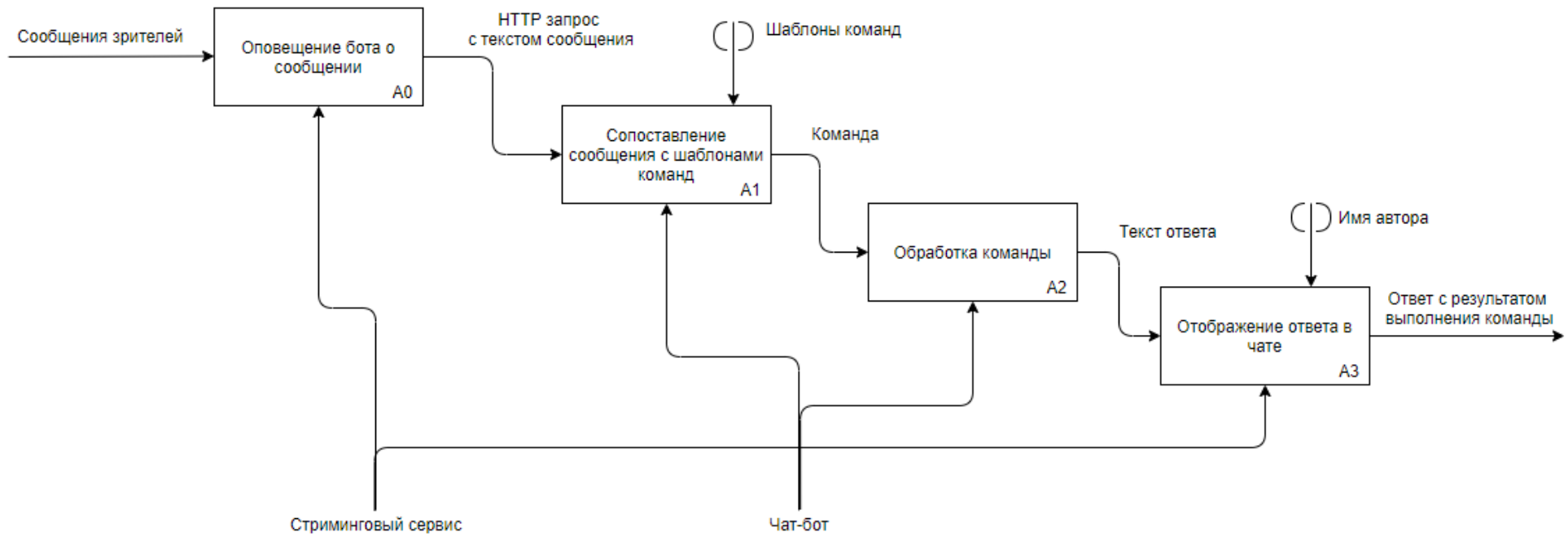


Рисунок 4. Процесс обработки команд в условиях интеграции с платформой потокового вещания

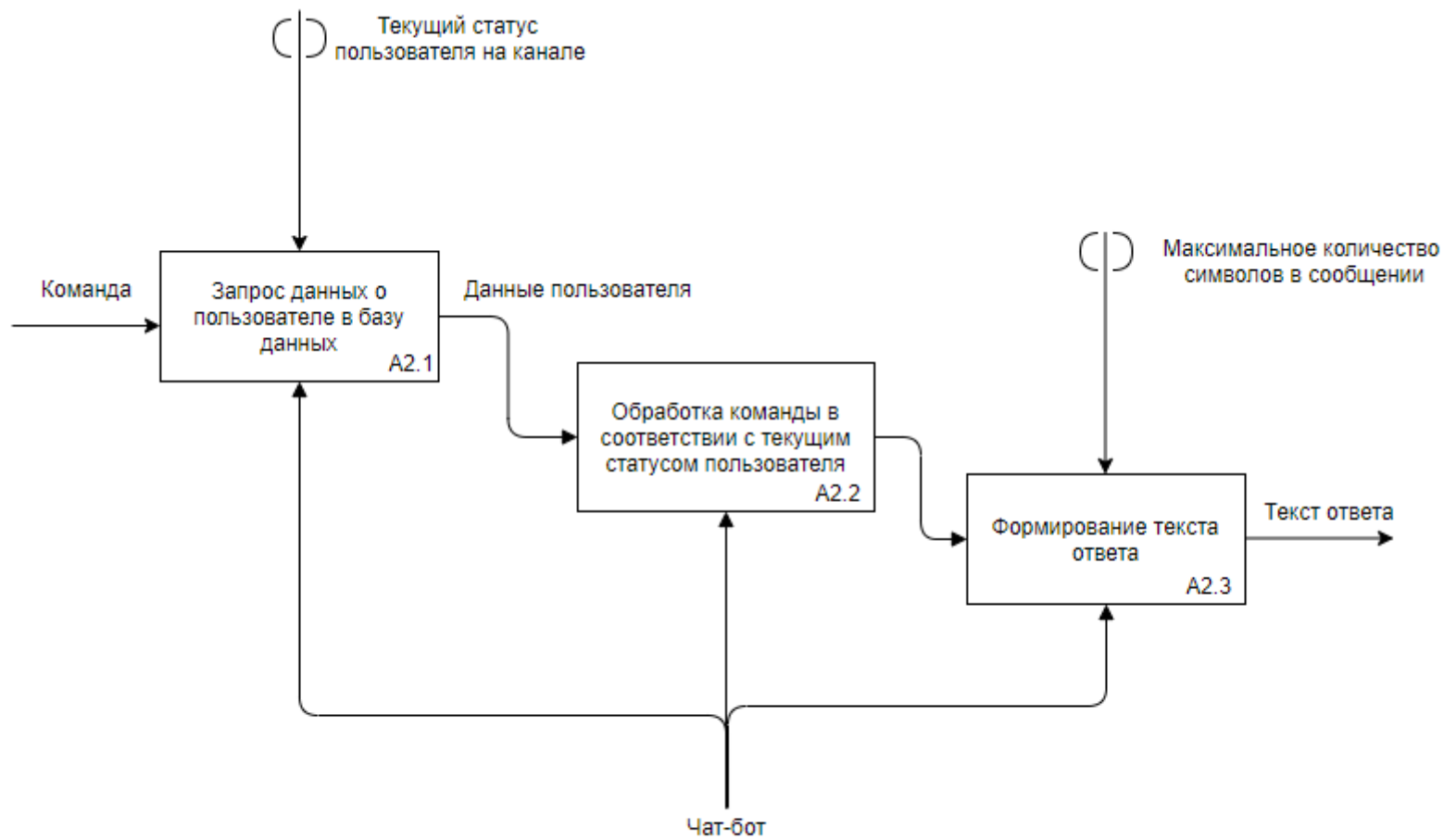


Рисунок 5. Процесс работы с данными пользователя при обработке команд

При проектировании данной информационной системы необходимо учесть последовательность процессов, алгоритм работы с данными, возможные хранилища этих данных, условия при которых та или иная операция будет совершена.

Логика взаимодействия элементов системы, последовательность процессов и их условность изображены на диаграмме IDEF3 (Рисунок 6). Логический узел J2 отображает разветвление процессов в зависимости от результата сопоставления сообщения пользователя с существующими шаблонами команд.

Диаграмма в нотации BPMN (Рисунок 7) описывает бизнес-процесс более подробно, а также позволяет отобразить сферу ответственности каждого компонента системы и внешних объектов.

Так как блок «Обработать команду» варьируется в зависимости от выполняемой команды, в качестве примера рассмотрена команда с наиболее сложным алгоритмом и наибольшим количеством используемых внешних объектов относительно других функциональных возможностей системы – получение случайной новости с ресурса «Habr» (Рисунок 7). Данный ресурс выбран в качестве источника контента, так как имеет научно-технологическую направленность, что соответствует целевой аудитории стриминговой платформы. В будущих версиях информационной системы источник информации для этой команды может быть сконфигурирован. Важно отметить, что «Habr» использует распространенный подход новостных ресурсов – RSS ленту. Это позволяет стороннему программному обеспечению в автоматизированном режиме запрашивать и использовать информацию с ресурса, единственной дополнительной операцией является алгоритм считывания информации из XML-объекта (блок «Десериализовать ленту» на Рисунке 8).

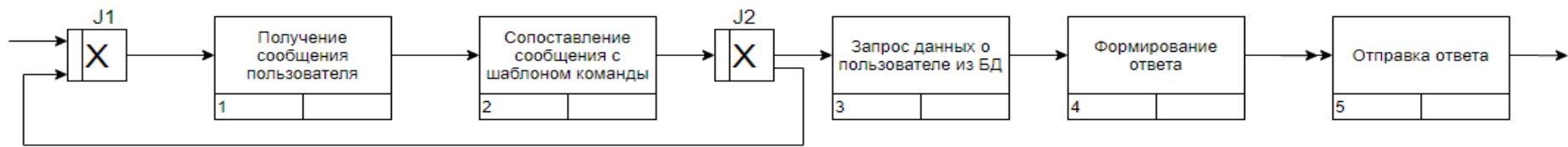


Рисунок 6. Моделирование последовательности процессов

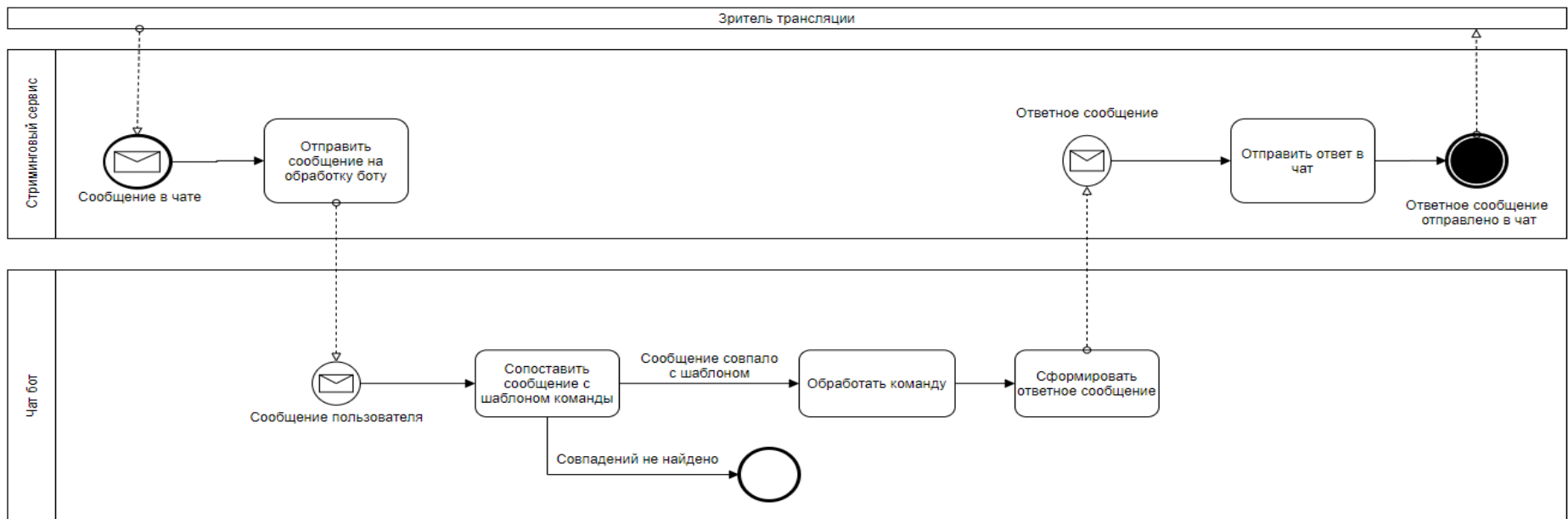


Рисунок 7. Бизнес-процесс в нотации BPMN

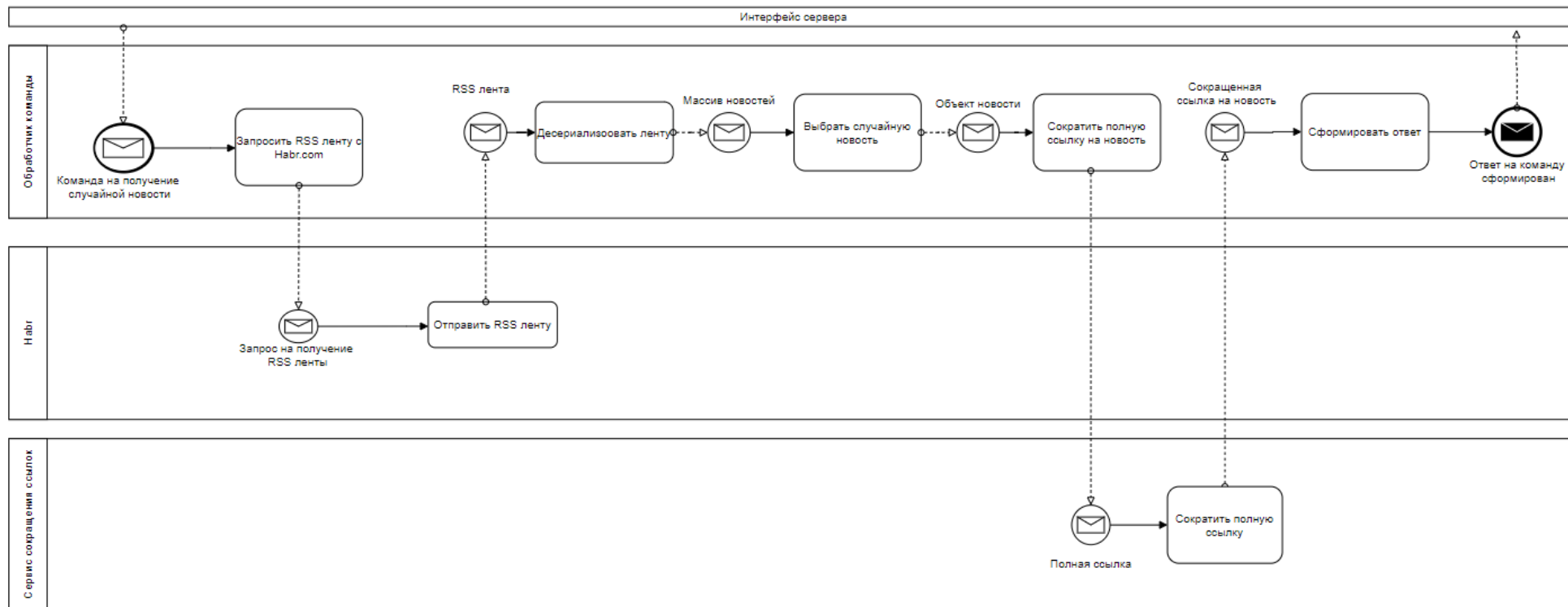


Рисунок 8. Процесс работы с данными пользователя при обработке команд

Другой важный аспект проектирования архитектуры информационной системы – работа с данными, их поток между компонентами системы.

В системе было выделено 6 сущностей:

1. Эфир – имеет в качестве свойств название канала, дату начала, дату окончания и ссылку на повтор;
2. Канал – название канала на стриминговой платформе, сообщения в чате которого обрабатываются чат ботом;
3. Сообщение – содержит в себе текст сообщения и логическое значение, являлось ли это сообщение командой;
4. Команда – автономная сущность, используемая для сопоставления с сообщениями пользователей;
5. Пользователь – имеет связь с сущностью канала, так как информационная система будет регистрировать общее количество часов просмотра эфиров на текущем канале;
6. Звание – в зависимости от описанного выше количества часов на канале, у пользователя будет возможность повышения до определенного ранга, при этом часы просмотров конвертируются в очки опыта для добавления элемента геймификации.

Разные компоненты системы будут работать и обмениваться разными сущностями, потоки данных в системе и процессы обмена ими между объектами были описаны в нотации DFD. Взгляд на всю систему в масштабе

представляет контекстная диаграмма потоков данных (Рисунок 9).

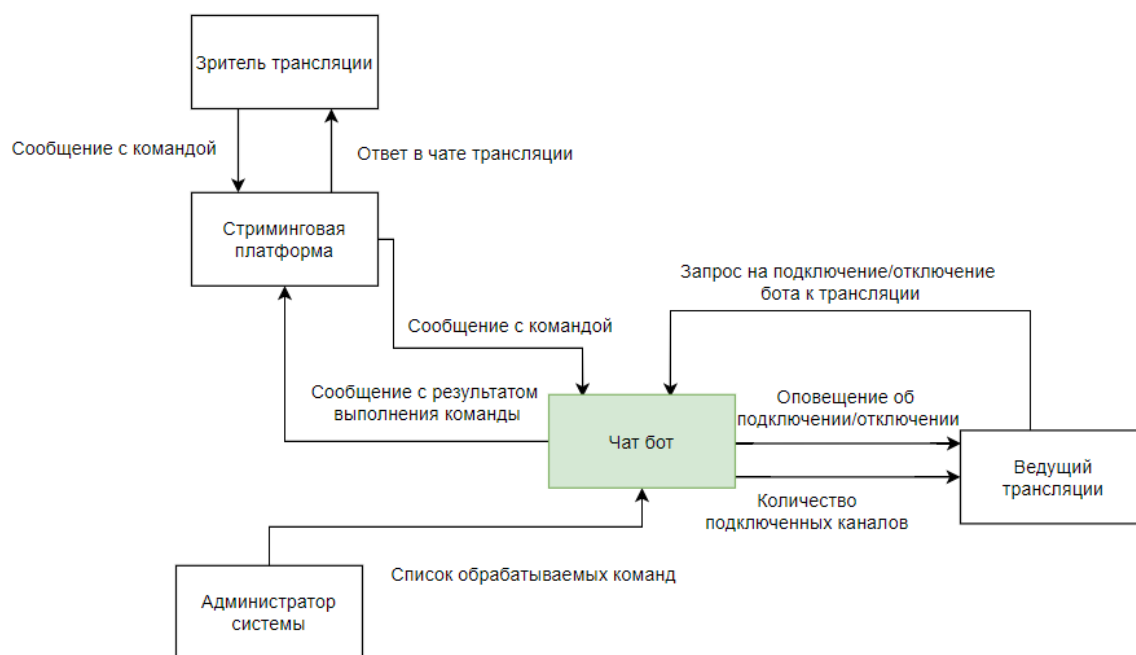


Рисунок 9. Контекстная диаграмма потоков данных

Более подробно процессы обмена данными, интерфейсы, которые используют участники бизнес-процесса, а также их взаимодействие с хранилищами данных описывает подробная диаграмма потоков данных (Рисунок 10). Из неё становится ясно, что у пользователей есть два интерфейса для взаимодействия с информационной системой: интерфейс сайта чат бота и интерфейс сервера.

На данном этапе можно выделить отдельный подпроцесс – проверка критического условия на максимальное количество символов в сообщении. Более подробно этот процесс рассмотрен на частной диаграмме потоков данных (Рисунок 11).

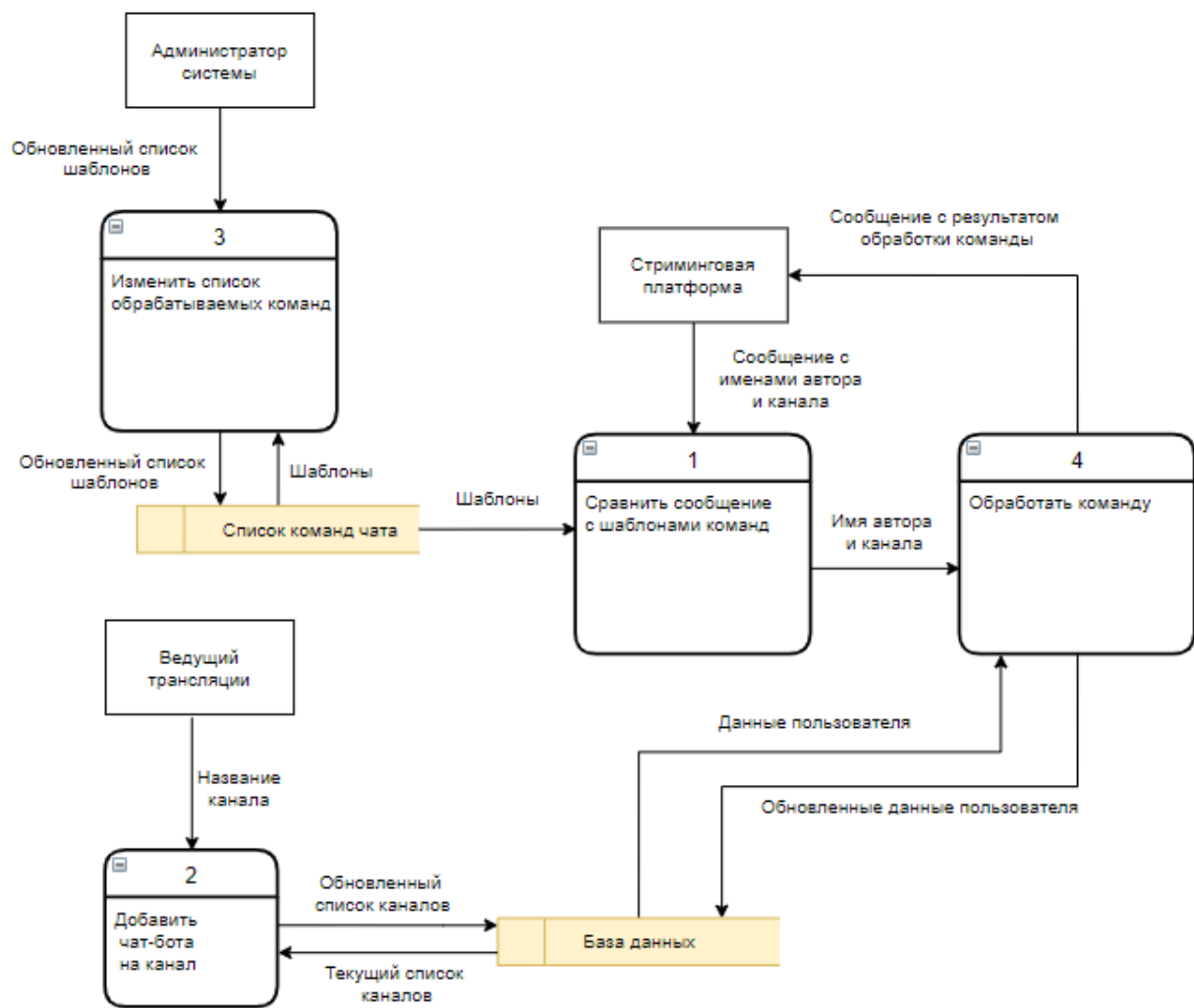


Рисунок 10. Частная диаграмма потоков данных

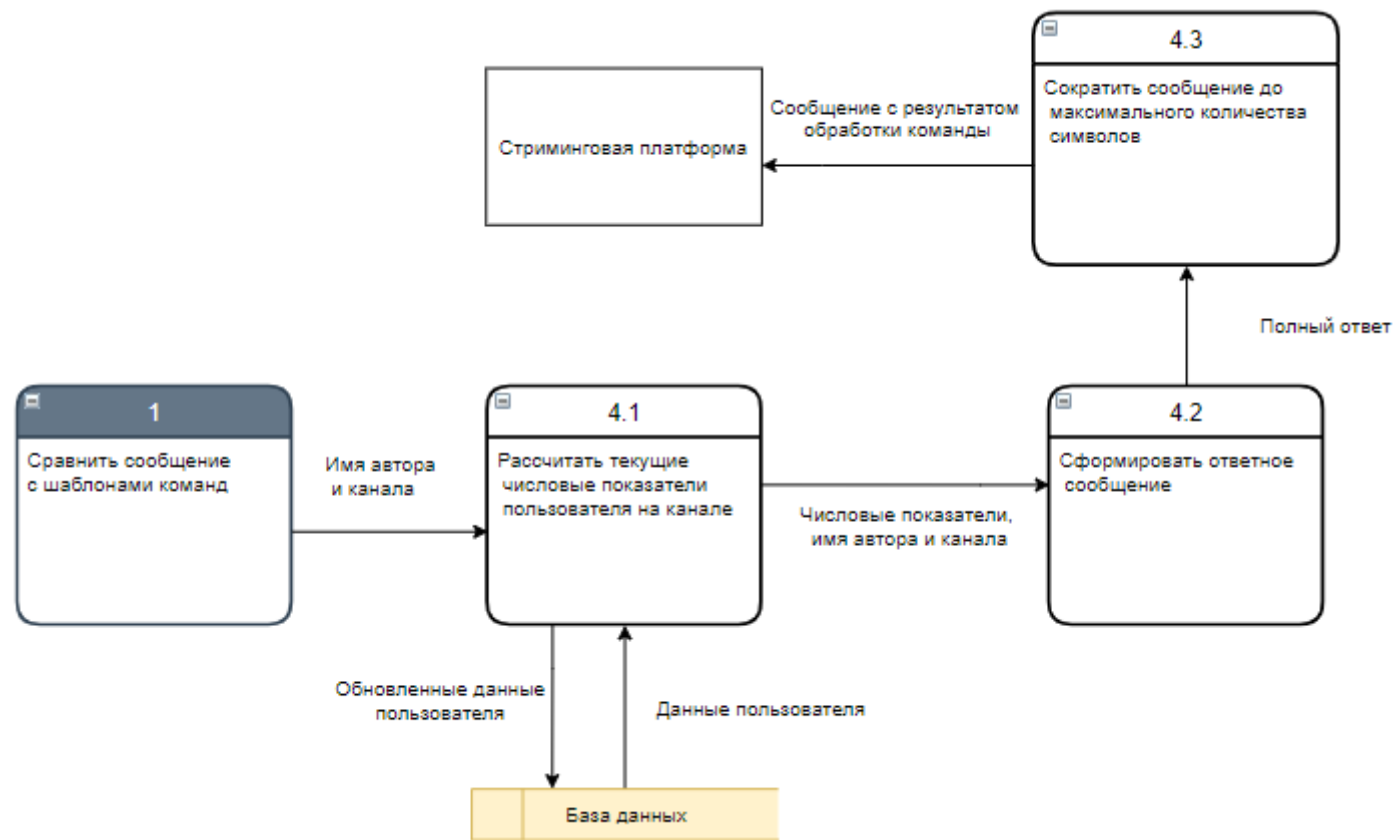


Рисунок 11. Подробная диаграмма потоков данных

При проектировании информационной системы необходимо учитывать события, возникающие в течение бизнес-процесса. Формулирование и четкое понимание этих событий позволяет различить ответственности объектов системы, выявить описать причинно-следственную связь между исходным состоянием и результатом. Данный вид моделирования системы представлен на диаграмме в нотации EPC (Приложение А). На диаграмме видно, что большинство событий в рамках процесса обработки команды происходят в пределах проектируемой программной системы, это означает минимизацию нагрузки на клиентов – зрителей трансляции и ведущего.

Глава 3. Обоснование выбора технологий

3.1 Обоснование выбора программных средств разработки

Необходимо выбрать универсальную среду разработки для работы как с серверной, так и с клиентской частью приложения. Обе части приложения разрабатываются на высоком языке программирования TypeScript, поддержка и инструменты работы с которым являются определяющим фактором в данном выборе. Для принятия наилучшего решения был проведен морфологический анализ (Таблица 2).

Таблица 2 – Морфологический анализ

Метрика	Вес метрики (макс.10)	VS Code	WebStorm	Atom
Стоимость	10	10	4	10
Наличие синтаксического анализа кода TypeScript	10	6	10	6
Инструменты работы с базами данных	5	2	9	9
Наличие инструментов работы с Git	5	7	10	7
Опыт работы со средой	2	5	10	0
Сумма	320	215	255	240

Шкала оценок:

- «10-9» – отлично;
- «8-6» – хорошо;
- «5-3» – удовлетворительно;
- «2-0» – неудовлетворительно;

Для расчета суммарной оценки технологии с учетом веса, была использована следующая формула:

$$\text{Сумма} = \sum_{i=1}^n (\text{вес метрики} * \text{оценка технологии}),$$

где n – количество технологий.

По результатам морфологического анализа, представленного в таблице 1, был выбрана среда разработки Webstorm. Основные преимущества данной среды – инструменты работы с базой данных, синтаксическая поддержка TypeScript без использования дополнительного ПО.

3.2 Фреймворк серверной части

При выборе фреймворка серверной части приложения также необходимо провести семантический анализ среди популярных решений для платформы NodeJS (Таблица 3) [4][5].

Таблица 3 – Морфологический анализ

Метрика	Вес метрики (макс.10)	Express	Koa	NestJS
Типизация TypeScript	10	0	0	10
Масштабируемость	10	5	6	10
Изначально определенная архитектура приложения	4	2	8	9
Объединение паттернов объектно-ориентированного и функционального программирования	3	6	5	6
Встроенные инструменты тестирования и отладки	8	4	8	8
Низкая связность компонентов архитектуры	7	2	7	8

Сумма	420	122	220	338
-------	-----	-----	-----	-----

Для обеспечения масштабируемости приложения, в качестве фреймворка серверной части выбран Nestjs – структура, диктуемая данным фреймворком, позволяет разбить приложение на логические части (модули), каждая из которых состоит из двух объектов [6]:

- Сервис – содержит в себе функции работы с данными;
- Контроллер – содержит структуру API модуля

Композиция функциональности в данную структуру также понижает порог вхождения для работы с проектом, что может понадобиться при открытой разработке и расширении команды.

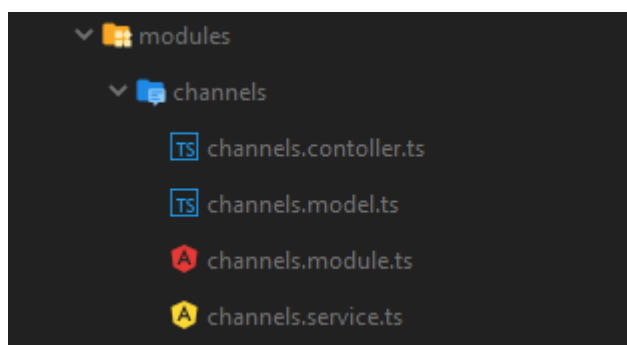


Рисунок 12. Структура модуля NestJS

Nestjs предполагает разработку на языке TypeScript, типизация в данном языке обеспечивает безопасность кода и устранение ошибок до запуска программы.

3.3 СУБД

При выборе СУБД необходимо учитывать особенности хранимых данных – низкая сложность структур и большое количество записей. Анализ альтернатив приведен в таблице 4 [7].

Таблица 4 – Морфологический анализ

Метрика	Вес метрики (макс.10)	PostgreSQL	Mongo	MySQL
Ориентированность на работу с JSON объектами	8	6	10	2
Поддержка сообщества	7	9	6	6
Наличие бесплатной версии	9	2	8	10
Ориентированность на Web приложения	10	7	10	4
Исчерпывающая документация	8	4	7	7
Сумма	420	231	350	244

Так как приложение не предполагает хранения сложных структур, а объекты языка TypeScript легко преобразуются в объекты JSON формата, в качестве СУБД выбрано noSQL решение – MongoDB. Веб-сайт mlab.com предоставляет бесплатное MongoDB хранилище, что также является плюсом для уменьшения бюджета проекта.

3.4 Клиентское приложение

Технологии представления клиентской части веб-приложений за последние 10 лет кардинально изменились. Клиентские интерфейсы стали включать значительно больше элементов управления, данные в блоках веб-страницы могут частично дублироваться, контроллеры интерактивных элементов стали включать значительно усложненную логику изменения содержимого других элементов. Поэтому сложность добавления новых возможностей возрастает нелинейно, для упрощения процесса разработки необходимо использовать более современный подход – концепцию *реактивных данных*. При такой концепции разработчики более не работают напрямую с реальным DOM, его состояние инкапсулирует виртуальный DOM. Реактивные фреймворки отслеживают изменения в

содержимом виртуального DOM и автоматически определяют компоненты, состояние которых необходимо изменить и перестроить [8]. Такой подход позволяет разработчику внедрять новые возможности в интерфейс пользователя, абстрагируясь от взаимодействия существующих компонентов. На рисунке 25 продемонстрировано взаимодействие между родительскими и дочерними компонентами в классическом реактивном приложении.

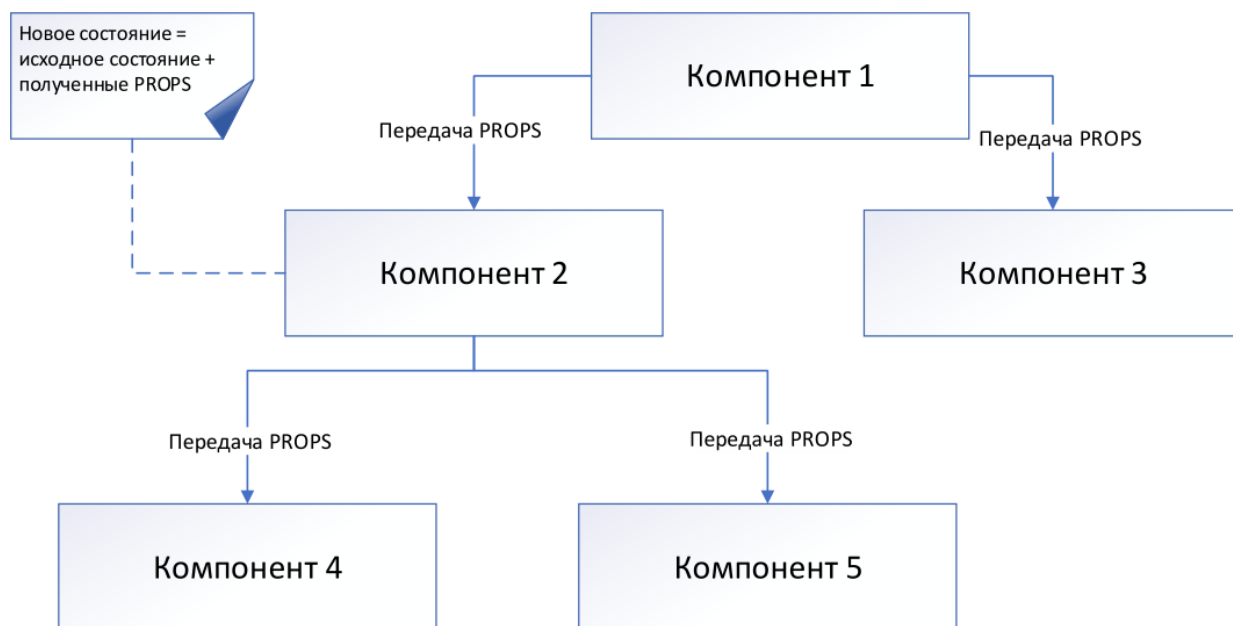


Рисунок 13. Передача данных между реактивными компонентами

Пользователь взаимодействует с корневым компонентом, изменяя его состояние, и тот реактивно распространяет изменения на дочерние компоненты, передавая им новую порцию данных (props) из которой будет строиться их новое состояние. Если их состояние изменится, процесс повторится для компонентов 4 и 5.

На данный момент доступно несколько решений для реализации реактивности данных в клиентской части веб-приложений – Angular, React, Vue. Команда разработчиков выделила показанные в таблице 7 критерии оценки реактивных фреймворков исходя из особенностей работы разрабатываемой системы (Таблица 5) [9].

Таблица 5 – Критериальное оценивание реактивных фреймворков

Критерий	Вес метрики	React	Angular	Vue
Количество языковых конструкций, не включенных в стандартный синтаксис JS	8	7	2	5 – новые конструкции для вывода коллекции
Возможность определения разметки компонента в основной функции	3	3	1	1
Поддержка и улучшение продукта	6	6 – поддерживается Facebook'ом	4 – отток комьюнити, поддерживается Google'ом	1 – широко распространено только в Китае
Производительность решения	7	7 – оптимизация отрисовки функциональных компонентов	5 – больше инструкций в собранном скрипте из-за транспиляции	7
Возможности по анимации компонентов	5	3	3	5
Сумма		26	15	23

Автором было принято решение использовать реактивный фреймворк React для реализации клиентского интерфейса из-за высокой поддержки программного продукта и использованию парадигм функционального программирования, призванного сделать программный код более декларативным, читаемым и предсказуемым.

3.5 Облачный сервис

«Облачность» предполагает возможность предоставления ресурсов с высокой гарантией их наличия. Иными словами, если аппаратные ресурсы на части физических компьютеров облака выйдут из строя или будут отключены, облако как целое может продолжать функционировать (в ряде случаев без существенной потери эффективности), или даже восстановить работоспособное состояние автоматически («самоизлечение», self-healing). Площадка для развертывания должна отвечать следующим критериям:

- Бесплатное размещение приложения
- Поддержка среды Nodejs
- Информация для обучения, документация

Таким образом, необходимо провести семантический анализ по выделенным критериям среди наиболее популярных решений среди облачных хостингов: Heroku, Digital Ocean, Amazon Web Services (AWS) (Таблица 6) [10].

Таблица 6 – Критериальное оценивание реактивных фреймворков

Критерий	Вес метрики	Heroku	Digital Ocean	AWS
Бесплатное размещение приложения	9	10	8	8
Поддержка среды NodeJS	10	10	9	10
Информация для обучения, документация	6	8	6	9
Сумма	250	238	198	226

Согласно приведенным критериям, был выбран сервис Heroku, также позволяющий осуществить интеграцию с git-репозиторием и настройку автоматического деплоя при изменении кода.

Глава 4. Разработка приложения

4.1. Сервер

Из обычного консольного приложения, чат-бот был преобразован в полноценный сервер на Nestjs. Для реализации функциональности был создан модуль channels (Рисунок 12). API модуля поддерживает следующие пути взаимодействия:

- GET запрос – возвращает массив каналов, к которым осуществлено подключение;
- POST запрос – позволяет добавить новый канал в базу данных и немедленно создать новый экземпляр соединения;
- DELETE запрос – выполняет функцию удаления канала из базы, соединение моментально прерывается.

Для поддержки типизации TypeScript, реализован ряд интерфейсов. Примером является интерфейс сущности соединения с каналом (Рисунок 14) [11].

```
export interface ClientEntity {  
  name: string;  
  client: any  
}
```

Рисунок 14. Интерфейс клиента подключения

Тип `any` не является рекомендуемым решением для типизации в TypeScript, но в данном случае является необходимым, т.к. библиотека `tmjjs`, предоставляющая функциональность взаимодействия с Twitch.tv, не поддерживает TypeScript и не является типизированной.

Экземпляры соединения с каналами трансляции хранятся в массиве `clients`, уникальным идентификатором элемента в котором является название

канала. Запрос на добавление/удаление с клиентской части инициирует изменение данного массива, а затем данные отправляются в базу данных.

Работа с базой MongoDB осуществляется с помощью ORM Mongoose, и её адаптацией для Nestjs @nestjs/mongoose. При использовании данного инструмента необходимо описать схему данных, которые хранятся в базе данных (Рисунок 15), затем происходит инъекция этой структуры в сервис модуля Nest (Рисунок 16), что позволяет статически обращаться к сущностям базы данных (Рисунок 17) в любом месте сервиса.

```
export const ChannelSchema = new mongoose.Schema({
  name: {type: String, required: true, unique: true},
});
```

Рисунок 15. Инициализация системы данных

```
@Injectable()
export class ChannelsService {
  private clients: ClientEntity[] = [];

  constructor(@InjectModel('Channel') private readonly channelModel: Model<Channel>) {
    channelModel.deleteMany({}, () => {
      console.log('channels collection is clear')
    });
  }
}
```

Рисунок 16. Инъекция схемы и её модели в сервис модуля

```
async addChannel(name: string) {
  const newChannel = new this.channelModel({ name });
  return await newChannel.save((err) => {
    console.log(`Unable to add channel ${name} to db, ${err}`)
  }).then( onfulfilled: () => name);
}
```

Рисунок 17. Метод добавления нового канала со статическим обращением к схеме

После изменения сущностей в базе данных, можно выполнить проверку с помощью пользовательского интерфейса сервиса mlab, после авторизации под личной учетной записью, предоставляется возможность

просмотра коллекций MongoDB и данных в них. Приложение использует коллекцию channels (Рисунок 18).

Для подключения приложения к базе данных, необходимо использовать уникальный адрес подключения, доступ к которому также осуществляется через интерфейс сервиса.

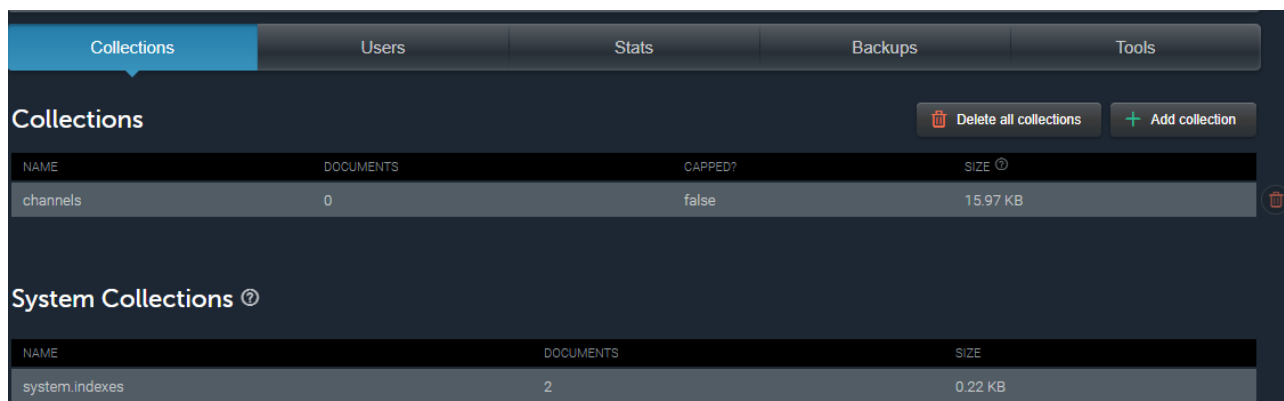


Рисунок 18. Интерфейс сервиса mlab

Обработка команд в чате организуется с помощью шаблона проектирования «цепочка ответственностей». При создании каждого экземпляра соединения с каналом инициализируется цепочка обработчиков, в которой каждый предыдущий имеет ссылку на следующий. Если обработчик распознал в сообщении команду, за которую он несет ответственность, вызывается соответствующая функция, если команда не найдена, сообщение передается следующему исполнителю в цепочке. В случае, если ни один из обработчиков не отреагировал на сообщение, оно игнорируется приложением.

4.2. Клиентская часть

Разрабатываемая программная система чат-бот требует тщательного подхода к выбору технологий и методологий для разработки клиентской и серверной части. Это обусловлено рядом причин, одна из которых - необходимость организовать клиентскую часть так, чтобы для пользователей с разными ролями оставался один и тот же собранный файл – bundle,

использовался единый дизайн, но при этом был реализован различная функциональность.

Также разрабатываемое веб-приложения должно соответствовать всем современным требованиям к разработке клиентской части, т.е. являться легковесным, реактивным, производительным, легко модифицируемым в соответствии с внезапно появляющимися бизнес-требованиями.

Для решения задачи с гибкостью приложения было решено реализовать модульную архитектуру. Данный подход позволяет в кратчайшие сроки менять структуру веб-приложения, добавлять новый функционал, при надобности избавляться от уже существующего функционала, а также модифицировать его.

В клиентской части веб-приложения была реализована структура директорий, содержащая WebPack, папку src и node_modules.

Директория webpack содержит в себе файлы настроек с расширением .js. Данные файлы позволяют настроить сборки проекта для различных целей и указывают сборщику, как собирать проекты в версии для клиентов и как во время разработки; какие применять настройки для сборки браузерной версии веб-приложения и какие для сборки мобильной версии. В файлах настроек указано, какие загрузчики необходимо применять для файлов из структуры проекта с различными расширениями. В конечном итоге задача данных файлов настроек заключается в том, чтобы превратить структуру проекта, удобную для разработки, в три файла: index.html, index.js, index.css. Три файла браузеры пользователей будут гораздо быстрее загружать с сервера, также сборщик сжимает код, что делает общий вес приложения в разы меньше исходного. Также система сборки, руководствуясь файлами настроек, выполняет одну из своих главных задач – транспиляцию кода.

При этом в результате клиент получит файл, использующий только язык JavaScript. Причем этот файл будет использовать исключительно

синтаксис старых версий языка на ECMaScript 5, что позволит работать веб-приложению со всеми браузерами, в том числе и считающимися устаревшими, а также на любых устройствах, начиная с мобильных телефонов и заканчивая персональными компьютерами.

Директория `node_modules` необходима для хранения библиотек, используемых в приложении. Процесс установки библиотек и фреймворков происходит посредством NPM [12]. NPM позволяет использовать большое количество современных библиотек, что дает возможность подобрать необходимый набор инструментов разработчика при разработке конкретного приложения. Установка пакетов происходит посредством команды `npm install <имя пакета>`. После запуска данной команды в командной строке, NPM автоматически подгружает в проект указанную библиотеку или иной инструмент разработчика. Также есть возможность установить пакет определенной версии в том случае, если вам необходимо обеспечить совместимость тех или иных средств разработки. Помимо этого, если была найдена достойная альтернатива используемой библиотеке, или в результате разработки было выявлено, что используемая библиотека не является необходимой в данном веб-приложении, с помощью NPM можно легко удалить эту библиотеку, используя команду `npm uninstall <имя пакета>` [13]. Это позволит уменьшить общий вес приложения, что повысит его производительность.

В директории `node_modules` разработчик может посмотреть исходный код любой загруженной библиотеки и, при желании, даже изменить его. Данное преимущество чаще используется в целях изучения работы тех или иных библиотек, а также получения каких-либо знаний в области программирования [14]. Это связано с тем, что большинство библиотек пишутся опытными, умелыми программистами, которые часто используют не совсем популярные, но иногда очень полезные подходы к программированию, о которых не всегда можно найти информацию в сети.

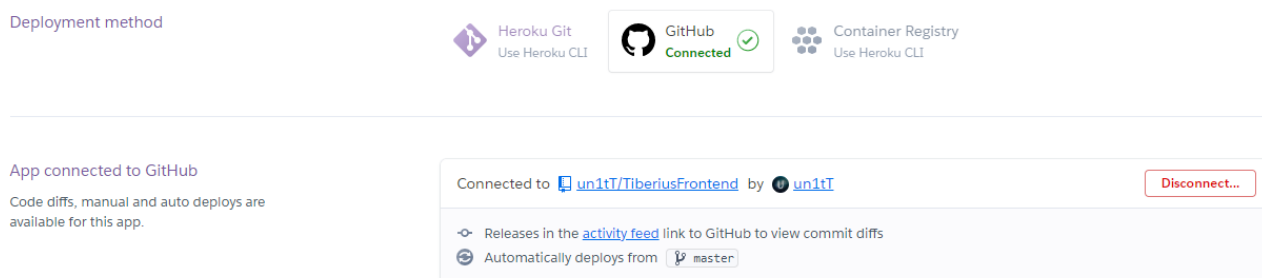
Непосредственно сам программный код веб-приложения располагается в директории `src` .

Для комфортного использования приложения было разработано приложение на `Reactjs`, позволяющее осуществлять добавление и удаление каналов из списка подключения бота [15]. На главной странице находятся поля ввода для осуществления данных операций, а также счетчик количества каналов, которые в данный момент используют чат бота (Приложение Б).

4.3. Деплой в Heroku

Выбранный сервис позволяет осуществить развёртку по упрощённой схеме при настройке интеграции с `git`-репозиторием. В пользовательском интерфейсе необходимо указать адрес репозитория, в котором находится код (Рисунок 19), после этого Heroku приступает к автоматическому деплою.

Рисунок 19. Интеграция с репозиторием



После этого, по умолчанию повторное развертывание проекта происходит при каждом обновлении кода в ветке `master` репозитория.

Для запуска приложения на виртуальной машине Heroku вызывает стандартные скрипты приложения для среды `Nodejs` – `npm start` [16]. Перед его запуском также устанавливаются все зависимости проекта.

В случае с клиентской частью, развертывание не предполагает дополнительных манипуляций, так как `Reactjs` не требует установки глобальных зависимостей.

Глобальная зависимость – утилита, которую необходимо установить на компьютер не в пределах проекта, или. В случае с облачным сервисом, на виртуальную машину. Для корректной сборки и запуска серверного приложения необходима глобальная утилита `ts-node`, компилирующая код TypeScript в JavaScript. Heroku имеет ограничения на установку такого рода зависимостей, поэтому запуск стандартного скрипта не приведет к результату. Данная проблема была решена путём дополнительной подготовки кода перед отправкой на облачный сервер:

Ручная компиляция кода с помощью скрипта `npm build`;

Сохранение этого кода в отдельной директории `/dist`;

Редактирование скрипта `npm start` для запуска входного файла `main.js` в директории `/dist`.

После преобразования, главный скрипт имеет вид: `node dist/main.js`.

Таким образом была решена проблема ограничения глобальных зависимостей Heroku.

Управление развернутыми приложениями осуществляется через личный кабинет пользователя (Приложение В).

Глава 5. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

5.1. Оценка перспективности проведения исследований

5.1.1 Потенциальные потребители результатов исследования

Рост количества и популярности стриминговых платформ за последние годы обусловлен растущим спросом на потребление контента через сеть Интернет. Ожидается, что в период до 2023 года мировой рынок OTT-видео будет расти в среднем на 13,8% в год, а его объем достигнет 72,8 млрд долл. (по сравнению с 38,2 млрд долл. в 2018 году). Пользователей привлекают методы монетизации, ориентированные на клиентов, бесплатные пробные версии контента, доступ к нему с любого устройства.

Целью исследовательской работы является повышение лояльности и вовлеченности аудитории посредством использования программной системы чат бота. Это позволит автоматизировать часть работы ведущего онлайн-трансляции, создать дополнительный источник контента (чат эфира) и, как следствие, увеличить показатели монетизации канала.

Целевой рынок следует разделить на два сегмента – платформы, предоставляющие услуги потокового онлайн вещания и предоставляющие уже готовый контент (сервисы для просмотра сериалов, кинофильмов). Оба сегмента схожи в способах монетизации, но обладают разной целевой аудиторией. Программная система ориентирована на использование в рамках онлайн трансляций, из чего следует, что целевой аудиторией являются, в первую очередь, игроки в видеоигры и поклонники компьютерного спорта. Об этом свидетельствует статистика просмотров на платформе Twitch.tv – к концу 2018 года количество видеоигр, по которым ведутся трансляции превысило 38 000, при этом пиковое количество зрителей одной видеоигры – 1 460 297

человек. Стоит отметить, что платформа находится на 35 месте в мире по количеству одновременных пользователей онлайн.

Программная система ориентирована на взаимодействие с аудиторией с помощью текстового чата трансляции, за все время существования Twitch, зрители отправили более 29 миллиардов сообщений в чат, данный способ коммуникации между ведущим и наблюдателями до сих является преобладающим.

5.1.2. Анализ конкурентных технических решений

Открытый интерфейс платформ и свободное использование стороннего программного обеспечения создает большое количество конкурентов на рынке текстовых ассистентов. Были выделены наиболее популярные решения среди авторов трансляций на Twitch.tv:

Nightbot – ассистент, главной специализацией которого является фильтрация спам сообщений, так же предоставляется функциональность регулярной отправки сообщений в чат (реклама, оповещения и т.д.).

Moobot – одно из популярнейших решений для модерации канала, позволяет наряду с стандартной функциональностью создавать опросы среди зрителей.

Чтобы оценить конкурентоспособность проекта, необходимо составить оценочную карту, для этого выделяются наиболее важные аспекты продукта и сравниваются с аналогами на рынке.

Оценки для каждого параметра выставляются по 10-балльной шкале, при этом оценивается влияние каждой характеристики на продукт в целом. Такой метод позволяет получить оценку продукта в целом, а также сравнить различные продукты по отдельным критериям. Таким образом, можно выяснить сильные и слабые стороны разрабатываемого продукта, что отражено в таблице 7.

Таблица 7 – Оценочная карта для сравнения конкурентных технических решений (разработок)

№ п/п	Продукт	Факторы конкурентоспособности						Итог
		Стоимость возможности использования полного функционала	Скорость обработки запросов	Интеграция со с различным и платформами	Учет общего прогресса пользователя на трансляции	Возможность горизонтального масштабирования	Производительность клиентского интерфейса	
1	Nightbot	4	5	5	6	10	9	6,5
2	Moobot	5	6	5	4	8	8	6
3	Tiberius (разрабатываемый продукт)	10	9	1	9	10	9	8
	b_i	4	7	6	8	9	6	40
	w_i	0,1	0,175	0,15	0,2	0,225	0,15	1,0

Из таблицы видно, что наиболее значимым критерием является возможность горизонтального масштабирования, это связано с тем, что рост пользователей программной системы возрастает в геометрической прогрессии, а значит необходимо обеспечить стабильное функционирование продукта вне зависимости от количества клиентов.

Положительной особенностью архитектуры разрабатываемого приложения является возможность распределения сетевой нагрузки путем добавления новых вычислительных узлов – система на платформе Node.js может быть запущена в нескольких экземплярах, что существенно сказывается на эффективности распределения запросов.

5.1.3. Технология QuaD

С помощью технологии QuaD можно оценить качество программной разработки, а также ее перспективность на рынке.

Таблица 8 – Оценочная карта для оценки качества программной разработки

Критерии оценки	Вес	Баллы	Макс. балл	Отн. знач.	Ср. – взвеш. знач.
Показатели оценки качества разработки					
1. Скорость обработки текстовых команд в чате	0,2	90	100	0,9	0,18
2. Удобство пользования	0,1	70	100	0,7	0,07
3. Качественный пользовательский интерфейс	0,05	75	100	0,75	0,0375
4. Соответствие требованиям	0,15	100	100	1,0	0,15
5. Полнота предоставляемой информации	0,15	90	100	0,9	0,135
Показатели оценки коммерческого потенциала разработки					
6. Оригинальность предоставляемого контента	0,2	90	100	0,9	0,18
7. Перспективность рынка	0,05	70	100	0,7	0,035
8. Законченность работы	0,05	65	100	0,65	0,0325

9.	Цена	0,2	80	100	0,8	0,16
					Итого	0,98

Анализ, произведенный по технологии QuaD, показал, что разработка программного продукта перспективна, так как итоговое показательное значение попадает в диапазон от 80 до 100.

5.1.4. SWOT-анализ

Чтобы выявить достоинства и недостатки продукта с целью дальнейшего улучшения качества необходимо провести SWOT анализ [17].

SWOT-анализ (показан в таблице 9) – 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 Threats (угрозы). Дополнительно все факторы следует проанализировать в парах, например слабые стороны и угрозы

Таблица 9 – SWOT-анализ разрабатываемого продукта

		Внутренние факторы	
Внешние факторы		<p>Сильные стороны</p> <ol style="list-style-type: none"> 1. Возможность эффективного масштабирования благодаря фреймворку Nest и платформе Node.js. 2. Балансировка нагрузки для оптимизации работы сервера. 3. Модульность клиентского интерфейса.. 5. Бесплатное использование полной функциональности продукта. 	<p>Слабые стороны</p> <ol style="list-style-type: none"> 1. Большая клиентская база конкурентов. 2. Использование серверов начального уровня производительности. 3. Отсутствие оптимизации клиентской части приложения для смартфонов. 4. Медленный старт из-за ограниченности рекламного бюджета.

		Внутренние факторы	
	<p>Возможности</p> <p>1. Рост производительности вычислительной техники.</p> <p>2. Попадание продукта на первые строчки выдачи поисковых систем.</p> <p>3. Интеграция продукта с сторонними системами.</p>	<p>Сильные стороны и возможности показывают, что исходной производительности сервера хватит с запасом на начальный этап (до 50000 первых клиентов), модульность системы позволит предлагать пользователям самую современную функциональность.</p>	<p>Исходя из слабых сторон и возможностей, можно предположить экспоненциальный рост числа пользователей, наблюдается сильная зависимость развития от первоначальной активности, приложение может завязнуть на начальном этапе.</p>
	<p>Угрозы</p> <p>1. Преобладание мобильных устройств при использовании клиентского интерфейса.</p> <p>2. Массированные атаки, направленные на отказ в обслуживании.</p> <p>3. Появление более совершенного конкурентного продукта.</p>	<p>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.</p>	<p>Выделенные приложения конкурентов могут неожиданно обрести дополнительную популярность. В будущем существует момент, когда конкуренты обратят внимание на продукт из-за сформировавшейся базы пользователей и могут намеренно выводить части системы из строя.</p>

Самой слабой стороной продукта является отсутствие поддержки корректного отображения клиентской части на мобильных устройствах, интерфейс приложения разрабатывался с целью корректной работы в браузерах. Для нейтрализации данного недостатка в проект необходимо привлечь несколько разработчиков, однако это потребует пересмотра бюджета проекта.

Система привносит новый пользовательский опыт в процесс общения зрителей на трансляции, но всегда остается вероятность появления нового 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 будут существенно снижены.

Наконец, даже выделенной группе разработчиков понадобится более месяца для адаптации приложения для мобильных устройств. Если пользовательские предпочтения неожиданно сместятся в сторону таких платформ, то на период их разработки интерес к продукту может значительно снизиться.

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

5.2. Планирование научно-исследовательских работ.

5.2.1. Структура работ в рамках научного исследования.

Таблица 10 – Структура работ в рамках проектирования и разработки веб-приложения

№ работы	Наименование работы	Исполнитель работы
1	Выбор научного руководителя бакалаврской работы	Щукин Василий Алексеевич
2	Составление и утверждение темы бакалаврской работы	Щукин Василий Алексеевич, Савельев Алексей Олегович

№ работы	Наименование работы	Исполнитель работы
3	Составление календарного плана-графика выполнения бакалаврской работы	Савельев Алексей Олегович
4	Подбор и изучение литературы по теме бакалаврской работы	Щукин Василий Алексеевич Савельев Алексей Олегович
5	Анализ предметной области	Щукин Василий Алексеевич Савельев Алексей Олегович
6	Проектирование информационной системы	Щукин Василий Алексеевич Савельев Алексей Олегович
7	Реализация модулей приложения	Щукин Василий Алексеевич
8	Тестирование приложения	Щукин Василий Алексеевич
9	Написание главы по проектированию приложения	Щукин Василий Алексеевич
10	Написание главы по реализации серверной части приложения	Щукин Василий Алексеевич
11	Написание главы по реализации клиентской части приложения	Щукин Василий Алексеевич
12	Выполнение других частей работы (финансовый менеджмент, социальная ответственность)	Щукин Василий Алексеевич
13	Подведение итогов, оформление работы	Щукин Василий Алексеевич

5.2.2. Разработка графика проведения научного исследования.

Согласно производственному календарю при расчете на 6 рабочих дней в неделю в 2020 году 366 календарных дней, 247 рабочих дней, 119 выходных или праздничных дней. Таким образом, коэффициент календарности на 2020 год вычисляется следующим образом:

$$T_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}} = 1,48$$

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:

$$T_{\text{ож}} = \frac{3 * T_{\text{min}} + 2 * T_{\text{max}}}{5}$$

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 11 содержит рассчитанные временные показатели каждого этапа работы.

Таблица 11 – Временные показатели проведения научного исследования

Наименование работы	Исполнители работы	Трудоемкость работ, чел-дни			Длительность работ, дни	
		Tmin	Tmax	Toж	Tr	Tк
Выбор научного руководителя бакалаврской работы	Щукин Василий Алексеевич	1	7	3,4	3	4
Составление и утверждение темы бакалаврской работы	Щукин Василий Алексеевич	2	4	2,8	3	3
	Савельев Алексей Олегович	2	4	2,8	3	3
Составление календарного плана-графика выполнения бакалаврской работы	Савельев Алексей Олегович	2	4	2,8	3	3
Подбор и изучение литературы по теме бакалаврской работы	Щукин Василий Алексеевич	7	14	9,8	10	12
	Савельев Алексей Олегович	1	1	1	1	1
Анализ предметной области	Щукин Василий Алексеевич	3	7	4,6	5	6
	Савельев Алексей Олегович	1	1	1	1	1
Проектирование информационной системы	Щукин Василий Алексеевич	7	9	7,8	8	10
	Савельев Алексей Олегович	1	1	1	1	1

Наименование работы	Исполнители работы	Трудоемкость работ, чел-дни			Длительность работ, дни	
		Tmin	Tmax	Toж	Tr	Tk
Реализация модулей приложения	Щукин Василий Алексеевич	28	35	30,8	31	36
Тестирование приложения	Щукин Василий Алексеевич	5	7	5,8	6	7
Написание главы по проектированию приложения	Щукин Василий Алексеевич	3	7	4,6	5	6
Написание главы по реализации серверной части приложения	Щукин Василий Алексеевич	3	7	4,6	5	6
Написание главы по реализации клиентской части приложения	Щукин Василий Алексеевич	3	7	4,6	5	6
Выполнение других частей работы (финансовый менеджмент, социальная ответственность)	Щукин Василий Алексеевич	7	10	8,2	8	10
Подведение итогов, оформление работы	Щукин Василий Алексеевич	2	3	2,4	2	3

Данные из таблицы можно использовать для построения диаграммы Ганта, при этом следует учитывать, что некоторые этапы исследовательской работы могут выполняться параллельно. Построенная диаграмма Ганта представлена на рисунке 20.

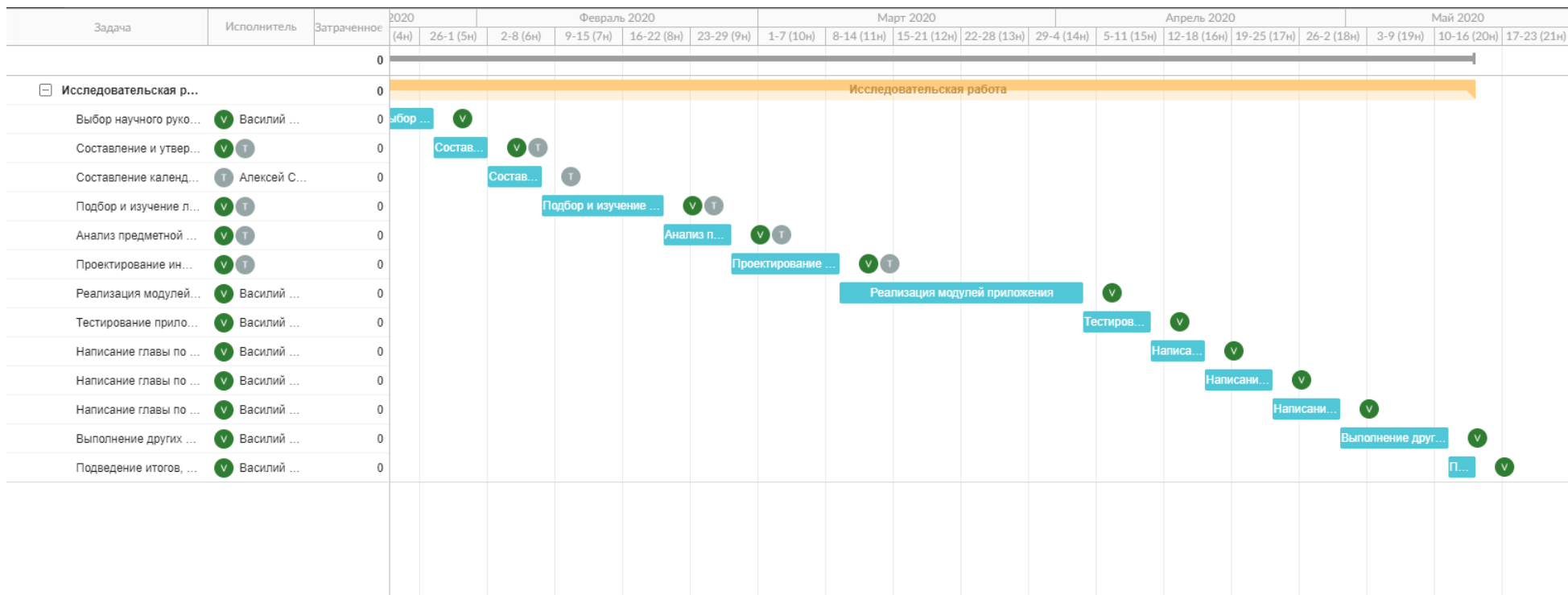


Рисунок 20. Диаграмма Ганта, охватывающая все задачи исследовательской работы.

Ориентировочные даты выполнения работы с 22.01.2020 по 13.05.2020.

5.3. Бюджет научно-технического исследования

5.3.1. Расчет материальных затрат научно-технического исследования

На момент начала исследовательской работы автор обладал всеми необходимыми аппаратно-техническим средствами.

Для реализации проекта использовался персональный компьютер повышенного класса мощности с периферийными устройствами на сумму 90200 рублей и ноутбук среднего класса мощности стоимостью 67000 рублей. Общая стоимость технического оборудования составила 157200 рублей. Срок полезного использования офисных машин (код 330.28.23.23) составляет от 2 до 3 лет, в расчетах его можно принять его за 3 года. Время, требуемое для написания ВКР – около 5 месяцев.

Норма амортизации вычисляется по следующей формуле:

$$A_n = \frac{100\%}{3} = 33,33 \%$$

Годовые амортизационные отчисления:

$$A_g = S * \frac{A_n}{100} \% = 157200 * 0.33 = 51876 \text{ рублей}$$

Ежемесячные амортизационные отчисления при этом составят:

$$A_m = \frac{A_g}{12} = 4323 \text{ рубля}$$

Итоговая сумма амортизации основных средств за время написания исследовательской работы составит: $A = 4323 * 5 = 21615$ рубля.

5.3.2. Основная заработная плата исполнителей исследовательской работы

Ниже представлена формула для расчета затрат на заработную плату

$$Z_n = Z_{\text{осн}} + Z_{\text{доп}}, \text{ где}$$

$Z_{\text{осн}}$ – выраженная в рублях основная заработная плата;

$Z_{\text{доп}}$ – выраженная в рублях дополнительная заработная плата.

Основную заработную плату можно вычислить по формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} * T_p * (1 + K_{\text{пр}} + K_{\text{д}}) * K_p, \text{ где}$$

$Z_{\text{дн}}$ – среднедневная заработная плата, руб.;

T_p – продолжительность работ, выполняемых работником (рабочие дни);

$K_{\text{пр}}$ – премиальный коэффициент (0,3);

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

K_p – районный коэффициент, для Томска составляет 1,3.

Среднедневная заработная плата вычисляется по формуле:

$$Z_{\text{дн}} = \frac{Z_m * M}{F_d}$$

Z_m – месячный оклад работника, руб. M – количество месяцев работы без отпуска в течение года (для 6-дневной рабочей недели $M=10,4$). F_d – действительный годовой фонд рабочего времени персонала, выражен в рабочих днях. Расчет баланса приведен в таблице 12.

Таблица 12 – Баланс рабочего времени (для 6-дневной недели)

Показатели рабочего времени	Дни
Календарные дни	366
Нерабочие дни (праздники/выходные)	118
Потери рабочего времени (отпуск/невыходы по болезни)	56
Действительный годовой фонд рабочего времени	243

Таким образом, в 2020 году действительный годовой фонд рабочего времени составляет 243 дня. Найденных показателей достаточно, чтобы составить таблицу расчета основной заработной платы, при этом зарплата студента в месяц равняется 21760, а руководителя – 33664 рублей.

Таблица 13 – Расчет основной заработной платы

Исполнители	З _{дн} , руб	К _{пр}	К _д	К _р	Т _р	З _{осн}
Щукин Василий Алексеевич	931,2	0	0	1	8	156903,74
Савельев Алексей Олегович	1440,76	0	0	1	1	41954,94
Итого						198858,68

5.3.3. Расчет дополнительной заработной платы

Дополнительная заработная плата может быть вычислена путем умножения основной заработной платы на соответствующий коэффициент. Для расчетов значение коэффициента было установлено в 0,13.

Таблица 14 – Расчет дополнительной заработной платы

Исполнитель	Основная заработная плата, руб.	Коэффициент дополнительной заработной платы	Дополнительная заработная плата, руб
Щукин В.А.	156903,74	0,13	20397,49
Савельев А.О.	41954,94	0,13	5454,15
Итого:			25851,64

5.3.4. Отчисления во внебюджетные фонды

Подсчитав основную и дополнительную заработную плату, можно рассчитать сумму отчислений во внебюджетные фонды через соответствующий коэффициент, что отражено в таблице 15.

Таблица 15 – Расчет отчислений во внебюджетные фонды

Исполнитель	Основная заработная плата + дополнительная, руб.	Коэффициент отчислений во внебюджетные фонды	Сумма отчислений
Трусов Е.Д.	177301,23	0,3	53190,37
Соколова В.В.	47409,09	0,3	14222,73
Итого:			67413,10

5.3.5. 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:

$$Z_{\text{накл}} = \sum_{i=1}^4 \text{Сумма } i \text{ – той статьи} * K_{\text{нр}}, \text{ где}$$

$K_{\text{нр}}$ – коэффициент, учитывающий накладные расходы (16 %)

Таким образом, накладные расходы составляют:

$$\begin{aligned} Z_{\text{накл}} &= 0,16 * (198858,68 + 25851,64 + 67413,10 + 21615) \\ &= 50198,15 \text{ руб.} \end{aligned}$$

5.3.6. Формирование бюджета затрат научно-исследовательского проекта

Зная затраты на амортизацию, выплату основной и дополнительной заработной платы, отчисления во внебюджетные фонды и накладные расходы, можно вычислить общий бюджет научно-исследовательского проекта, что показано в таблице 16.

Таблица 16 – Бюджет затрат научно-исследовательского проекта

Наименование статьи	Сумма (руб.)	Удельный вес (%)
Амортизационные затраты на спецоборудование	21615,00	5,9
Затраты на основную заработную плату	198858,68	54,6
Затраты на дополнительную заработную плату	25851,64	7,1
Затраты на отчисление во внебюджетные фонды	67413,10	18,6
Накладные расходы	50198,15	13,8
Бюджет затрат НИИ	363936,57	100

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

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования [18]. Его нахождение связано с определением двух средневзвешенных величин:

финансовой эффективности и ресурсоэффективности.

Интегральный финансовый показатель разработки определяется как:

$$I_{\Phi}^p = \frac{\Phi_p}{\Phi_{max}}, (.1)$$

где: Φ_p – стоимость исполнения, руб.;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в т. ч. аналоги).

$$I_{\Phi}^p = \frac{1\ 161\ 622,29}{1\ 161\ 622,29} = 1$$

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное удешевление стоимости разработки в размах.

Интегральный показатель ресурсоэффективности исполнения объекта исследования можно определить следующим образом:

$$I_p = \sum_n a \times b, (.2)$$

где a – весовой коэффициент параметра;

b – бальная оценка параметра для аналога и разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Расчет интегрального показателя ресурсоэффективности приведен в таблице 17.

Таблица 17 – Расчет интегрального показателя ресурсоэффективности

Критерии	Объект исследования	Весовой коэффициент параметра	Оценка выполнения
1. Способствует росту производительности труда пользователя		0,25	4
2. Удобство в эксплуатации (соответствует требованиям потребителей)		0,2	5
3. Функциональные возможности		0,2	4
4. Надежность		0,2	4
5. Экономия времени		0,15	5
Итого		1	

$$I_p = 0,25 * 4 + 0,2 * 5 + 0,2 * 4 + 0,2 * 4 + 0,15 * 5 = 4,35$$

Интегральный показатель эффективности исполнения разработки ($I_{исп}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп} = \frac{I_p}{I_\phi} = \frac{4,35}{1} = 4.35. \quad (.3)$$

Результат работы можно считать положительным, так как оценка интегрального показателя ресурсоэффективности выше 4.

Таким образом, полученные при анализе конкурентных решений данные позволяют сделать вывод, что разработка является привлекательной для инвесторов. Продукт имеет много преимуществ перед рассмотренными конкурентами, в особенности по таким критериям, как удобство в эксплуатации, надежность и повышение производительности труда. *SWOT*-анализ позволил выявить слабые и сильные стороны, возможные перспективы и угрозы, а также предложены рекомендации по минимизации их влияния.

Также была построена структура работ проекта и определены ответственные должности для их выполнения. В соответствии с назначенными работами была рассчитана их трудоемкость и составлен план-график работ в виде диаграммы Ганта. Общая длительность проектирования и разработки программного продукта составила 247 дней. Общий бюджет НТИ составил

363936,57 рублей. Бюджет включает в себя затраты на основную и дополнительную заработную плату работников, материальные затраты, отчисления во внебюджетные фонды и накладные расходы.

5.5. Выводы по главе

В разделе исследовательской работы, посвященной оценке коммерческого потенциала и перспективности, было проведено подробное планирование проектных работ. Также проведен SWOT-анализ, составлены выводы на основе возможностей, угроз, сильных и слабых сторон проекта. С помощью анализа разных аспектов ресурсозатрат, Общая длительность проведения работ по проекту ориентировочно составляет 120 календарный дней. В заключении, рассчитан бюджет научно-технического исследования. Потенциальная стоимость разработки информационной системы составляет 363936,57 руб.

Данная разработка позволит повысить показатели монетизации потоковых онлайн трансляций, поддерживая положительный тренд как в финансовой модели платных подписок, так и в модели добровольных пожертвований.

Глава 6. Социальная ответственность

6.1. Введение

Исследовательская работа заключалась в проектировании и разработке клиент-серверного приложения чат-бота, автоматизирующего обработку текстовых запросов в чате онлайн-трансляций. Используя приложение, авторы эфиров на стриминговых платформах могут получить дополнительный источник контента для зрителей трансляции, а также давать ответ на часто задаваемые вопросы в виде команд текстового чата. Более того, интеграция программной системы с чатом трансляции дает возможность использования сторонних ресурсов (новостей, статистики и т.п.), таким образом повышается разнообразие контента на трансляции, при этом сам контент может любым, единственное ограничение – возможность изложения информации в виде текста.

Согласно требованиям, предъявленным к разрабатываемой программной системе, клиентами могут являться пользователи, использующие как стационарные персональные компьютеры и ноутбуки, так и мобильные устройства. По этой причине интерфейс клиентской части системы необходимо было адаптировать под различные пользовательские устройства. Сама разработка при этом велась с использованием ноутбука и персонального компьютера, в связи с чем автор проекта подвергался таким вредным факторам, как:

1. Некачественная TN-матрица монитора компьютера под острым углом сильно искажает цвета, вплоть до полного инвертирования при угле, близком к 180 градусов. Неправильная цветопередача может привести к легкой дезориентации и головным болям;

2. Частота вертикальной развертки мониторов ограничена 60 Гц, мерцание, возникающее при этом из-за особенностей работы ШИМ-контроллера, быстро утомляет глаза;
3. В непосредственной близости от электронно-вычислительных устройств существует область повышенного электромагнитного излучения, длительное нахождение в которой вредно для внутренних органов;
4. Специфичность трудовой деятельности, связанной с разработкой программных систем, состоит в необходимости выполнения работ преимущественно в сидячем положении, что может привести к нарушению кровообращения и образованию зловредных образований [19];
5. В вечернее и ночное время разработчик вынужден использовать искусственное освещение.

6.2. Правовые и организационные вопросы обеспечения безопасности

6.2.1. Организационные мероприятия обеспечения безопасности

Разработка веб-приложения велась в учебной аудитории Томского Политехнического Университета по адресу г. Томск, ул. Советская 84/3, 204 общей площадью 26 квадратных метров. В качестве искусственного источника освещения использовались 3 лампы накаливания, общей мощностью 85 Вт. Помещение оборудовано восемью компьютерными столами с выдвижной подставкой для клавиатуры, являющимися персональным рабочим местом. Автор научно-исследовательской работы взаимодействовал с электронно-вычислительными устройствами, находясь в операторском кресле, выполненном в виде компьютерного кресла с регулируемыми подлокотниками и углом наклона спинки. Перемещение кресла внутри

помещения обеспечивают 5 пластиковых колес диаметром 50 мм. Доступ к свободному креслу в помещении был постоянно.

С целью минимизации воздействия вредных факторов на разработчика при выполнении научно-исследовательской работы «Проектирование и реализация чат-бота стримингового сервиса для повышения вовлеченности аудитории», 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»):

1. 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.

2. При работе двумя руками органы управления размещают с таким расчетом, чтобы не было перекрещивания рук.
3. Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.
4. В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.
5. Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.
6. 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.
7. 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.

Автором научно-исследовательской работы «Проектирование и реализация чат-бота стримингового сервиса для повышения вовлеченности аудитории» были соблюдены в допустимой мере все требования, предусматриваемые государственным стандартом 12.2.032-78.

6.3 Производственная безопасность

В данном пункте производится анализ вредных и опасных факторов, которые могут возникнуть на одном из этапов выполнения работы.

Рассмотрены следующие факторы: отклонение показателей микроклимата и естественного, недостаточная освещенность и повышенный уровень электромагнитного излучения. Таблица 18 содержит информацию о воздействии факторов в зависимости от этапов научного исследования.

Таблица 18 – Возможные опасные и вредные факторы

Факторы (по ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Проектирование	Разработка	Формирование отчетности	
Отклонение показателей микроклимата	+	+	+	СанПиН 2.2.2.548-96
Отсутствие или недостаток естественного света	+	+	+	СП 52.13330.2016.
Недостаточная освещенность рабочей зоны	+	+	+	СанПиН 2.2.1/2.1.1.1278-03
Повышенный уровень электромагнитных излучений	+	+	+	СанПиН 2.1.8/2.2.4.1383-03

Как видно из таблицы 18, опасные и вредные факторы воздействовали на разработчика на всем протяжении работ, так как все этапы научно-исследовательской работы были произведены в одном и том же помещении.

6.3.1 Анализ опасных и вредных производственных факторов

Таблица 19 – 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	Возможное 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 20.

Таблица 20 – 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
	21,0-22,9	20,0-29,0		

6.3.2 Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов

6.3.2.1 Мероприятия по снижению воздействия недопустимого микроклимата

Для восстановления и поддержания допустимого микроклимата при разработке проекта «Проектирование и реализация чат-бота стримингового сервиса для повышения вовлеченности аудитории» необходимо придерживаться следующих правил [20]:

- 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.
- Рациональное размещение рабочих мест.
- Ежедневная влажная уборка рабочего помещения.

6.3.2.2. Мероприятия по снижению воздействия недостатка освещенности рабочего места и естественного света

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.

6.3.2.3. Мероприятия по снижению воздействия недопустимого уровня электромагнитного излучения

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.

6.4 Экологическая безопасность

Экологической безопасностью называется комплекс мероприятий по снижению негативных влияний производственной деятельности человека на окружающую среду и защиту человека от последствий этого влияния.

Для выполнения научно-исследовательской работы «Проектирование и реализация чат-бота стримингового сервиса для повышения вовлеченности аудитории» 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. Рост потребления электроэнергии может привести к строительству дополнительных атомных электростанций, что повышает вероятность ядерной катастрофы [21].

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.

6.5 Безопасность в чрезвычайных ситуациях

Чрезвычайная ситуация (ЧС) – обстановка на определенной территории, сложившаяся в результате аварии, опасного природного явления, катастрофы, стихийного или иного бедствия, которая может повлечь за собой человеческие жертвы, а также ущерб здоровью человека или окружающей среде, значительные материальные потери и нарушение условий жизнедеятельности [22].

6.5.1 Пожар

Научно-исследовательская работа «Проектирование и реализация чат-бота стримингового сервиса для повышения вовлеченности аудитории» проходила в помещении, подходящем под определение офис. Одной из наиболее возможной чрезвычайной ситуацией, которая может возникнуть при работе в помещениях такого типа – пожар. К пожару могут привести неисправности в технических средствах, оргтехнике, а также действия самих сотрудников [23]. 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:

1. 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».

2. Сообщения о пожаре, как правило, передаются по телефону. Следует помнить, что с помощью сотового телефона можно вызвать помощь даже при отсутствии денег на счете или SIM-карты по номеру «112».

6.5.2 Землетрясение

Под землетрясением понимают подземные толчки и колебания земной поверхности. Землетрясения отражают процесс геологического преобразования планеты, первопричиной землетрясений являются глобальные геологические и тектонические силы

Томская область располагается на значительном отдалении от зон сейсмической активности, однако, за последние 10 лет в пределах городской зоны произошло несколько землетрясений, максимальная зафиксированная амплитуда толчков составила 5,9. В связи с тем, что научно-исследовательская работа происходила в помещении на 6 этаже, что создает дополнительные факторы риска при земных толчках, участникам работ следует знать инструкции по поведению во время землетрясения [24]:

1. При возможности захватить с собой документы, деньги, предметы первой необходимости, фонарик.
2. Остерегаться падающих предметов, оборванных проводов и других источников опасности.
3. Сохранять спокойствие и не допускать паники.
4. При нахождении на верхних этажах многоэтажного здания — оставаться в здании, предварительно открыть входную дверь, которая в дальнейшем может оказаться перекошенной и заклиненной.

5. Быстро занять наиболее безопасное место в помещении: в дверных проемах капитальных стен, у ближайшей к центру здания капитальной стены, опорной колонны, в углу комнаты, всегда подальше от окон, тяжелых предметов и мебели, которые могут опрокинуться.
6. В случае разрушения здания, сопровождающегося падением отдельных элементов перекрытия или частей капитальных стен, необходимо немедленно покинуть здание.
7. Покидая здание, не выпрыгивать из окон, расположенных выше первого этажа, стекла выбивать подручными средствами (стулом, табуреткой), в крайнем случае, рукой, обмотанной тряпкой.

6.6 Выводы по главе

В результате изучения и анализа стандартов и правил, касающихся работы в помещениях с электронно-вычислительными устройствами, можно сделать вывод, что выполнение проекта «Проектирование и реализация чат-бота стримингового сервиса для повышения вовлеченности аудитории» соответствовало всем заявленным нормам безопасности жизнедеятельности. Автор проекта не подвергался серьезному воздействию опасных факторов.

Рабочее место и помещение в целом во время проведения исследовательской работы соответствовало региональным стандартам, а также санитарно-эпидемиологическим правилам и нормам.

Заключение

В результате работы спроектирована и разработана программная система чат бот. В процессе аналитической работы над архитектурой приложения, были построены модели системы в нотациях IDEF, BPMN, EPC, DFD, позволяющие точно описать процессы, потоки данных и события внутри системы и между объектами, связанными с ней.

Детальное моделирование позволило декомпозировать компоненты системы и задачи по их разработке, а также избежать ошибок и неверных решений на раннем этапе работы над приложением. Алгоритм обработки команд, выполненный в рамках паттерна «Цепочка ответственностей», а также тщательный выбор технологических инструментов позволяет обеспечить высокий потенциал к масштабированию системы.

Разработанная система позволяет создать единый прогресс зрителя в рамках канала, система мотивации и повышения статуса позволит сделать пользователя более лояльным. Видимые цели, сформулированные в игровой форме, повысят среднее время просмотра эфиров, что является ключевым в механизме монетизации большинства стриминговых сервисов. Модель свободного распространения системы позволит создать пользовательскую базу, что на поздних этапах разработки станет основанием для внедрения монетизации в саму информационную систему.

Продолжение работы над программной системой, а именно проведение статистических метрик на реальных пользователях, их анализ и систематизация планируется в магистратуре.

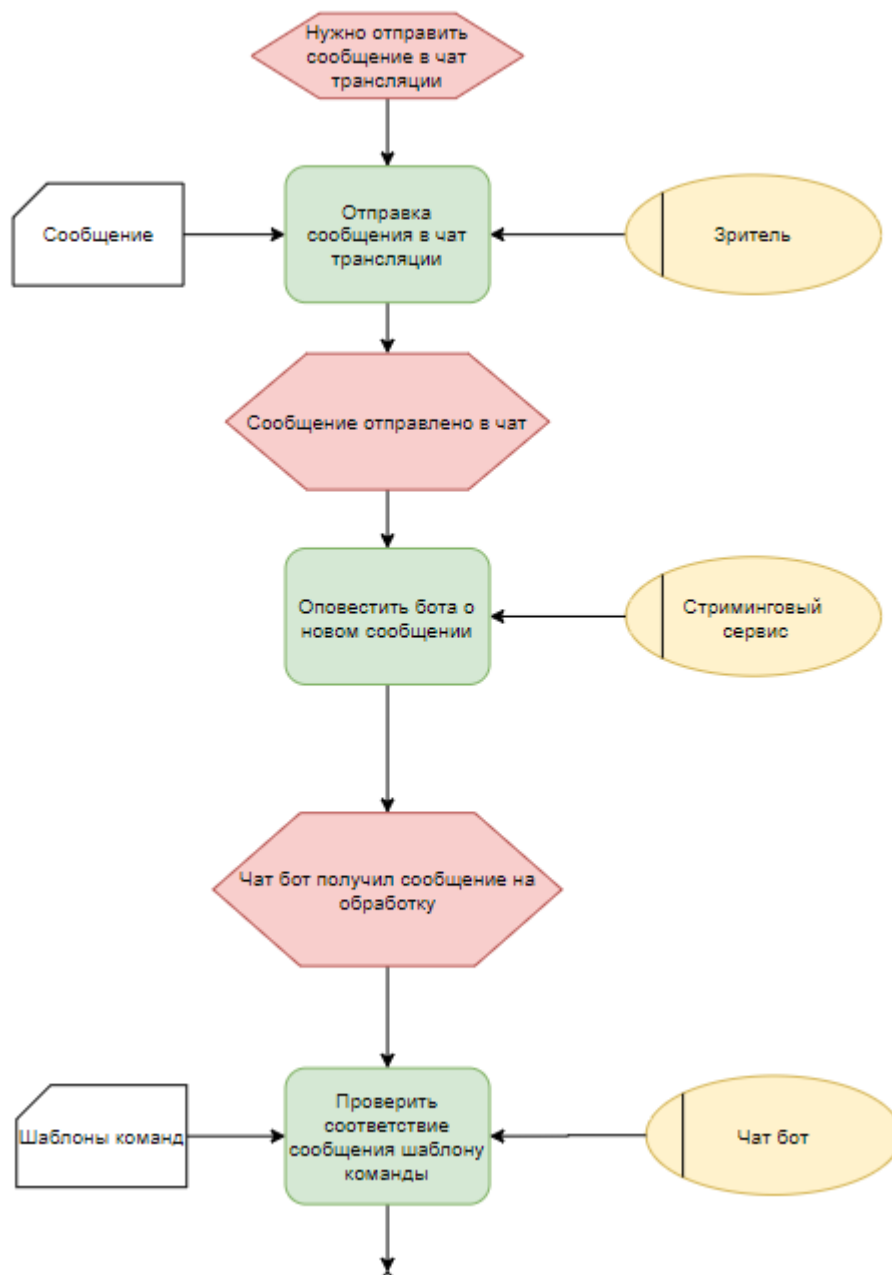
Список использованных источников

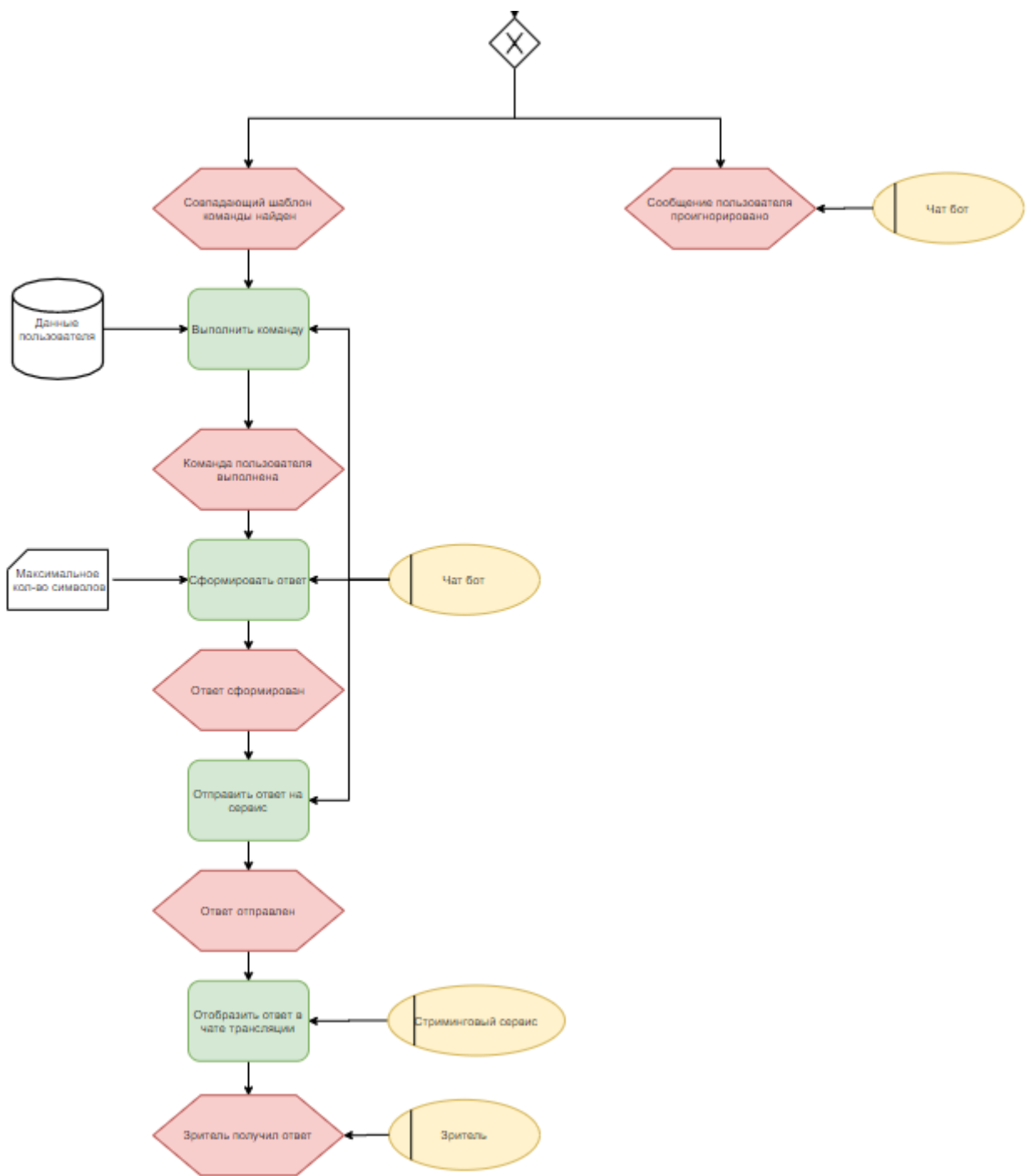
1. Video Streaming Market Report [Электронный ресурс]. – Режим доступа: <https://www.grandviewresearch.com/industry-analysis/video-streaming-market> (дата обращения: 02.03.2020).
2. Рынок ООТ-видео [Электронный ресурс]. – Режим доступа: <https://www.pwc.ru/ru/publications/mediaindustriya-v-2019/rynok-ott-video.html> (дата обращения: 10.03.2020).
3. 25+ Incredible Twitch Statistics to Know in 2020 [Электронный ресурс]. – Режим доступа: <https://leftronic.com/twitch-statistics/> (дата обращения: 12.03.2020).
4. Node.js Frameworks Which Will Rule In 2020 [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/486886/> (дата обращения: 24.04.2020).
5. Performance Comparison of Three Modern DBMS Architectures [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/3187532_Performance_Comparison_of_Three_Modern_DBMS_Architectures (дата обращения: 25.04.2020).
6. Express, Koa, Meteor, Sails.js: Four Frameworks Of The Apocalypse [Электронный ресурс]. – Режим доступа: <https://www.toptal.com/nodejs/nodejs-frameworks-comparison> (дата обращения: 28.04.2020).
7. Overview of modern database systems [Электронный ресурс]. – Режим доступа: <https://www.xplenty.com/blog/overview-of-modern-database-systems-which-database-is-right-for-your-use-case/> (дата обращения: 28.04.2020).

8. Как работает виртуальный DOM [Электронный ресурс]. – Режим доступа: <https://learn-reactjs.ru/faq/virtual-dom-and-internals> (дата обращения: 01.05.2020).
9. React A JavaScript library for building user interfaces [Электронный ресурс]. – Режим доступа: <https://reactjs.org/> (дата обращения: 02.05.2020).
10. Heroku: Cloud Application Platform [Электронный ресурс]. – Режим доступа: <https://www.heroku.com/> (дата обращения: 05.05.2020).
11. TypeScript – JavaScript that scales [Электронный ресурс]. – Режим доступа: <https://www.typescriptlang.org/> (дата обращения: 20.04.2020).
12. MobX: Documentation [Электронный ресурс]. – Режим доступа: <https://mobx.js.org/> (дата обращения: 20.04.2020).
13. Redux и Thunk вместе с Redux-thunk, пособие для начинающих [Электронный ресурс]. – Режим доступа: <https://tuhub.ru/posts/redux-i-thunk-vmeste-react-rukovodstvo-dlya-chajnikov> (дата обращения: 29.04.2020).
14. Препроцессор SASS/SCSS: документация на русском для начинающих [Электронный ресурс]. – Режим доступа: <https://zaurmag.ru/html5-css3/preprocessor-sass.html> (дата обращения: 04.05.2020).
15. Чистые функции – Введение в язык программирования Haskell [Электронный ресурс]. – Режим доступа: https://ru.hexlet.io/courses/introduction_to_programming/lessons/pure/theory_unit (дата обращения: 25.05.2020).
16. JSReport – платформа с открытым исходным кодом, использующая движки шаблонов [Электронный ресурс]. – Режим доступа: <https://jsreport.net/> (дата обращения: 09.06.2019).
17. Применение метода SWOT-анализа в стратегическом управлении [Электронный ресурс]. – Режим доступа: <http://powerbranding.ru/biznes-analiz/swot/> (дата обращения: 29.05.2020).

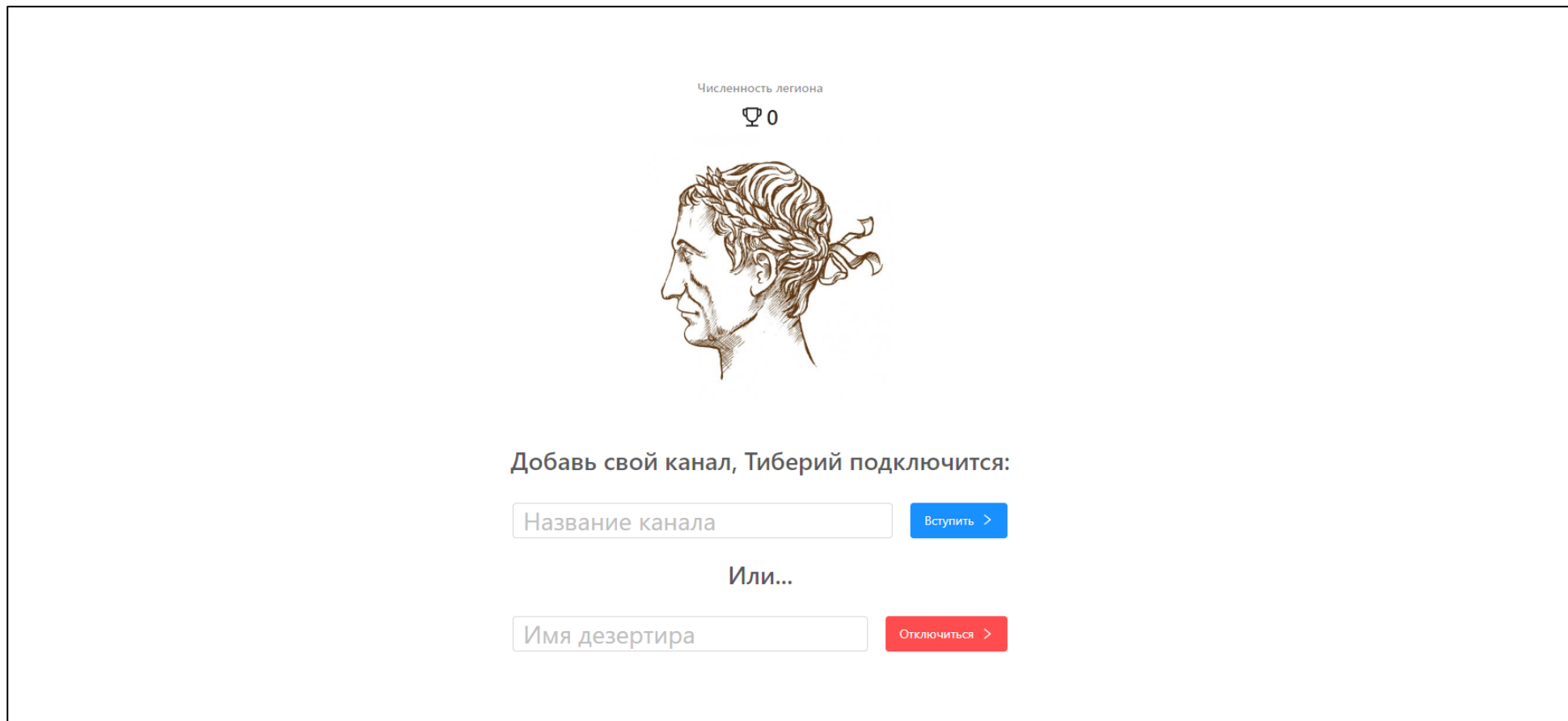
18. Интегральный финансовый анализ [Электронный ресурс]. – Режим доступа:
http://afdanalyse.ru/publ/finansovyj_analiz/integralnyj_finansovyj_analiz/9-1-0-81 (дата обращения: 01.06.2019).
19. ГОСТ 12.2.0.32-78. Система стандартов безопасности труда. Рабочее место при выполнении работ сидя [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/31970/> (дата обращения: 15.05.2020).
20. Оптимальные и допустимые параметры воздуха [Электронный ресурс]. – Режим доступа:
<http://www.technoholod.ru/site.aspx?SECTIONID=1940097> (дата обращения: 16.05.2020).
21. Состояние окружающей среды. Экологическая безопасность [Электронный ресурс]. – Режим доступа:
<http://www.infoeco.ru/index.php?id=58> (дата обращения: 15.05.2020).
22. МЧС России. Чрезвычайные ситуации [Электронный ресурс]. – Режим доступа: <https://www.mchs.gov.ru/dop/terms/item/86578/> (дата обращения: 15.05.2020).
23. Должностные инструкции при пожаре [Электронный ресурс]. – Режим доступа: <https://hr-portal.ru/doki/dolzhnostnaya-instrukciya-specialista-po-rozharnoy-bezopasnosti> (дата обращения: 20.05.2020).
24. Должностные инструкции при землетрясении [Электронный ресурс]. – Режим доступа: <http://my.sfu-kras.ru/safety/earthquake> (дата обращения: 20.05.2020).

Приложение А. Диаграмма событий бизнес-процесса





Приложение Б. Клиентский интерфейс программной системы



Приложение В. Рабочая среда облачного сервиса Heroku

