



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Уральский федеральный университет имени
первого Президента России Б. Н. Ельцина» (УрФУ)
Институт радиоэлектроники и информационных технологий – РТИ

ОТЧЕТ
о проектной работе
по теме: Игра-симулятор ИТ-компании
по дисциплине: Проектный практикум 1А

Команда: XYT STUDIOS

Екатеринбург

2022

Содержание

ВВЕДЕНИЕ	3
1. Команда.....	5
2. Целевая аудитория.....	6
3. Календарный план проекта.....	8
4. Определение проблемы.....	10
5. Подходы к решению проблемы.....	13
6. Анализ аналогов.....	15
7. Требования к продукту и к MVP.....	16
8. стек для разработки.....	18
9. Прототипирование.....	20
10. Разработка системы.....	22
11. Заключение.....	23
ЗАКЛЮЧЕНИЕ	23
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	26
ПРИЛОЖЕНИЕ А	29
ПРИЛОЖЕНИЕ Б	30

ВВЕДЕНИЕ

Во введении приводится обоснование актуальности работы, определяются ее цель и задачи.

Введение должно начинаться с обоснования актуальности выбранной темы. Освещение актуальности не должно быть многословным (примерно 2-3 абзаца текста). Требуется показать суть проблемной ситуации, степень ее разработанности; в исследовательских работах следует указать границу между знанием и незнанием о предмете исследования, противоречия, необходимость разрешения которых обосновывает актуальность работы. Цель обоснования актуальности – доказать, что проблема, которая является темой проекта, еще недостаточно разработана и не освещена.

Далее формулируется цель и задачи проекта, которые предстоит решить для реализации поставленной цели. Формулировка задач производится в форме перечисления (изучить..., описать..., установить..., выявить..., разработать..., составить..., и т.п.). Формулировку задач следует производить возможно более тщательно, поскольку описание их решения должно составлять основное содержание работы и ими определяются заголовки разделов работы. Количество задач должно быть 4-5; на основании формулировок задач в конце работы пишется заключение.

Объем введения составляет 2-3 страницы компьютерного текста.

КОМАНДА

- Ерофеев Александр Владиславович РИ-110915 – тимлид
- Голубева Кристина Владимировна РИ-110949 – дизайнер
- Носкова Дарья Николаевна РИ-110914 – аналитик
- Голосов Дмитрий Сергеевич РИ-110915 – программист

1. ЦЕЛЕВАЯ АУДИТОРИЯ

Для определения целевой аудитории используется методика 5W Марка Шеррингтона. Это наиболее распространенный способ определения целевой аудитории и психологических характеристик, которыми обладают потенциальные потребители.

Сегментация рынка проводится по 5 вопросам:

- а) Что? (What?) – сегментация по типу товара: что вы предлагаете потребительской группе? какие товары/услуги?
- б) Кто? (Who?) – сегментация по типу потребителя: кто приобретает товар/услугу? какой пол, геоположение, возраст?
- в) Почему? (Why?) – сегментация по типу мотивации к совершению покупки и потребления: какова потребность или мотивация клиента? какую проблему решает товар/услуга?
- г) Когда? (When?) – сегментация по ситуации в которой приобретается продукт: когда потребители хотят приобрести товар/услугу?
- д) Где? (Where?) – сегментация по месту покупок: в каком месте происходит принятие решения о покупке и сама покупка? – имеются ввиду точки контакта с клиентом, где можно повлиять на решение.

Анализ целесообразно представить в виде графика или диаграммы (рисунок 1).

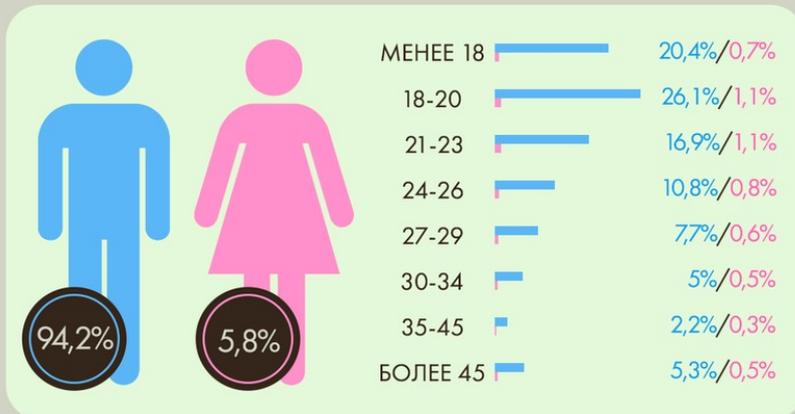


АНАЛИЗ ЦЕЛЕВОЙ АУДИТОРИИ

По данным официального сообщества ВКонтакте

96 199

FIFA19



ЦЕЛЕВЫЕ СООБЩЕСТВА

	vk.com/igm	34 126
	vk.com/mudakoff	30 490
	vk.com/virtuspro	28 052
	vk.com/dota2	27 225
	vk.com/4ch	25 356

Рисунок 1 – Анализ целевой аудитории

2. КАЛЕНДАРНЫЙ ПЛАН ПРОЕКТА

Название проекта:

Руководитель проекта:

Таблица 1 – Календарный план

№	Название	Ответственный	Длительность	Дата начала	Временные рамки проекта			
					1 нед	2 нед	3 нед	4 нед
Анализ								
1.1	Определение проблемы							
1.2	Выявление целевой аудитории							
1.3	Конкретизация проблемы							
1.4	Подходы к решению проблемы							
1.5	Анализ аналогов							
1.6	Определение платформы и стека для продукта							
1.7	Формулирование требований к MVP продукта							
1.8	Определение платформы и стека для MVP							
1.9	Формулировка цели							
1.10	Формулирование							

	<i>ие требований к продукту</i>							
1.11	<i>Определение задач</i>							
	...							
Проектирование								
2.1	<i>Архитектура системы (компоненты, модули системы)</i>							
2.2	<i>Разработка сценариев использования системы</i>							
2.3	<i>Прототипы интерфейсов</i>							
2.4	<i>Дизайн-макеты</i>							
2.5	<i>Архитектура системы (компоненты, модули системы)</i>							
	...							
Разработка								
3.1	<i>Написание кода</i>							
3.2	<i>Тестирование приложения</i>							
	...							
Внедрение								
4.1	<i>Оформление MVP</i>							
4.2	<i>Внедрение MVP</i>							
4.3	<i>Написание отчета</i>							

4.4	<i>Оформление презентации</i>							
	...							
	<i>Защита проекта</i>			07.06 - 15.06				

3. ОПРЕДЕЛЕНИЕ ПРОБЛЕМЫ

Универсальным способом определения потребностей и более целевой аудитории являются маркетинговые исследования. Но их проведение занимает много времени, усилий и требуют значительных вложений. Существует ряд эффективных способов выявления основных проблем клиента:

- опрос, проведение интервью;
- изучение конкурентов;
- тематические форумы;
- социальные сети;
- отзывы;
- анализ поисковых запросов.

С реальными клиентами проще выйти на контакт, чем с потенциальными потребителями. Лучше всего не отсылать анкету по почте, не получится отследить эмоции респондентов. Тем более многие не умеют выражать эмоции в письменном виде. В ходе разговора можно узнать больше о боли и проблемах своих клиентов.

Важную информацию можно найти на страницах конкурентов: отзывы, опыт использования товара, комментарии их целевой аудитории. Они содержат боль реальных клиентов. Много можно узнать и через жалобную книгу.

Следующий способ – это поиск информации на тематических форумах. На них можно найти вопросы, которые задают потенциальные или реальные клиенты перед покупкой или на что они жалуются. Боли клиентов можно обнаружить в социальных сетях. Это самый надежный и быстрый способ.

Необходимо найти профили и группы компаний, которые предлагают аналогичные товары, и почитать комментарии и отзывы их клиентов и подписчиков. Можно мониторить отзывы клиентов на специальных сайтах.

Это метод подходит для товаров, которые давно продаются на рынке и имеют широкое распространение. Именно про такие товары и можно найти достаточно много отзывов. Изучение отзывов для определения болей клиентов актуально для сферы услуг.

И последний способ – это анализ поисковых запросов пользователей. Это изучение тех запросов, по которым покупатели заходят на веб-ресурс компании. Кроме этого следует посмотреть, что ищут люди через поисковые машины Яндекс и Google. Для этого есть специальные сервисы Wordstat и Google Trends.

Условно проблему клиента можно отнести к определенной категории:

- когда у человека отсутствуют ресурсы на решение проблемы: время, деньги и умения;
- когда были попытки решить проблему, но они закончились неудачей;
- когда человек недоволен своим физическим и эмоциональным состоянием.
- Если отнести проблему своих клиентов к одной из данных групп, то картина будет ясна, что требуется целевой аудитории и проще составить предложение.

Выделяют также три уровня «боли»:

- технический;
- влияние на финансы;
- личный интерес.

Первый уровень предполагает наличие неудобств, которые вызваны конкретными бытовыми проблемами (поломка товара). В этом случае следует задать вопрос: Что создало эту проблему?

В рамках второго уровня методы решения проблем себя не оправдывают. Это отсутствие спроса на продукцию, уменьшение прибыли, неэффективность продаж, потеря выгоды. Вопрос: Как это вредит?

Третий уровень – это наличие личной мотивации клиента на решение проблемы. Он может получить существенную экономию времени или премию для осуществления своей мечты. Вопрос: Зачем нужно устранить проблему?

4. ПОДХОДЫ К РЕШЕНИЮ ПРОБЛЕМЫ

Старайтесь предлагать как можно больше идей альтернативных решений – лучше откинуть неподходящее решение на этапе анализа, чем упустить выгоду из-за малого количества рассмотренных вариантов. Для этого предлагается проводить «мозговые штурмы» с проектной командой, интервьюировать заказчиков и пользователей и использовать методики ТРИЗ (Теория Решения Изобретательских Задач). Тщательное изучение проблемы, критериев и целей является эффективным началом в выявлении альтернативных решений.

Подготовка нескольких альтернатив на ранних стадиях процесса позволяет повысить вероятность принятия адекватного решения с четким пониманием всех возможных последствий и рисков.

Важно отметить, что зачастую для выбора окончательного решения может потребоваться несколько итераций. На каждой итерации мы получаем все больше информации об анализируемой проблеме и лучше понимаем факторы, влияющие на принятие решения. При этом можно уточнять и изменять аналитические параметры (например, критерии и их веса).

Результат анализа покажет, насколько каждый из выбранных критериев реализован в каждой из рассмотренных альтернатив.

Очевидно, что при использовании структурированного метода анализа предпочтение отдается тому решению, которое наберет наибольшее количество очков по шкале соответствия выбранным критериям. Однако возможны ситуации, при которых результаты «взвешивания» нескольких решений могут оказаться очень близкими. В этом случае стоит провести дополнительный анализ, например, вовлекая экспертов и представителей заинтересованных сторон, выявляя и добавляя новые критерии выбора.

Например

Чтобы сократить процесс согласования счетов, мы можем:

- а) Перераспределить задачи между согласующими. В результате несколько человек могут быть исключены из процесса. Суммарное время процесса сократится за счет периодов передачи данных/ожидания/коммуникации при передаче.
- б) Перейти на электронный документооборот — достоверность счетов и данных в них будет подтверждена оператором ЭДО, подтверждение человеком не потребуется.
- в) Автоматически распознавать сканы счетов и сравнивать данные с цифрами из системы закупок. Ручная проверка и согласование не потребуются.

Чтобы продумать все варианты, надо разобраться – а что же происходит сейчас? Как устроен процесс без вашей системы, как работают пользователи и заказчики? Даже если процесса еще нет, подробная информация про текущее состояние очень важна. Так мы поймем, какое решение устранил проблему, а не создаст еще одну.

5. АНАЛИЗ АНАЛОГОВ

Прямые конкуренты – это аналогичное программное обеспечение на аналогичном рынке и работающий с целевой аудиторией, что и разрабатываемый продукт. Косвенные конкуренты – это похожее программное обеспечение с другими характеристиками или абсолютно другой продукт, но работающие с целевой аудиторией, что и разрабатываемый продукт.

Как правильно провести анализ конкурентов?

Рассмотрим шесть базовых шагов, которые надо выполнить при проведении подобной аналитики.

- определить цели анализа конкурентов;
- определить конкурентов;
- определить критерии анализа конкурентов;
- приступить к анализу (заполнение таблиц);
- сопоставить полученные данные;
- сделать выводы.

Используйте аналогичный продукты для того, чтобы заимствовать наиболее удачные приемы и способы решения проблем.

В таблице 2 представлен пример анализа конкурентов по трем критериям.

Таблица 2 – Анализ конкурентов

Критерии сравнения	Разделим чек	Splitwise	Делить чек	Скинемся!
Многопользовательский режим	нет	да	да	да
Возможность сканировать фото	нет	нет	да	нет
Отдельное приложение	да	да	да	нет

6. ТРЕБОВАНИЯ К ПРОДУКТУ И К MVP

Требования систематизируются несколькими способами.

Требования клиентов

Клиенты – это те, кто выполняет основные функции системного проектирования, со специальным акцентом на пользователе системы как ключевом клиенте. Пользовательские требования определяют главную цель системы.

Функциональные требования

Функциональные требования объясняют, что должно быть сделано. Они идентифицируют задачи или действия, которые должны быть выполнены. Функциональные требования определяют действия, которые система должна быть способной выполнить, связь входа/выхода в поведении системы.

Нефункциональные требования

Нефункциональные требования – требования, определяющие свойства, которые система должна демонстрировать, или ограничения, которые она должна соблюдать, не относящиеся к поведению системы. Например, производительность, удобство сопровождения, расширяемость, надежность, факторы эксплуатации.

Производные требования

Требования, которые подразумеваются или преобразованы из высокоуровневого требования. Например, требование для большего радиуса действия или высокой скорости может привести к требованию низкого веса.

Например

Анкета должна содержать файл с фото, так как фото необходимо при оформлении документов – это бизнес-требование. А возможно, и бизнес-правило.

Из бизнес-требования следует, что у пользователя должна быть возможность прикрепить фото к анкете – это пользовательское требование. То есть требование, описывающее действия пользователя.

Получается, что система должна иметь функционал прикрепления фото к анкете – это уже функциональное требование, описывающее поведение системы. Или как должна работать система, чтобы выполнять исходное пользовательское требование.

Будем хранить все фото в формате base64 в отдельной таблице в БД – это нефункциональные требования.

Фото в очень хорошем качестве нам не нужно, а также мы не хотим покупать много памяти для сервера. Поэтому сделаем ограничение на размер загружаемого фото: не более 10 Мб.

На каждое бизнес-требование, как правило, приходится несколько пользовательских. Пользовательское требование декомпозируется на какое-то число функциональных. К каждому функциональному требованию нужно продумать нефункциональные требования и ограничения.

Также нефункциональные требования и ограничения могут напрямую вытекать как из пользовательских требований, так и из бизнес-требований и правил.

Таким образом получаются деревья из требований, в вершине каждого из которых – бизнес-требование.

7. СТЕК ДЛЯ РАЗРАБОТКИ

Все чаще можно наблюдать ситуацию, когда именно скорость запуска программного проекта является решающим фактором успеха. Долго и напряженно делать что-то ненужное хуже, чем быстро выпустить решение, полезное для клиентов и собрать обратную связь для следующего рывка

Быстрый запуск возможен только при использовании максимума готовых составных элементов, фреймворков и библиотек с адекватным сроком поддержки (с учетом ожидаемого времени жизни вашего продукта).

К сожалению, алгоритмические и архитектурные знания востребованы все реже. Чаще приветствуется опыт решения похожих задач и практика использования комплексов инструментов и сильных сторон облачных провайдеров.

Не нужно ограничивать себя и все делать на одной технологии и только одном, ЛУЧШЕМ, языке программирования – огнестрел в руках ребенка эффективнее изучения боевых искусств самураем всю сознательную и бессознательную жизнь. Перфекционизм и идеализация убивают программные системы. Выбирайте верное оружие для каждой задачи в проекте. Берите готовое и сосредоточьте оставшиеся усилия на нестандартном.

Выбранные технологии должны быть простыми, понятными и доступными большинству в команде. Посмотрите, на чем пишут подобный проект в основном и возьмите этот стек технологий себе. Не автоматизируйте хостинг хаскелем и не пишите веб-сайты на C++ . Если ваш проект «взлетит» и начнет развиваться, только тогда можно рассмотреть возможность переписывания его небольших частей на более сложных технологиях и языках программирования. Но обычно до этой стадии программные проекты либо не доходят, либо доходят через несколько лет.

Фреймворк позволяет на порядки ускорить запуск программного проекта. Обязательно проверьте наличие хорошей документации и уточните

срок поддержки фреймворка, чтобы не случилась беда с полным переписыванием через 2-3 года. Это очень частый кейс.

Технология и язык(и) программирования не являются определяющими факторами успеха. Сосредоточьтесь на выстраивании самых необходимых процессов и внедрении ключевых практик разработки и проектирования. Правильные процессы, как правило, гарантированно приводят к «правильному» результату. Связавшись с «сектантами чистой технологии» или «попав на науку», вы потратите кучу времени впустую и можете так ничего и не запустить. Помните, 2-3 месяца это максимум для запуска в бой очередной итерации программного проекта.

8. ПРОТОТИПИРОВАНИЕ

Процесс создания прототипа состоит из четырёх шагов:

- определение начальных требований;
- разработки первого варианта прототипа (в ПО, например, который содержит только пользовательский интерфейс системы);
- этап изучения прототипа заказчиком и конечным пользователем. Получение обратной связи о необходимых изменениях и дополнениях;
- переработка прототипа с учетом полученных замечаний и предложений.

Качества, которыми должен обладать эффективный прототип

Этап создания прототипа не должен быть затяжным.

Эффективные прототипы являются одноразовыми. Они предназначены для того чтобы донести идею до заинтересованного лица. После того как идея была донесена, прототип может быть отвергнут.

Эффективные прототипы являются сфокусированными, это означает что следует обращать внимание на сложные части при создании прототипов. Необходимо найти шаблонные взаимодействия, которые давно известны в теории юзабилити.

Необходимо обращать внимание на элементы взаимодействия, которые принесут пользу вашему продукту.

Даже самые опытные специалисты-разработчики допускают ошибки. Это правило особенно очевидно в коллективной работе. По мере выполнения проекта накапливаются мелкие ошибки, допущения, неверные решения. В результате получается плохой продукт при хорошей исходной идеи. Разработка прототипа – средство, позволяющее проанализировать идеи, прежде чем на них будут потрачены время и деньги. Это распространенный в инженерной практике метод. Основная цель, достигаемая при создании прототипа, – это экономия времени и ресурсов. Ценность прототипа

заключается в том, что он является внешней оболочкой-моделью отражающей существенные отношения разрабатываемого продукта. По сравнению с реальным продуктом прототипы просты и недороги в разработке. При минимальном вложении средств можно обнаружить ошибки создателей и юзабилити проблемы, и улучшить пользовательский интерфейс до того, как сделаны значительные инвестиции в окончательную разработку и технологии.

Создание эффективного прототипа интерфейса является чрезвычайно важной задачей. Прототип должен хорошо выглядеть, чтобы понравиться заказчику и не вызвать вопросов у субъектов тестирования, он должен быть максимально дешёвым, максимально полным и, что немаловажно, должен с лёгкостью обновляться.

Требования к прототипу изменяются со временем. Сначала наиболее актуальными его свойствами являются скорость создания и простота модификации. Эти свойства позволяют быстро разработать и проверить несколько версий интерфейса, при этом ещё и исправить значительную часть ошибок.

Затем на первый план выходят функциональность и эстетичность, простота же модификации уже не столь важна, поскольку с каждой новой исправленной ошибкой снижается вероятность того, что прототип придётся полностью переделывать при обнаружении новой ошибки.

Поэтому всегда правильно сделать прототип настолько похожим на результирующую систему, насколько позволяет самая поздняя его версия. Первый прототип стоит делать максимально примитивным. Только после того, как тестирование подтверждает его правильность, стоит сделать более детализированный прототип.

Прототипы интерфейсов возможно привести в приложении к отчету.

9. РАЗРАБОТКА СИСТЕМЫ

На этом этапе формируется общая структура программного комплекса. В соответствии с технологией нисходящего структурного программирования программный комплекс разбивается на небольшие части – программные модули (блоки). Для каждого программного модуля формулируются требования по реализуемым функциям и разрабатывается алгоритм, реализующий эти функции. Необходимо привести описание этих модулей, а также задач, которые они решают.

Необходимо привести алгоритм работы приложения, т.е. точного предписания и последовательности действий, заложенных в логике продукта.

Также на этом этапе можно привести схему взаимодействия программных модулей, т.е. схема потоков данных программы.

В этой части можно привести описание самого процесса разработки разнотипных сложных блоков программного обеспечения.

—

ЗАКЛЮЧЕНИЕ

Заключение должно содержать общие выводы, обобщенное изложение основных проблем. Могут быть указаны положительные и отрицательные стороны разработанного решения. Должны быть указаны перспективы дальнейшей разработки темы. Примерный объем заключения 2-3 страницы.

В заключительной части работы должно быть дано последовательное, логическое изложение полученных результатов в соответствие поставленными изначально задачами.

На данный момент сервис готовится к запуску и тестированию на реальных пользователях.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Веб-технологии для разработчиков Обзор протокола HTTP / Веб-технологии для разработчиков. – 2022. – URL: <https://developer.mozilla.org/ru/docs/Web/HTTP/Overview> (дата обращения: 30.04.2022).
2. Вигерс К. Разработка требований к программному обеспечению / К. Вигерс, Дж. Битти. – 3-е изд. – Москва: Русская редакция, 2014. – 736 с. – ISBN 978-5-9909805-3-2.
3. Евлантьев С. Карта эмпатии клиента / Станислав Евлантьев. – 2022. – URL: <https://in-scale.ru/blog/karta-empatii/> (дата обращения: 01.05.2022).
4. Ильяхов М. Пиши, сокращай: как создать сильный текст / М. Ильяхов, Л. Сарычева. – 3-е изд. – Москва: Альпина Паблишер, 2022. – 440 с. – ISBN 978-5-9614-6526-6.
5. Кинзябулатов Р. IDEF0. Знакомство с нотацией и пример использования / Рамиль Кинзябулатов. – 2022. – URL: <https://trinion.org/blog/idef0-znakomstvo-s-notaciey-i-primer-ispolzovaniya> (дата обращения: 30.04.2022).
6. Коберн А. Современные методы описания функциональных требований к системам / А. Коберн. – Москва: Издательство «Лори», 2012. – 264 с. – ISBN 978-5-85582-326-4.
7. Мидоус Ф. Введение в протоколы HTTP и HTTPS / Филлип Мидоус. – 2022. – URL: <https://docs.microsoft.com/ru-ru/azure/rtos/netx-duo/netx-duo-web-http/chapter1> (дата обращения: 05.05.2022).
8. Сырцев В. Использование диаграммы вариантов использования UML при проектировании программного обеспечения / Вадим Сырцев. – 2021. – URL: <https://habr.com/ru/post/566218/> (дата обращения: 30.04.2022).
9. Фирстов Л. О развитии навыка говорения / Леонид Фирстов. – 2020. – URL: <https://habr.com/ru/post/494424/> (дата обращения: 30.04.2022).

10. Фитцпатрик Р. Спроси маму: Как общаться с клиентами и подтвердить и подтвердить правоту своей бизнес-идеи, если все кругом врут? / Р. Фитцпатрик. – Москва : Альпина Паблшер, 2021. – 160 с. – ISBN 978-5-9614-3045-5.

11. Шейнблат В. Естественный подход к освоению языка / Владимир Шейнблат. – 2022. – URL: <https://englishsimple.ru/articles/natural-approach-to-language-learning/> (дата обращения: 25.04.2022).

12. Шторкин С. Средневзвешенная система голосования / Семен Шторкин – 2022. – URL: <https://habr.com/ru/post/63664/> (дата обращения: 30.04.2022).

13. Apple MacOS / Apple. – 2022. – URL: <https://www.apple.com/za/macOS/what-is/> (дата обращения: 30.04.2022).

14. Best Programmer Функциональные и нефункциональные требования: полное руководство / Best Programmer. – 2021. – URL: <https://bestprogrammer.ru/izuchenie/funktsionalnye-i-nefunktsionalnye-trebovaniya-polnoe-rukovodstvo> (дата обращения: 15.05.2022).

15. Google Chrome Браузер от Google / Google Chrome. – 2022. – URL: <https://www.google.ru/chrome/> (дата обращения: 30.04.2022).

16. HeyLady Женское сообщество для создания конференций и практики английского языка / HeyLady. – 2022. – URL: <https://www.heylady.io/> (дата обращения: 30.04.2022).

17. Lingbe Бесплатные звонки с носителями в любое время / Lingbe. – 2022. – URL: <https://lingbe.com/> (дата обращения: 30.04.2022).

18. PostgrePro Что такое PostgreSQL? / PostgrePro. – 2020. – URL: <https://postgrespro.ru/docs/postgresql/12/intro-what-is> (дата обращения: 30.04.2022).

19. Proglib Об Agile / Denver. – 2022. – URL: <https://proglib.io/p/klyuchevye-razlichiya-mezhdu-agile-scrum-i-kanban-2022-02-24> (дата обращения 05.04.2022).

20. SimilarWeb Аналитика трафика Tandem / SimilarWeb. – 2022. – URL: <https://www.similarweb.com/ru/website/tandem.net/#traffic>

(дата обращения: 30.04.2022).

21. Tandem / Tandem.net. – 2022. – URL: <https://www.tandem.net/ru>

(дата обращения: 30.04.2022).

22. JavaScript WebSocket / JavaScript. – 2022. – URL: <https://learn.javascript.ru/websocket> (дата обращения: 02.05.2022).

23. VC.RU Анализ конкурентов на практике / VC.RU. – 2020. – URL: <https://vc.ru/marketing/181297-analiz-konkurentov-na-praktike-10-shagov-shablon>

(дата обращения: 30.04.2022).

24. VC.RU Как составить портрет клиента (целевой аудитории) / VC.RU. – 2020. – URL: <https://vc.ru/marketing/156147-kak-sostavit-portret-klienta-celevoy-auditorii-instrukciya-s-primerami>

(дата обращения: 30.04.2022).

ПРИЛОЖЕНИЕ А

