

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е. Н. Скуратенко
подпись

« _____ » _____ 2020 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

Тема: «Разработка информационной системы сервиса по оказанию услуг
программного и аппаратного обеспечения»

Руководитель _____ зав. кафедрой, к.т.н. Е.Н. Скуратенко
подпись, дата

Выпускник _____ С.В. Кокшаров
подпись, дата

Консультанты
по разделам:

Экономический _____ Е. Н. Скуратенко
подпись, дата

Нормоконтролер _____ В. И. Кокова
подпись, дата

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е. Н. Скуратенко
подпись

« _____ » _____ 2020 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Кокшарову Сергею Валерьевичу

Группа ХБ 16-03

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: «Разработка информационной системы сервиса по оказанию услуг программного и аппаратного обеспечения»

Утверждена приказом по институту № 216 от 06.04.2020

Руководитель ВКР: Е. Н. Скуратенко, зав. кафедрой «Прикладная информатика, математика и естественно-научные дисциплины», кандидат технических наук, ХТИ – филиал СФУ

Исходные данные для ВКР: создать он-лайн сервис, позволяющий проводить коммуникацию с потребителями услуг программного и аппаратного обеспечения.

Перечень разделов ВКР:

1. Анализ предметной области информационной системы.
2. Практическая реализация сервиса по оказанию услуг программного и аппаратного обеспечения.
3. Оценка экономической эффективности внедрения информационной системы.

Перечень графического материала: нет

Руководитель ВКР _____
подпись

Е.Н. Скуратенко

Задание принял к исполнению _____
подпись

С.В. Кокшаров

«__» _____ 2020 г.

РЕФЕРАТ

Бакалаврская работа по теме «Разработка сервиса по оказанию услуг программного и аппаратного обеспечения» содержит 85 страниц текстового документа, 11 таблиц, 74 рисунка, 20 использованных источников, 16 формул.

ИНФОРМАЦИОННАЯ СИСТЕМА, САЙТ, ПРОЕКТИРОВАНИЕ, УСЛУГИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, СТОИМОСТЬ, РАСХОДЫ, АНАЛИЗ, РАЗРАБОТКА, IDEF, DFD.

Объект: процесс оказания услуг.

Цель выпускной квалификационной работы (ВКР): создать информационную систему, являющуюся местом коммуникации производителя и потребителей услуг программного и аппаратного обеспечения.

Задачи ВКР:

- проанализировать предметную область;
- разработать информационную систему для ХТИ – филиала СФУ;
- посчитать экономическую эффективность проекта.

ВКР выполняется в соответствии с индивидуальным заданием. В результате выполнения ВКР был разработан сервис по оказанию услуг программного и аппаратного обеспечения.

SUMMARY

The theme of the Bachelor's thesis is «Development of Software and Hardware to Field Services». It comprises 85 pages, 11 charts, 74 drawings, 20 reference items, 16 formulae.

IT SYSTEM, WEBSITE, PLANNING, SERVICES, SOFTWARE, COST, OPERATIONAL COSTS, ANALYSIS, DEVELOPMENT, IDEF, DFD.

The object of the research is the process of fielding services.

The purpose of the research is to design an IT system to support communication between the manufacturer and consumers of software and hardware services.

The objectives of the graduation thesis: to analyze the subject matter, to develop the IT system for KhTI – branch of SibFU, to evaluate the cost-effectiveness of the project implementation.

The graduation thesis is performed in accordance with an individual task. The software and hardware service provision has been developed.

English language supervisor

(signature, date)

N.V.Chezybaeva

СОДЕРЖАНИЕ

1. Анализ предметной области	8
1.1 Актуальность	8
1.2 Характеристика деятельности предприятия заказчика	10
1.3 Анализ рынка разрабатываемого продукта	13
1.4 Сравнение средств разработки ИС	18
1.5 Выводы по разделу «Анализ предметной области»	22
2. Проектирование ИС	23
2.1 Выбор жизненного цикла информационной системы.....	23
2.2 Модель ИС в нотации IDEF0	23
2.3 Развернутая диаграмма IDEF0.....	25
2.4 Модель ИС в нотации IDEF3	27
2.5 Модель ИС в нотации DFD.....	29
2.6 Разработка проекта	32
2.7 Выводы по разделу «Проектирование ИС»	66
3 Экономическое обоснование реализации проекта	66
3.1 Расчет затрат реализации проекта создания ИС	66
3.2 Капитальные затраты	67
3.3 Расчёт экономической эффективности проекта	76
3.4 Выводы по разделу «Экономическое обоснование реализации проекта»	81
ЗАКЛЮЧЕНИЕ	82
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	83

ВВЕДЕНИЕ

В современном мире всё большую популярность набирают информационные технологии. Любую информацию можно найти в любом месте при помощи интернета. Разработка проекта предоставляет людям возможность оказывать услуги для решения проблем с техническими устройствами.

В ВКР решаются следующие важные вопросы:

- анализ предметной области;
- определяется жизненный цикл для проекта;
- описывается ИС в нотациях IDEF0, IDEF3 и DFD;
- описывается практическая часть разработки проекта;
- подсчитывается финансовая выгода проекта.

На первом этапе работы рассмотрена актуальность работы, проанализированы предметная область и рынок программного обеспечения. Проведено сравнение средств разработки для создания информационной системы.

На втором этапе проведено моделирование бизнес-процессов: создания информационной системы и коммуникации, организованной с помощью сервиса. Используются нотации IDEF0, IDEF3 и DFD. На основе проведенного моделирования создана информационная система – сайт. Во второй части выпускной квалификационной работы описаны шаги его разработки.

На третьем этапе произведен расчёт затрат реализации проекта создания ИС. А именно амортизация, расчёт капитальных, эксплуатационных и прямых затрат. На основе этого оценивается экономическая эффективность проекта.

1. Анализ предметной области

1.1 Актуальность

Информация – это источник знаний, который необходим для существования и дальнейшего развития человеческого общества. В настоящее время существует интернет, который заполняет всё большие отрасли экономической жизни. Интернет-сайты позволяют не только информировать о чём-либо, но и предоставлять обслуживание в областях разного рода индустрий.

В России применение информационных технологий в повседневной жизни является приоритетным направлением политики государственной власти. Например, для экономии времени и денег можно использовать ФГИС "Единый портал государственных и муниципальных услуг (функций)" - портал Госуслуги. При помощи них удобно подавать электронные заявления, проверять уведомления, получать выписки, оплачивать госпошлины, налоги и штрафы.

В жизни обычных людей информационные технологии и интернет стали повседневным делом. Например, у людей, у которых жизнь тесно связана с переездами или поездками на дальние расстояния, часто меняется место пребывания. Для таких ситуаций существует такая услуга как бронирование отелей. Другим примером является покупка различных товаров по низким ценам и вне очереди. По этой причине существуют интернет магазины одежды, электроники, бытовой техники и т. д. Ещё одним важным элементом в жизни человека является перемещение, вследствие чего имеются сервисы для покупок онлайн билетов. Время – это невозполнимый ресурс и его нужно грамотно использовать. Все вышеперечисленные услуги позволяют этого достичь.

Возникает огромная потребность изучения новых информационных технологий разных слоёв населения и при всём многообразии интернет ресурсов, трудно найти единое пространство для получения данных знаний.

Основанием для разработки сервиса по оказанию услуг является заказ Хакасского технического института – филиала ФГАОУ ВО «Сибирский федеральный университет» для оказания помощи людям, которые имеют небольшую область знаний об информационных технологиях, а также для категории людей, которые не обладают достаточным временем или денежными средствами для решения своей проблемы. Иначе говоря, услуги будут предоставляться бесплатно, в основе которой будет лежать деятельность на добровольной основе. Кроме этого, студентам даётся возможность получить опыт работы с клиентами IT-сферы, именно обучающиеся института будут проявлять активность в работе сервиса.

Цель: создать программный продукт, при поддержке которого будут осуществляться IT-услуги.

Задачи:

- анализ предметной области;
- выбор и обоснование проектного решения;
- создание онлайн-сервиса сопровождения и коммуникации оказания услуг;
- тестирование и внедрение системы;
- оценка экономической эффективности проекта.

Быстрый темп в жизни человека является ключевым фактором в усвоении большого количества знаний и умений. Динамичное развитие событий в компаниях влечёт за собой высокие результаты в их деятельности. Это касается и людей. Поэтому есть необходимость увеличивать быстродействие во всех областях работы, особенно в интернете, так как человек проводит в нём большое количество времени. Создание легкого в понимании и освоении сервиса, а значит экономящего время и главное, позволяющего решить вопрос коммуникации производителя и потребителей услуг программного и аппаратного обеспечения, является актуальным.

1.2 Характеристика деятельности предприятия заказчика

Заказчиком проекта является Хакасский технический институт – филиал ФГАОУ ВО «Сибирский федеральный университет», расположенный по адресу: респ. Хакасия, г. Абакан, ул. Щетинкина, д. 27. Формирование института началось с середины шестидесятых годов двадцатого века с формирования Саянского промышленного территориально-производственного комплекса. Чтобы успешно решить задачи строительства, Хакасии потребовалась значительная кадровая перестройка. Было принято решение об открытии в г. Абакане общетехнического факультета (ОТФ) Красноярского политехнического института. На протяжении всего существования институт приобрёл широкую известность.

Согласно Федеральному закону от 29 декабря 2012 г. № 273-ФЗ "Об образовании в Российской Федерации" главы 3, статье 27, пункта 1 "Образовательные организации самостоятельны в формировании своей структуры, если иное не установлено федеральными законами", и пункта 2 "Образовательная организация может иметь в своей структуре различные структурные подразделения, обеспечивающие осуществление образовательной деятельности с учетом уровня, вида и направленности реализуемых образовательных программ, формы обучения и режима пребывания обучающихся (филиалы, представительства, отделения, факультеты, институты и т. д.)", Хакасский технический институт – филиал ФГАОУ ВО «Сибирский федеральный университет» имеет право осуществлять данную деятельность.

На рисунке 1 показана структурная схема института.

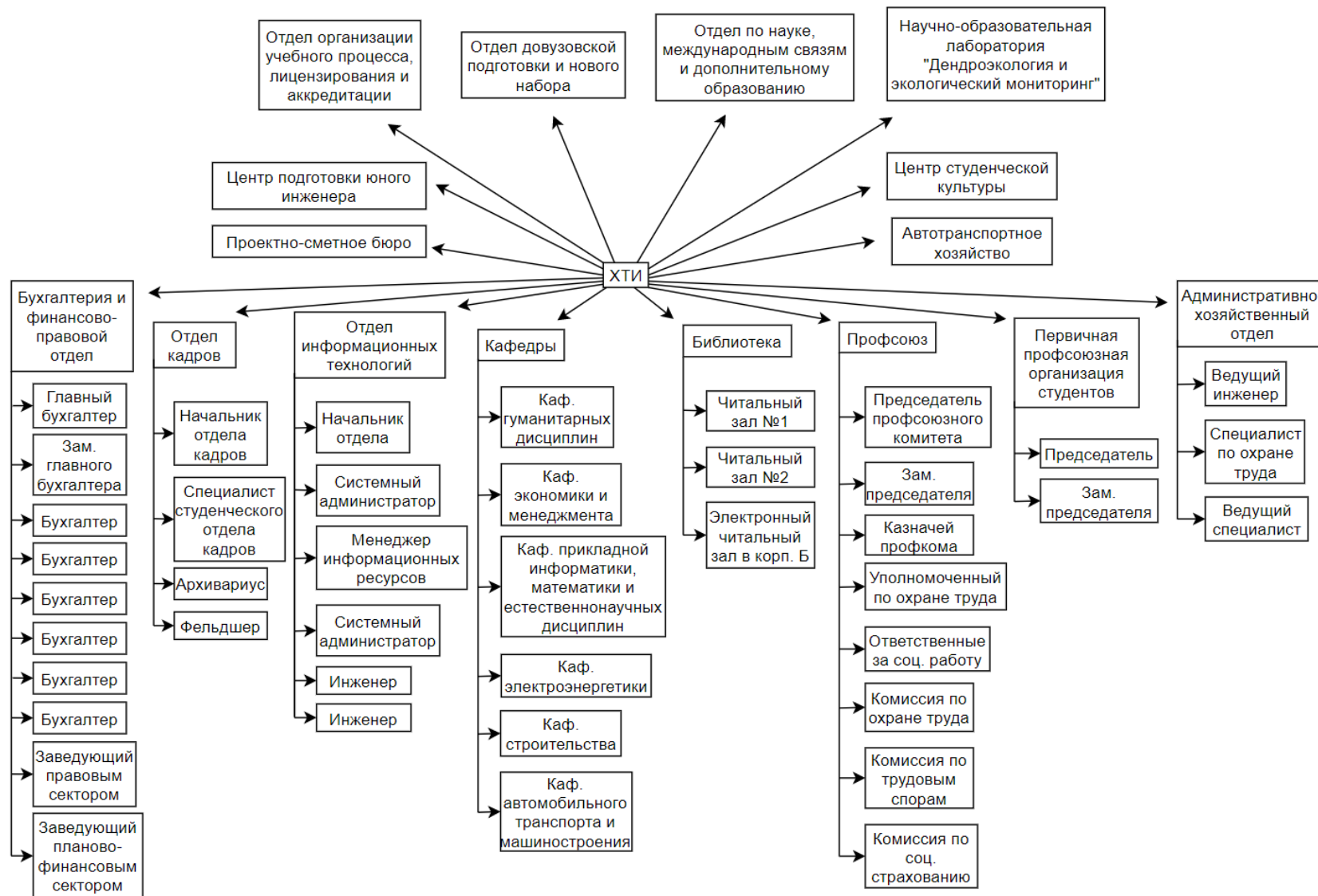


Рисунок 1 – Схема института

Одним из направлений обучения в институте является «Прикладная информатика». Учащиеся данного направления изучают область профессиональной деятельности, которая включает в себя:

- системный анализ прикладной области, формализация решения прикладных задач и процессов информационных систем;
- разработка проектов автоматизации и информатизации прикладных процессов и создание информационных систем в прикладных областях;
- выполнение работ по созданию, модификации, внедрению и сопровождению информационных систем и управление этими работами.

Студенты рассматривают прикладные и информационные процессы, занимаются информационными технологиями, изучают информационные системы.

Многие студенты имеют опыт работы как с прикладным ПО, так и периферийными устройствами. В связи с этим у будущих специалистов появляется возможность реализовать свои навыки на практике.

Помимо получения знаний в учебном заведении на стадии, особенно раннего обучения, реальное участие в какой-либо практической работе позволяет учащимся дать преимущества не только в своей сфере обучения, но и раскрывает возможности взаимодействия с работодателем. Говоря другими словами, коммуникация с заказчиком является ещё одним неизменным параметром для предстоящего самостоятельного жизненного пути по профессии. Взаимоотношения между исполнителем и клиентом это часть действий, без которых дальнейшее формирование профессиональной деятельности затруднительно. Хороший контакт между людьми даёт потенциальный приятный бонус в виде последующего сотрудничества.

Учащимся в высшем учебном заведении для более комфортного и удобного обучения нередко требуются материальные средства. Во время выполнения заказов с динамичной инициативой, у студентов появляется возможность получить больше положительных отзывов. Хорошие впечатления у клиентов после выполнения заказов формируются не только в сторону

исполнителей, но и в сторону учебного заведения. Поэтому, в связи с активным участием в данном деле, у студентов появляется возможность получить материальную поддержку от института. Также существует некоторая допустимость самих заказчиков, которая предполагает дополнительное вознаграждение в качестве, например, денежных средств.

На основании вышесказанного было принято решение разработать проект, в котором студенты смогут нарастить свой практический опыт, отработать навыки коммуникации.

Прикладное применение знаний и умений способствует более интенсивному развитию технических навыков и как следствие закреплению уже существующих и овладению новых. Вышеперечисленные последовательные аспекты дают вкуче логический вывод, который говорит о том, что при осуществлении данного проекта, его достоинство не может не снискать положительные стороны для студента.

Результат проекта – это сервис, обеспечивающий людям доступ к различным услугам программного и аппаратного обеспечения, способствует повышению информационной грамотности среди населения, а также, на основе исполнения работы, даёт эвентуальную вероятность изучения основам будущей деятельности для студентов.

1.3 Анализ рынка разрабатываемого продукта

Зачастую у людей не находится достаточного количества времени для того, чтобы уделять его на поиск какой-либо необходимой информации. Поэтому есть определённая потребность предоставлять гражданам нужные сведения с минимальной затратой времени. Создаваемый сервис идёт навстречу интересам общества, поэтому обеспечивает соответствующим предложением. Сервис будет функционировать при помощи сайта, тем самым предоставляя к нему свободный доступ каждому человеку. Интернет рынок занимает с каждым годом всё большую нишу. Это говорит о том, что веб-сайты имеют высокую

востребованность. С использованием интернет-сайтов оказание услуг вышло на новый уровень и стало возможным на расстоянии. Осуществление работы сервиса таким образом является привлекательной вариантом для выбора покупателей.

Разрабатываемый программный продукт, в частности веб-сайт, представляет собой SPA (single page application), обладающий некоммерческим характером и направленный на совершение покупки пользователями. Данный тип продукта имеет достаточное признание на рынке, поэтому приобрёл широкое распространение среди онлайн-сервисов. По данным статистики во всём мире пользуются интернетом почти 60% населения Земли или более чем 4,5 миллиарда человек. Это указывает на то, какой диапазон охватывает интернет-связь на планете. По причине широкого распространения всемирной паутины разработка сайта предстаёт релевантным решением. Необходимость в экономии времени и такое состояние как лень вызывают потребность человека использовать все доступные ресурсы для ухищрения в овладении новых знаний и умений. Иначе говоря, сайты, содержащие некую важную информацию или предоставляющие какую-либо помощь, помогают людям быстрее получать знания или выполнять ту или иную работу. Исходя из этого, востребованность данного вида услуг является довольно большой. Тем самым спрос на рынке становится весьма высоким.

Использование переносимого устройства по несколько часов в день для коммуникации, называемого смартфон, в современном мире является обыденным явлением. Нельзя недооценивать возможности этого аппарата как средства не только для познания окружающей информации, но и для создания её. Смартфон практически полностью заменяет стационарный компьютер. Это значит, что при помощи мобильного устройства есть возможность делать видео и фотографии с последующей передачей их в интернет. Сейчас на телефонах поддерживается множество сервисов, которые позволяют публиковать личный контент в сеть. Во время поиска необходимой информации существует возможность предоставлять её в наиболее благоприятных для пользователя

условиях. А именно, расположение требующихся сведений на одном ресурсе, чтобы быстро находить инструкции по эксплуатации того или иного устройства, или онлайн-сервиса. В связи с этим на сайте будет реализован функционал, содержащий ответы на часто задаваемые вопросы людей, относящиеся к телефонам. Благодаря чему, пользователю можно будет находить подобный материал в одном месте.

Пользователи должны находить места, где они способны узнавать нужные им данные. К примеру, можно взять официальный сайт компании Xiaomi, где в полном объёме представлен материал по эксплуатации смартфона. В компаниях аналогичной деятельности имеется такая же документация, однако их варианты в меньшей степени подходят под условия и желания клиентов. Бегло найти руководства пользователей, находящихся на различных сайтах, либо не всегда представляется возможным, либо в конечном итоге результат поиска составляет ноль информации.

Вследствие этого среди услуг на разрабатываемом сайте будет организована опция чата, которая будет осуществлять прямое общение с клиентами. Такого рода информирование человека позволяет донести быстро и кратко полезную информацию. Это способствует экономному расходу необходимого конструктивного промежутка времени клиента, который он мог бы провести в более плодотворной деятельности.

Похожая работа функционала осуществлена на многих предприятиях в городе Абакане.

Исследовав и проанализировав данный функционал, имеется возможность приобрести нужные сведения для последующей собственной разработки. Компания "Орион телеком", находящаяся по адресу: г. Абакан, ул. Карла Маркса, 44, наглядно показывает надлежащую миссию. Сайт предприятия работает в 6 городах, в том числе городе Абакан. Они оказывают услуги в сфере домашнего интернета, телефонии, компьютерной помощи и другие. Рассмотрев все плюсы и минусы качества их сервисов, впоследствии их учёт поможет форсировать разработку сайта. Похожую деятельность

выполняют и другие компании, которых в г. Абакан нашлось свыше 60 штук. В качестве конкурентов взяты компании, которые имеют сайт с большим набором услуг и опций, чем у других. Сравнение разрабатываемого сервиса и других компаний представлено в таблице 1.

Таблица 1 – Сравнение разрабатываемого продукта с конкурентами

Критерий	Орион Телеком	ПКСервис	Датацентр+	Мультимедиа	Разрабатываемый продукт
Онлайн-консультант	Есть	Нет	Нет	Нет	Есть
Количество услуг в обслуживании ПК и периферийных устройств	45	18	25	78	20
Личный кабинет	Есть	Нет	Нет	Нет	Есть
Стоимость услуг	Бесплатно/ платно	Бесплатно/ платно	Платно	Платно	Бесплатно

На основе данных в таблице 1 можно сделать вывод, что создаваемый продукт имеет приоритет в плане стоимости услуг среди трёх других рассматриваемых сервисов. Орион Телеком имеет некоторое преимущество среди всех, однако у разрабатываемого продукта бесплатная стоимость услуг. ПКСервис имеет меньше всех услуг, но позволяет тоже обслуживать бесплатно, в отличие от Датацентра+ и Мультимедии. Мультимедиа в свою очередь обладает самым большим количеством услуг, но все они платные. У разрабатываемого продукта имеется также немного услуг, но они в будущем будут пополняться.

Итогом анализа участников рынка разрабатываемого продукта является обоснование его проектного решения. Функционируя, сервис должен быть

выполнен в стиле ФГАОУ ВО «Сибирский федеральный университет», содержать регистрацию, авторизацию. Содержать каталог услуг. Иметь блок часто задаваемых вопросов. Обладать системой оценок сайта и услуг, а также обратной связью.

Студенты, обучающиеся по направлению «Прикладная информатика», окажут услуги по ремонту и обслуживанию любых видов компьютерной техники. Данную услугу можно получить посредством удалённого доступа, можно воспользоваться онлайн консультацией или вызвать специалиста на дом.

Будут оказываться следующие виды услуг:

- установка и настройка прикладного и офисного программного обеспечения;
- настройка и оптимизация работы операционных систем Windows 10/8/7;
- подключение и настройка периферийного оборудования;
- устранение любых неисправностей локальной сети;
- внедрение систем антивирусной защиты и организация процесса автоматического обновления антивирусных баз;
- модернизация компьютера;
- чистка компьютера от пыли;
- чистка файлов в смартфоне;
- и др.

На сайте будет располагаться раздел с контактной информацией, который включает в себя телеграмм канал, скайп, почту и номер телефона. Это поможет пользователям связаться с администратором для уточнения каких-либо вопросов, связанных с услугами.

Для того, чтобы узнать мнение пользователей на счёт сервиса, будет встроена опция, предполагающая оценку по пятибалльной шкале и отзыв в письменном виде, где пользователь может написать своё впечатление о сайте или услуге. Схема состава сайта представлена на рисунке 2.

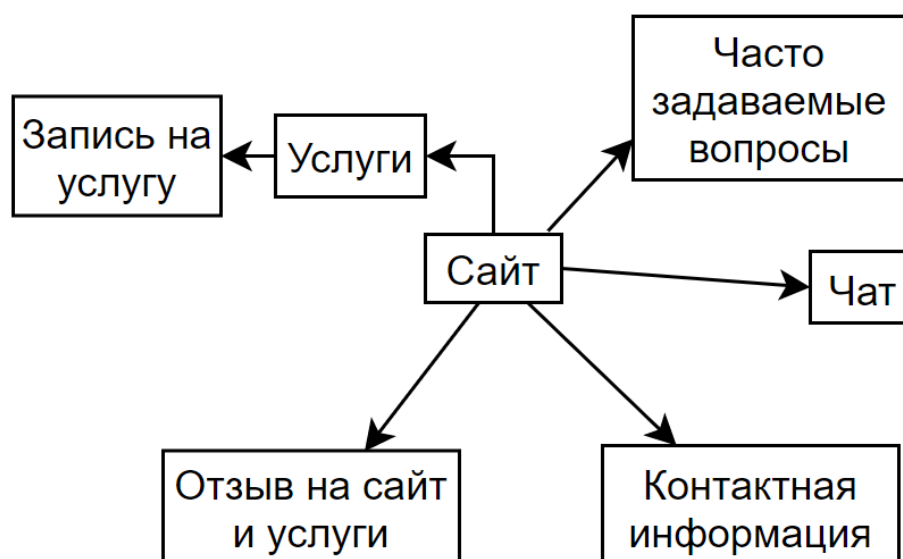


Рисунок 2 – Схема состава сайта

1.4 Сравнение средств разработки ИС

Сайт можно разработать с помощью разных языков, разными методами и в разных редакторах. Основным языком front-end для написания веб-страниц является HTML. HTML – это язык гипертекстовой разметки. Впоследствии он преобразуется в JSX разметку. Также будет использоваться SCSS – это язык описания внешнего вида документа. Является улучшенной версией обычного CSS. Данные два языка неотъемлемые части при разработке сайтов.

Расширить возможности сайта позволяет Javascript. Этот язык программирования позволяет оптимизировать работу веб-страниц, а также упрощает взаимодействие пользователя с ними. В частности будет использоваться React.js, который является javascript-библиотекой. Все элементы для сайта будут построены на React.js. В нём будет создаваться как front-end разработка, так и back-end. Он отлично подходит для создания современных одностраничных приложений любого размера и масштаба. Его применение гармонирует под SPA приложения.

Рассмотрим, какая база данных хорошо подходит под небольшое SPA приложение. Для одностраничных приложений хорошо подойдёт Firebase от

компании Google. Это платформа для разработки мобильных и веб-приложений. Она предоставляет разработчикам множество инструментов и услуг. Достоинств у этой платформы для целесообразного выбора достаточно. В ней содержатся такие особенности, как:

- возможность просматривать статистику сайта;
- база данных с изменениями в реальном времени;
- хранилище для файлов пользователей;
- сбор информации о сбоях;
- привязка учётных записей и др.

Для имеющихся обстоятельств бесплатной версии firebase будет хватать в полной мере. В противном случае есть возможность расширить функционал, купив тарифный план, выбрав необходимые продукты под свои интересы и платя только за фактическое использование.

Бесплатный тарифный план Firebase включает в себя:

- Около 20 миллионов сообщений в чате;
- около 600 тысяч записанных данных в день;
- около 2500 фотографий в высоком разрешении.

Firebase работает с JavaScript SDK и Node.js. SDK Javascript это расширение для Javascript, которое упрощает реализацию определённых функций. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера. Эта платформа позволяет писать серверный код для динамических веб-страниц и веб-приложений, а также для программ командной строки.

Инструментом создания сайтов является также среда разработки. Это может быть текстовый редактор или же полноценная IDE. Для небольших проектов подойдёт текстовый редактор. Для того, чтобы раскрыть его

потенциал, необходимо установить нужные модули. Они помогают в написании кода и обеспечивают его лучшее понимание. Один из популярных текстовых редакторов это бесплатный Visual Studio Code. Его возможности увеличиваются при установке бесплатных расширений. Использоваться для разработки будет именно данный программный продукт. Также необходим веб-браузер потому, что это неотъемлемый элемент как для пользования, так и для создания сайта. Разработка будет вестись на любом браузере, который обладает движком Chromium. Тестирование работ сайта будет проходить в каждом браузере, поскольку необходимо учитывать, что каждый пользователь имеет тот браузер, который ему удобен. Такие браузеры, как: Google Chrome, Yandex.Browser, Firefox, Opera, Internet Explorer, Microsoft Edge. Эти веб-обозреватели, основные, которыми пользуются люди.

Сравнение сред разработки представлено в таблице 2.

Таблица 2 – Сравнение сред разработки

Среда разработки	Цена	Функционал	Минусы
Visual Studio Code	Бесплатно	Большой	нет
WebShtorm	1-й месяц бесплатно, затем 129\$ в мес.	Большой	Цена
Sublime Text 3	Бесплатно	Средний	Недостаточно много функций
Atom	Бесплатно	Большой	Тяжёлая программа

Веб-сайт будет выполнен в стиле Single Page Application. Как преимущества, так и недостатки у этого вида разработки существуют.

Преимущества SPA:

1. Универсальность. SPA приложения одинаково хорошо функционируют на персональных компьютерах и на мобильных устройствах. На компьютерах, планшетах и смартфонах проекты, разработанные по принципу SPA, работают

без затруднений. Отсюда вытекает дополнительный плюс – это охват большей целевой аудитории, чем при использовании обычных методов веб-разработки.

2. Удобство использования. На одной странице вся информация подгружается без перезагрузки. Вместо того, чтобы переходить несколько раз по разным ссылкам на сайте для выполнения каких-либо действий, имеется возможность сделать их без переадресации. Все сведения находятся на одной странице, что даёт полную фокусировку в одной области работы.

3. Высокая производительность. В обычных сайтах очень часто можно встретить загрузку одного и того же содержимого. Например, шапка сайта, футер, меню и другие элементы, которые не меняются от страницы к странице, тем не менее, каждый раз загружаются с сервера. С использованием SPA подхода такой проблемы просто не будет, т.к. содержание будет подгружаться по мере необходимости, что значительно повысит скорость работы приложения.

Недостатки SPA:

1. Низкий уровень безопасности. JavaScript имеет низкий уровень безопасности, но если использовать современные фреймворки, они могут сделать веб приложение безопасным

2. Javascript. Некоторые пользователи отключают javascript в своих браузерах, а без него приложение не будет работать.

Разработать сайт по технологии SPA это не только заказ, но и наиболее подходящий вариант для такого типа сайта. SPA становится всё более признанным в сфере web-приложений. Его использование является очень частым. Он прост и быстр одновременно, поэтому его выбор наиболее соответствующий для эффективной эксплуатации.

Веб-сайты, созданные как SPA-приложения, сокращают ожидания во время загрузки. Сайт, работающий на SPA, использует единственный HTML-документ как оболочку для всех веб-страниц и организует взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript. SPA позволяет имитировать работу десктоп приложений. Архитектура устроена

таким образом, что при переходе на новую страницу, обновляется только часть содержания. Таким образом, нет необходимости повторно загружать одни и те же элементы. Данный принцип разработки очень удобен для разработчиков и пользователей.

1.5 Выводы по разделу «Анализ предметной области»

Сайт может дать людям дополнительный опыт в сфере цифровых технологий и компьютерной техники, а также возможность получать студентам опыт работы с заказчиком. Объём рынка электронной коммерции в России растёт с каждым годом. Это наглядно показывает высокую заинтересованность и потребность общества в электронных продуктах, которые имеют доступ в интернет практически с любого места пребывания потребителя.

Обобщая всё вышесказанное, можно сказать, что создавая данную информационную систему, есть возможность добиться успеха в сфере услуг. Её актуальность наглядно показывает, что подобного рода сервисы имеют спрос. Проанализировав рынок конкурентов и ознакомившись с аналогичными сервисами, можно сказать, что в разрабатываемой информационной системе появляются хорошие перспективы в оказании услуг студентами. А ИС служит для помощи людям в решении проблем технического обслуживания. Будет разработан сайт в стиле SPA приложения, написанный с помощью React.js, JavaScript SDK и Node.js, который будет находиться на платформе Firebase. Данное приложение будет создаваться в программе Visual Studio Code, которое является бесплатным. Тестирование будет вестись в браузере Yandex.Browser.

2. Проектирование ИС

2.1 Выбор жизненного цикла информационной системы

Для проекта наиболее подходящей моделью жизненного цикла стала спиральная. Данная модель подразумевает разработку в виде последовательности версий, но в начале проекта определены не все требования. Требования уточняются в результате разработки версий. В начале работы над проектом у заказчика и разработчика нет чёткого видения итогового продукта или стопроцентной уверенности в успешной реализации проекта. В связи с этим принимается решение разработки системы по частям с возможностью изменения требований.

Жизненный цикл ИС представлен на рисунке 3.

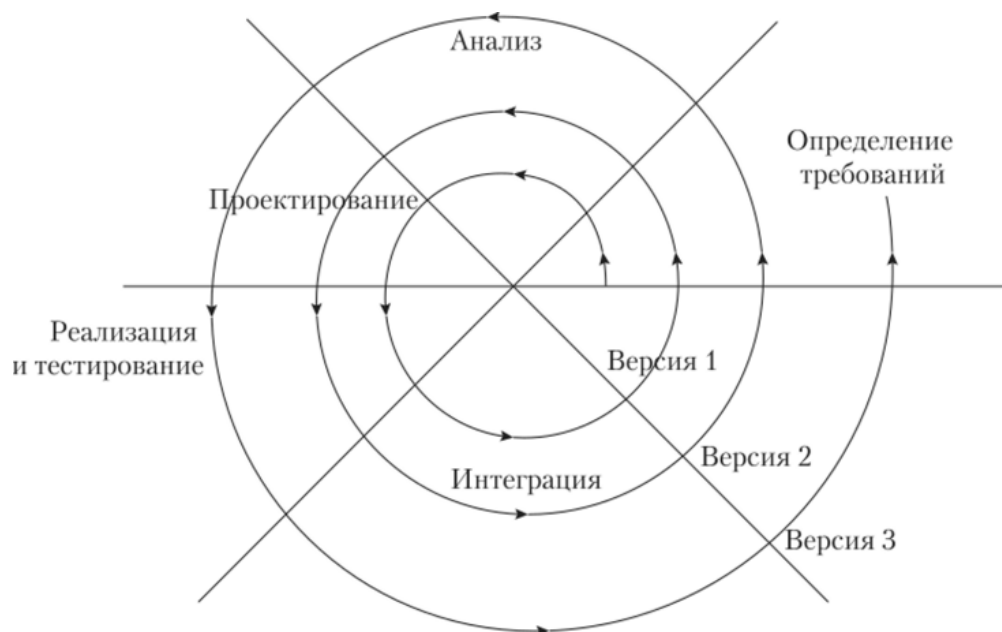


Рисунок 3 – Спиральная модель жизненного цикла

2.2 Модель ИС в нотации IDEF0

Чтобы понимать как работает информационная система, была составлена диаграмма IDEF0, представленная на рисунке 4.

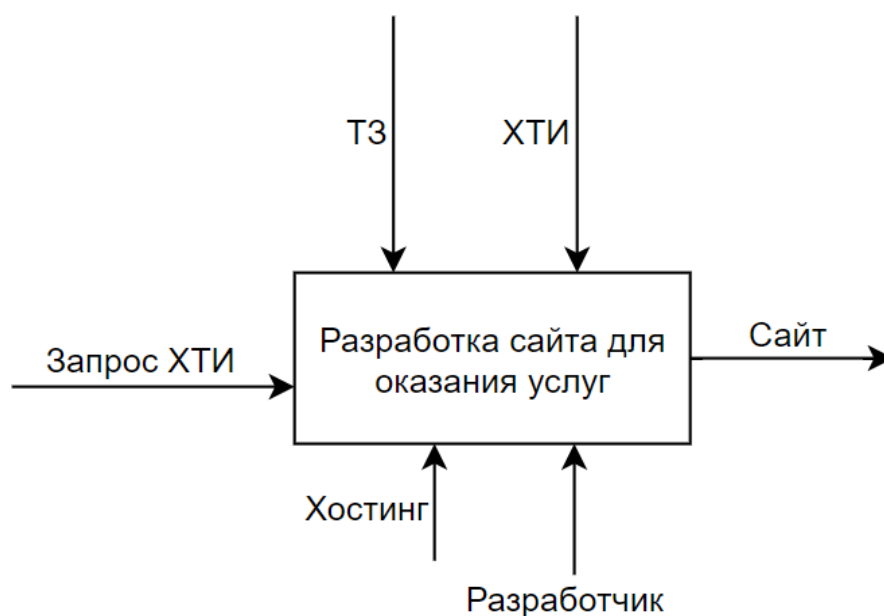


Рисунок 4 – Диаграмма IDEF0

Входами диаграммы являются:

Запрос ХТИ – необходимостью создания проекта было требование Хакасского технического института – филиала ФГАОУ ВО «Сибирский федеральный университет».

Управлением диаграммы являются:

– Техническое задание – в данном пункте прописаны основные требования по разработке проекта.

– ХТИ – филиал СФУ – институт предъявляет дополнительные требования и условия в функционировании сайта.

Механизмами диаграммы являются:

– Хостинг – хост необходим для работы сайта и поддержания её в дальнейшем;

– разработчик – человек, который будет строить внешний вид сайта и писать алгоритмы, по которым будет работать сайт;

Выходами диаграммы является:

– Сайт – разработанный интернет-сайт, готовый к использованию.

2.3 Развернутая диаграмма IDEF0

На рисунке 5 представлена развёрнутая диаграмма IDEF0.

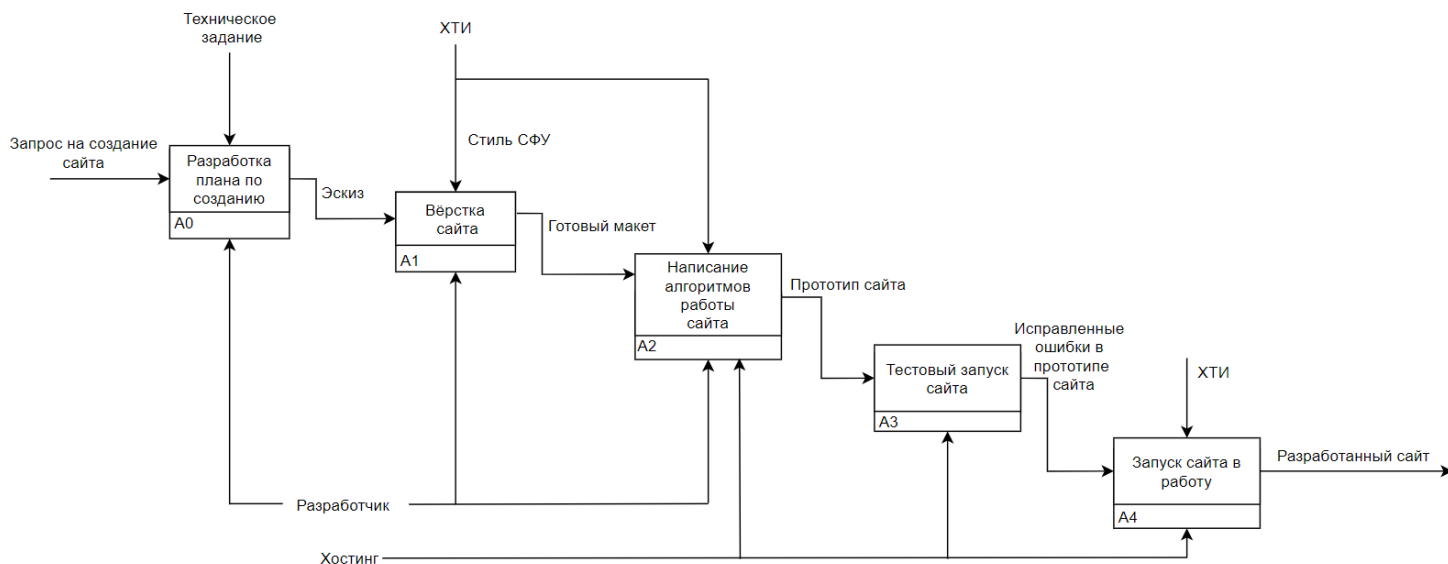


Рисунок 5 – Развернутая диаграмма IDEF0

С самого начала, с блока A0 «Разработка плана по созданию сайта» обдумывается весь план по реализации будущей работы. То есть то, как будет выполняться вся разработка сайта.

На входе этого блока идёт «Запрос на создание сайта». Это требование на представление сайта студентам-исполнителям институту.

Управлением является «Техническое задание», которое необходимо для согласования всех правил по выполнению работы.

Механизмом является разработчик. Он составляет план, по которому будет идти в процессе создания сайта.

Дальше идёт блок A1 «Вёрстка сайта». Здесь начинается сам процесс создания сайта. А именно внешний вид, это то, как будет выглядеть сайт для пользователя.

На входе стоит «Эскиз». По нему будет идти вёрстка сайта.

В управлении располагается «ХТИ – филиал СФУ». От требований заказчика будут зависеть некоторые элементы дизайна. В данном случае будет применяться стиль СФУ.

В механизме содержится «Разработчик». Он будет верстать сайт.

Следующий блок А2 «Написание алгоритмов работы сайта». В этом блоке описывается работа сайта. То есть то, как будет функционировать сайт, как будет проходить его работа.

На входе «Готовый макет». По макету будет идти программирование или бэкенд-разработка.

В управлении остаётся «ХТИ – филиал СФУ». Здесь институт также будет диктовать свои условия.

В механизме находятся «Разработчик» и «Хостинг». Разработчик будет писать код. Хостинг необходим для проверки работоспособности сайта. В дальнейшем он будет использоваться постоянно.

Предпоследний блок А3 «Тестовый запуск сайта» обозначает первый запуск сайта в работу. Он нужен для того, чтобы посмотреть, как сайт будет функционировать. И если возникнут ошибки в работе, то они будут скорректированы.

На входе «Прототип сайта». Без готового прототипа тестовый запуск не имеет смысла. Так как проверять готовый продукт следует тогда, когда он закончен.

Последний блок А4 – это «Запуск сайта в работу». После всех испытаний запускается процесс эксплуатации. На этом этапе начинается полноценная работа сайта.

На входе идёт «Исправленные ошибки в прототипе сайта». После тестового запуска возникают ошибки в работе, которые нужно обязательно исправить.

В управлении стоит «ХТИ – филиал СФУ». Так как заказчиком является Хакасский технический институт – филиал ФГАОУ ВО «Сибирский федеральный университет», то этот сайт принадлежит этому учебному

заведению и практически всю деятельность сайта будет осуществлять данное учреждение.

На выходе стоит уже готовый продукт, то есть разработанный сайт, который полностью идёт в эксплуатацию.

2.4 Модель ИС в нотации IDEF3

Основой модели IDEF3 служит так называемый сценарий бизнес-процесса, который выделяет последовательность действий или подпроцессов анализируемой системы.

Сценарий для большинства моделей должен быть документирован. Важным является понимание цели моделирования — набора вопросов, ответами на которые будет служить модель.

Была построена диаграмма IDEF3, чтобы наглядно иллюстрировать сценарий работы пользователя с сайтом.

В данной методологии показано взаимодействие пользователя с сайтом, то есть как он будет осуществлять покупку услуги.

Идёт описание технологических процессов, с указанием того, что происходит на каждом этапе технологического процесса. А также описание переходов состояний объектов, с указанием того, какие существуют промежуточные состояния у объектов в моделируемой системе.

Модель ИС нотации IDEF3 представлена на рисунке 6.

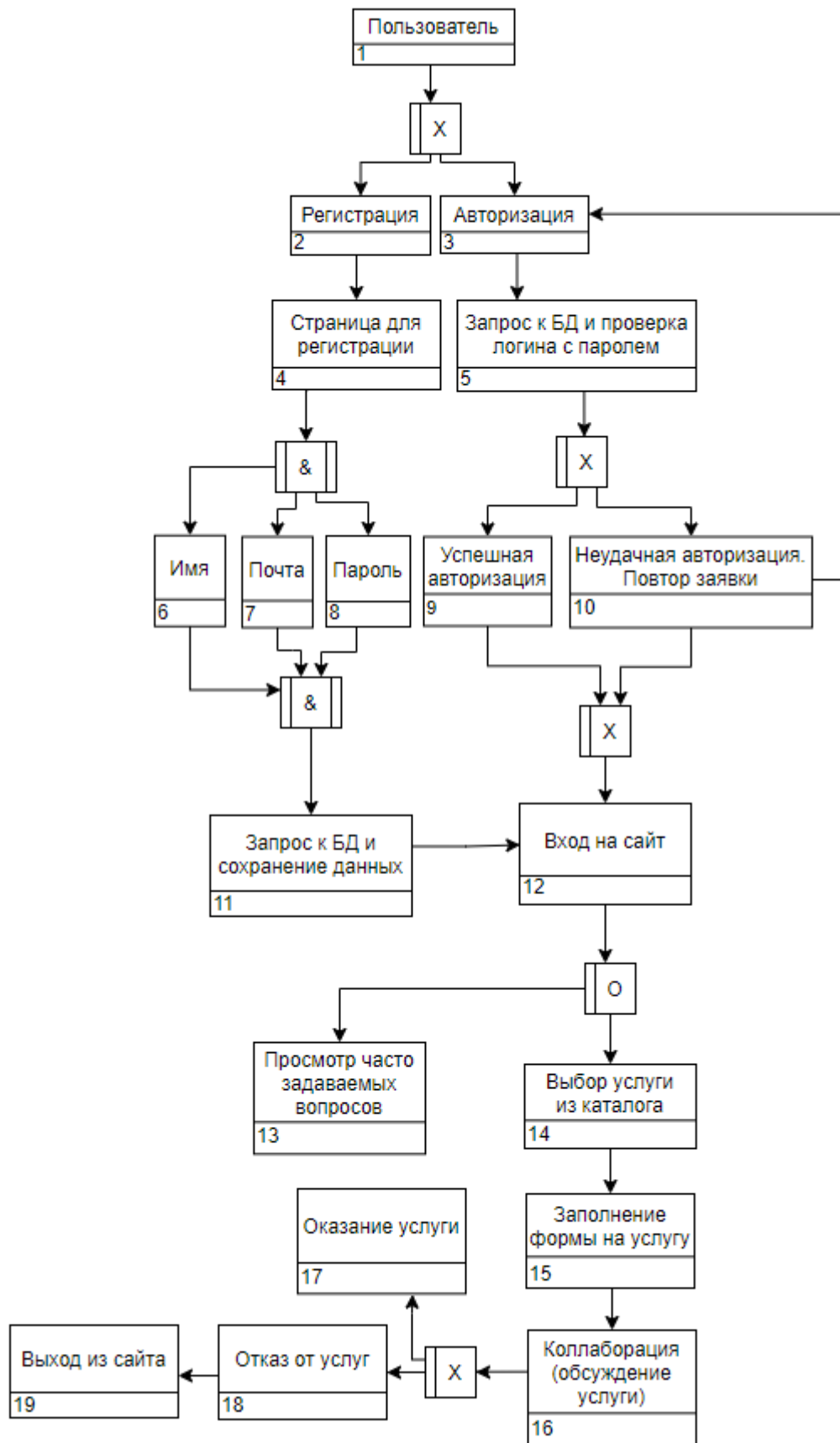


Рисунок 6 – Диаграмма IDEF3

Сначала пользователь для просмотра каких-либо услуг заходит на веб-страницу. Чтобы сделать заказ необходимо авторизоваться или, если нет аккаунта, зарегистрироваться. Во время регистрации открывается соответствующая страница (блок 4), в которой клиенту требуется указать имя, почту и пароль. После предыдущих действий введённые данные сохраняются в базу данных (блок 11) и пользователь войдёт в свой аккаунт (блок 12). В случае с авторизацией потребуется ввести свои данные для входа. Эти данные сравнятся с содержащимися данными в базе. После обнаружения аналогичных данных будет открыт доступ к собственному аккаунту пользователя (блок 9) или закрыт, если пользователь не зарегистрирован или неправильно ввёл личные данные (блок 10). По завершении входа на сайт клиент может выбрать услугу из каталога или посмотреть часто задаваемые вопросы, чтобы найти ответ на свой вопрос в них. После выбора услуги клиент может заполнить форму, в которой опишет свою проблему (блок 15). Затем произойдёт сотрудничество между клиентом и администратором (блок 16), в ходе которой клиент согласится на оказание услуги (блок 17) или же откажется с последующим выходом из сайта (блоки 18, 19).

2.5 Модель ИС в нотации DFD

Методология DFD необходима, чтобы описывать внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

Применяется этот вид нотации в случае, когда требуется описание системы как хранилища данных. Нотация должна наглядно показать из чего состоит информационная система и что нужно, чтобы обработать информацию.

Методология DFD представлена на рисунке 7.

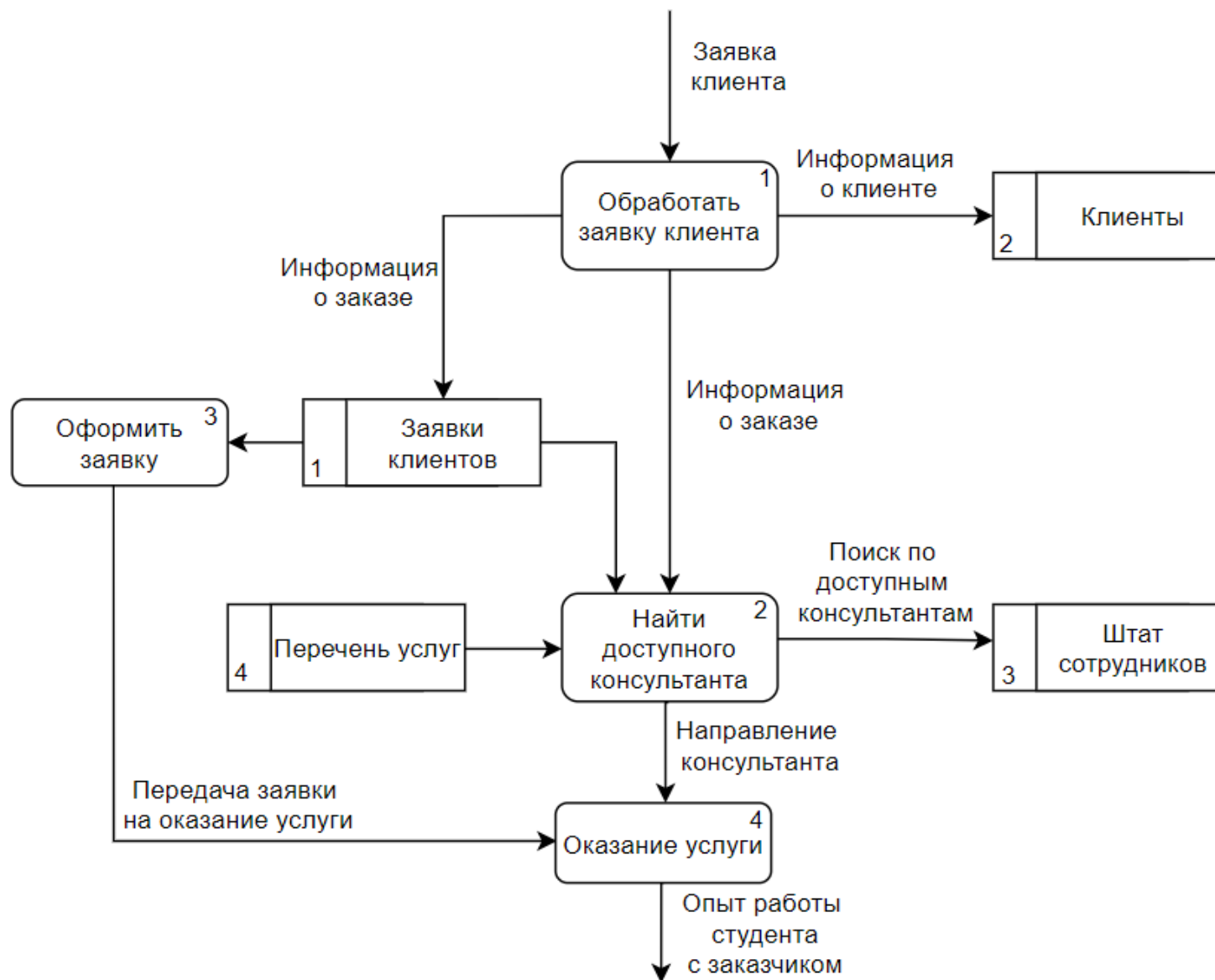


Рисунок 7 – Методология DFD информационной системы

Вначале клиент подаёт заявку на свой заказ, которая в дальнейшем обрабатывается (процесс 1). Далее поступают все заявки клиентов (накопитель данных 1), то есть туда, куда идёт вся информация обо всех заказах. А именно название услуги, которую выбрал клиент. Также сохраняется вся информация о клиенте (накопитель данных 2). Это контактная информация – номер телефона и адрес электронной почты. После приёма заявок начинаются оформление заявок (процесс 3) и поиск доступных студентов (процесс 2) для выполнения услуги. Оформление заявок необходимо для передачи заявки студенту об услуге. Поиск свободных людей для выполнения услуги выбирается из всех желающих работать (накопитель данных 3). Перечень услуг (накопитель данных 4) помогает понять, смогут ли консультанты выполнить данный заказ. После выбора консультанта его направляют на выполнение услуги, который в дальнейшем получает необходимый опыт работы во время выполнения заказа.

Сайт выполнен в стиле SPA – Single Page Application. SPA-приложение позволяет получать беглость в работе сайта при его использовании. Это достигается путём разных фреймворков и библиотек. Одной такой библиотекой является React. React – это javascript-библиотека с открытым исходным кодом. Написание кода на react сегодня является одним из самых популярных и востребованных способов создания приложений. Он позволяет упрощать разработку, а также создавать не только десктопные приложения, но и мобильные. Таким образом, сайт, сделанный на SPA технологии, написан при помощи react.

Сайт, включающий процесс регистрации, должен обладать собственной базой данных, чтобы хранить информацию пользователей. Аутентификация, хранение файлов, база данных и много другое содержится в одном сервисе – firebase. Firebase – это мобильная платформа компании Google, которая помогает быстро разрабатывать высококачественные приложения. В firebase встроены разные функции, которые можно сочетать в соответствии с требованиями.

2.6 Разработка проекта

Для создания сайта необходим Node.js. Node.js – это кроссплатформенная среда выполнения для JavaScript, работающая на серверах. В основном она используется для создания веб-серверов, однако сфера её применения этим не ограничивается. В среде Node.js выполняется код, написанный на JavaScript. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Создание локального сервера и необходимые модули станут неотъемлемой частью разработки.

Первое, что нужно сделать, это скачать Node.js. Для этого нужно зайти на сайт <https://nodejs.org/ru/> и выбрать текущую версию.

Окно с выбором загрузки Node.js представлено на рисунке 8.

Node.js® – это JavaScript-окружение построенное на движке Chrome V8.

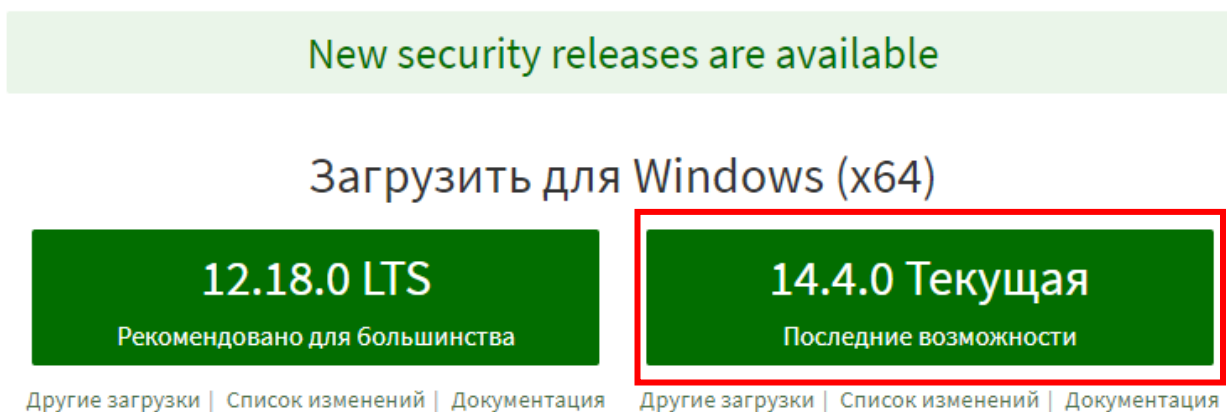


Рисунок 8 – Загрузка Node.js

После этого нужно сохранить инсталлятор на компьютер и запустить его. После установки Node.js появится возможность запускать локальный сервер, где и будет происходить разработка сайта. Далее необходимо открыть консоль windows и зайти в папку будущего проекта при помощи команды «cd

путь_к_папке», и написать в ней команду «`npm install -g create-react-app`», которая установит react. Далее ввести команду «`create-react-app имя_папки`», которая установит проект react на компьютер. В папке появятся следующие файлы.

Папка с файлами проекта представлена на рисунке 9.

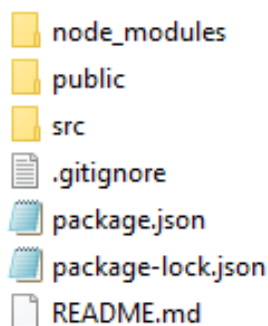


Рисунок 9 – Начальные файлы проекта

Чтобы запустить приложение, достаточно ввести команду «`npm start`», после этого браузер откроет локальный сервер localhost:3000. На этом сервере будет происходить дальнейшая разработка.

Так как приложение связано с платформой firebase, которая является собственностью Google, то необходимо иметь аккаунт в Google. После нужно создать проект в firebase. Для этого необходимо зайти на сайт <https://firebase.google.com/> и кликнуть по кнопке «Get started».

Сайт Firebase представлен на рисунке 10.

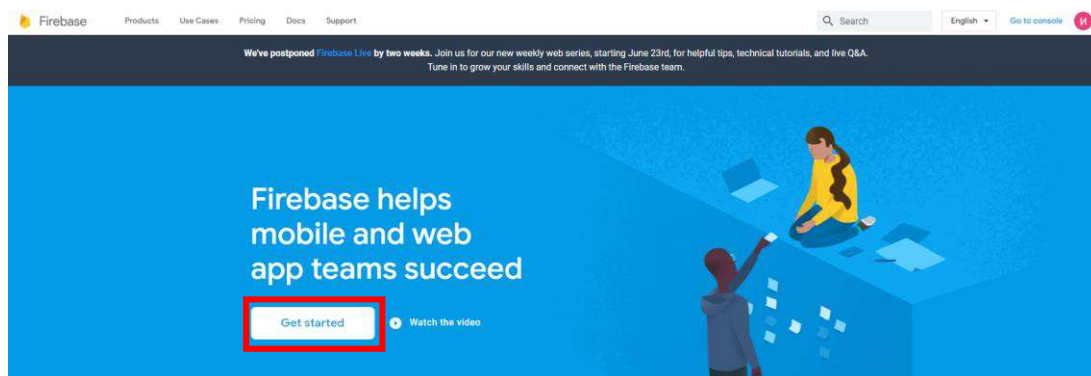


Рисунок 10 – Начало работы с firebase

Откроется окно с предложением создать проект, необходимо кликнуть по кнопке «Создать проект». После этого нужно указать название проекта, согласиться с условиями пользования Firebase и кликнуть по кнопке «Продолжить», затем «Далее» Далее указать страну проживания, принять все условия пользования и нажать «Создать проект».

Окно с созданием проекта представлено на рисунке 11.

Настройка Google Аналитики

Местоположение Аналитики ⓘ

Россия

Настройки доступа к данным и Условия использования Google Аналитики

- Использовать настройки по умолчанию для доступа к данным Google Аналитики [Learn more](#)
 - ✓ Разрешить Google использовать ваши данные Analytics для совершенствования продуктов и сервисов
 - ✓ Разрешить Google использовать ваши данные Analytics для сравнения
 - ✓ Разрешить Google использовать ваши данные Аналитики для предоставления технической поддержки
 - ✓ Разрешить доступ к вашим данным Analytics специалистам Google по работе с аккаунтами
- Я принимаю [Условия защиты данных при взаимодействии между контролерами данных](#) и обязуюсь соблюдать [Правила в отношении согласия пользователей из ЕС](#). Требуется, чтобы разрешить Google использовать ваши данные Google Аналитики для совершенствования продуктов и сервисов. [Подробнее...](#)
- Я принимаю [Условия использования Google Аналитики](#)

При создании проекта в будет добавлен новый ресурс Google Аналитики, который будет связан с вашим проектом Firebase. После установления связи начнется обмен данными между этими сервисами. Данные, экспортируемые из ресурса Google Аналитики, должны соответствовать Условиям использования Firebase, а данные, импортированные из Firebase, – Условиям использования Google Аналитики. [Подробнее...](#)

[Назад](#) [Создать проект](#)

Рисунок 11 – Настройка Google Аналитика

Следующим шагом нужно добавить новый проект в firebase. Для этого необходимо войти в своё приложение и выбрать «Веб-приложение»

Добавление нового сервиса представлено на рисунке 12.

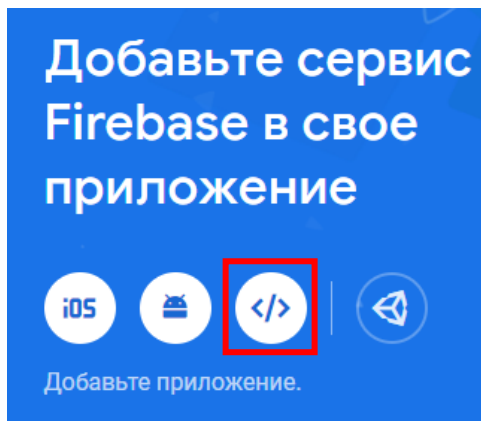


Рисунок 12 – Добавление нового сервиса

Начнётся регистрация. Для этого надо указать псевдоним приложения и кликнуть по кнопке «Зарегистрировать приложение». Далее пропущено следующее действие, оно будет выполнено в будущем, но другим способом.

Данные для БД представлены на рисунке 13.

Скопируйте и вставьте эти скрипты в конец тега <body> перед сервисами Firebase:

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.15.0/firebase-app.js"></scrip

<!-- TODO: Add SDKs for Firebase products that you want to use
https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/7.15.0/firebase-analytics.js"><

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyBckeS5nVbRD4x8Vtcky-X1fmeT2Qm4I_A",
    authDomain: "qwer-921c0.firebaseio.com",
    databaseURL: "https://qwer-921c0.firebaseio.com",
    projectId: "qwer-921c0",
    storageBucket: "qwer-921c0.appspot.com",
    messagingSenderId: "809842709125",
    appId: "1:809842709125:web:f7cb58e5d5a6c87ef70fec",
    measurementId: "G-LKJV0NJ0G1"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>
```

Рисунок 13 – Связующие данные для БД и проекта

Так как на сайте будет реализована функция входа по почте и паролю, необходимо включить данный параметр. Для этого во вкладке «Authentication» нужно нажать на «Настроить способ входа».

Раздел «Authentication» представлен на рисунке 14.

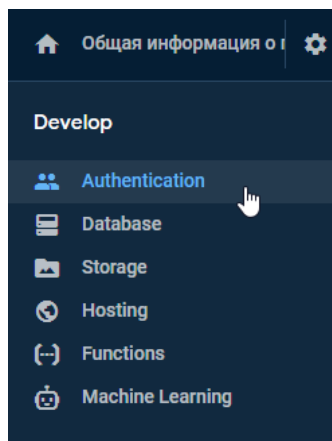


Рисунок 14 – Раздел «Authentication»

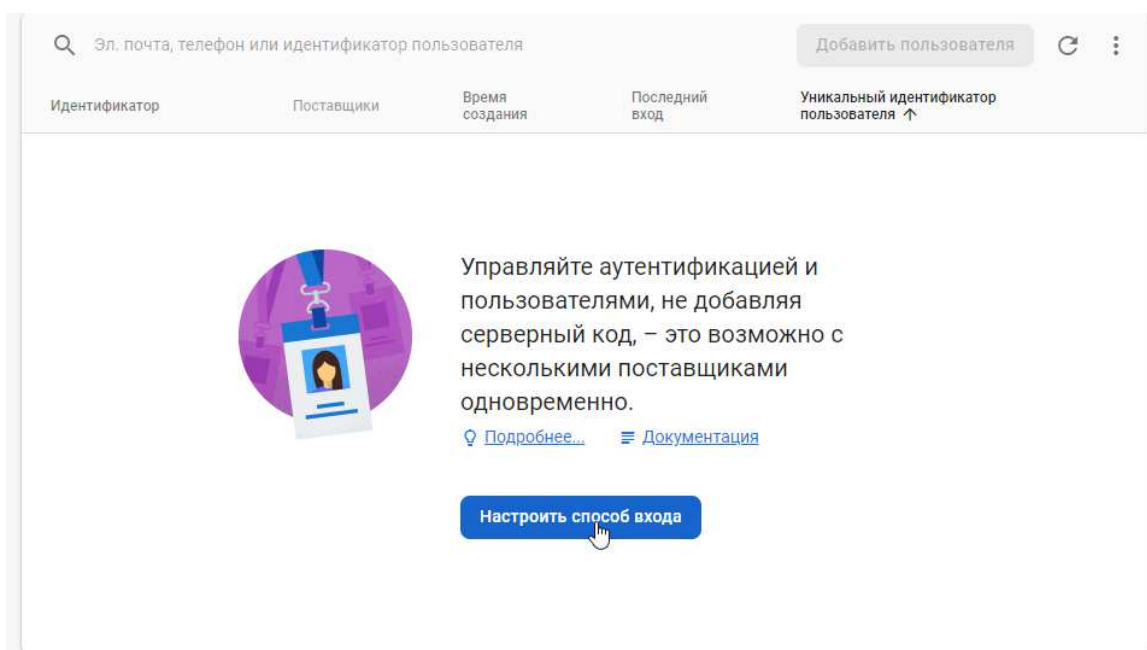


Рисунок 15 – Настройка способа входа

Выбрать «Адрес электронной почты и пароль».

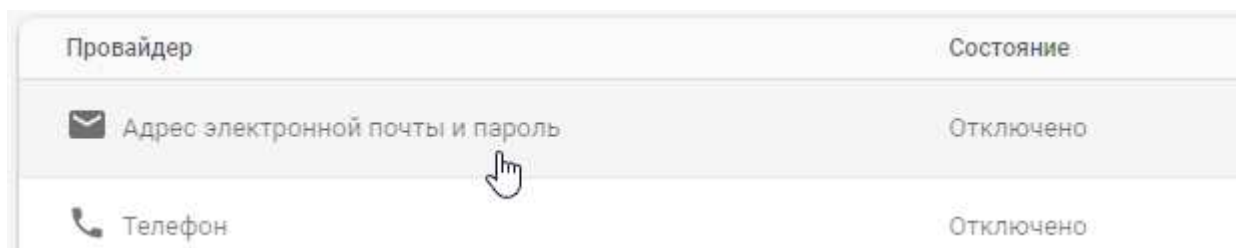


Рисунок 16 – Выбор электронной почты и пароля

Включить первый переключатель, затем сохранить изменения.
Вернёмся к firebase SDK, чтобы связать наш firebase с нашим приложением.
Для этого нужно зайти в настройки проекта.

Раздел с настройками представлен на рисунке 17.

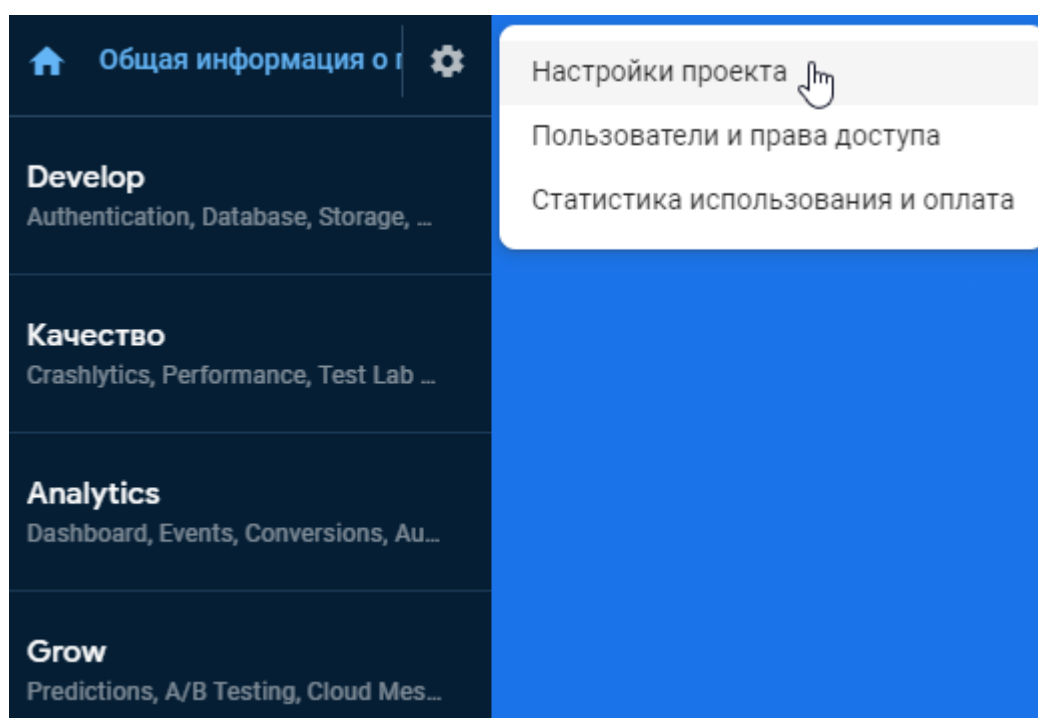


Рисунок 17 – Раздел «Настройки проекта»

Внизу важно выбрать «Конфигурация» и скопировать скрипт, указанный на скриншоте.

Скрипт представлен на рисунке 18.

Firestore SDK snippet

SDK ? 1 Конфигурация ?

Скопируйте и вставьте эти скрипты в конец тега <body>, но перед фрагментами кода сервисов Firebase:

```
const firebaseConfig = {
  apiKey: "AIzaSyBckeS5nVbRD4x8Vtcky-X1fmeT2Qm4I_A",
  authDomain: "qwer-921c0.firebaseio.com",
  databaseURL: "https://qwer-921c0.firebaseio.com",
  projectId: "qwer-921c0",
  storageBucket: "qwer-921c0.appspot.com",
  messagingSenderId: "809842709125",
  appId: "1:809842709125:web:f7cb58e5d5a6c87ef70fec",
  measurementId: "G-LKJV0NJ0G1"
};
```

2

Рисунок 18 – Данные для связи БД с приложением

После этого в папке «src» необходимо создать дополнительную папку с названием «db» и в этой папке создать файл index.js, в котором нужно написать код с последующей вставкой скрипта.

Скрипт в коде представлен на рисунке 19.

```
1
2
3 import firebase from 'firebase/app'
4 import 'firebase/firestore'
5
6 const db = firebase
7   .initializeApp(
8   {
9     apiKey: "AIzaSyDXBbQEJzU8Z0g7-Dvq6hC-aTsboBxZ0Ck",
10    authDomain: "site-hti.firebaseio.com",
11    databaseURL: "https://site-hti.firebaseio.com",
12    projectId: "site-hti",
13    storageBucket: "site-hti.appspot.com",
14    messagingSenderId: "454492947087",
15    appId: "1:454492947087:web:0113676c64c290fb3ee6fa",
16    measurementId: "G-5EBJT0ZZCJ"
17  }
18 )
19 .firestore()
20
21 export default db
22
23 const { Timestamp } = firebase.firestore
24 export { Timestamp }
25
```

Рисунок 19 – Код в проекте

Чтобы взаимодействовать с данными в базе данных, а именно их отправка, получение, хранение и изменение, необходимо создать саму базу данных. Для этого нужно перейти слева на вкладке «Develop» в раздел «Database».

Раздел «Database» представлен на рисунке 20.

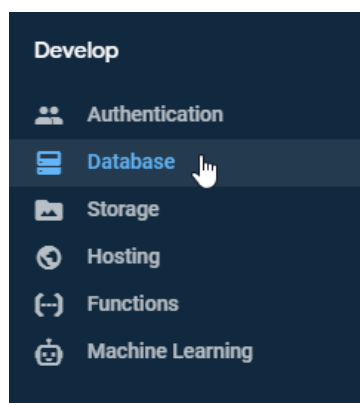


Рисунок 20 – Раздел «Database»

После перехода появляется возможность создать базу данных по кнопке «Создать базу данных».

Раздел создания базы данных представлен на рисунке 21.

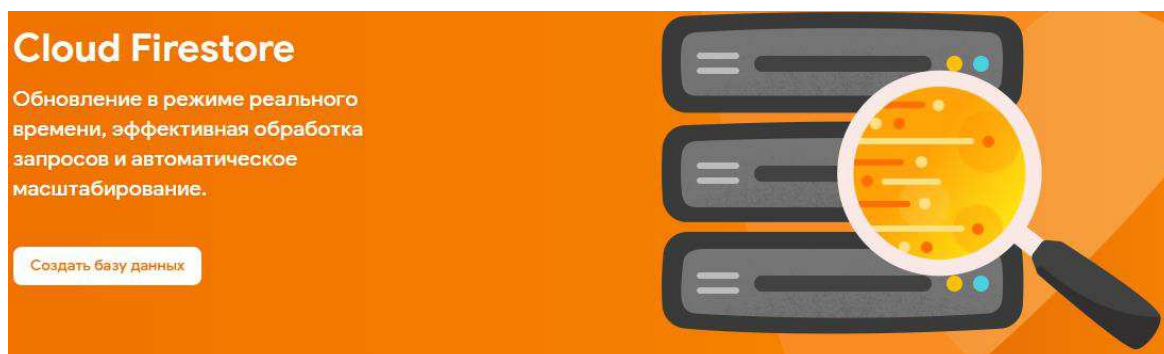


Рисунок 21 – Раздел создания базы данных

Ниже находится альтернатива в виде другой базы Realtime Database.

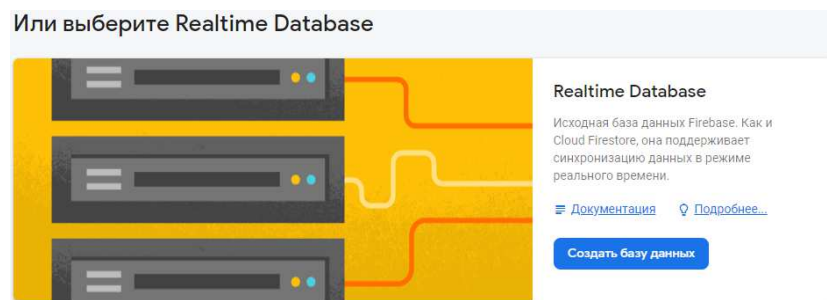


Рисунок 22 – База данных Realtime Database

Для разработки была выбрана Cloud Firestore. Отличия между этими базами данных имеются и среди них есть те, которые повлияли на выбор в сторону первой. Одна из особенностей в Cloud Firestore это более структурированные данные. Realtime Database представляет собой гигантское дерево из данных в формате JSON, в то время как в Cloud Firestore данные состоят из документов и коллекций, которые могут содержать другие документы. Таким образом можно обращаться к конкретному документу в базе, не захватывая ненужные данные. Во-вторых данные в Cloud Firestore, удобно просматривать и редактировать, что делает разработку более удобной.

Чтобы создать базу данных, нужно кликнуть по кнопке «Создать базу данных».

Кнопка создания базы данных представлена на рисунке 23.

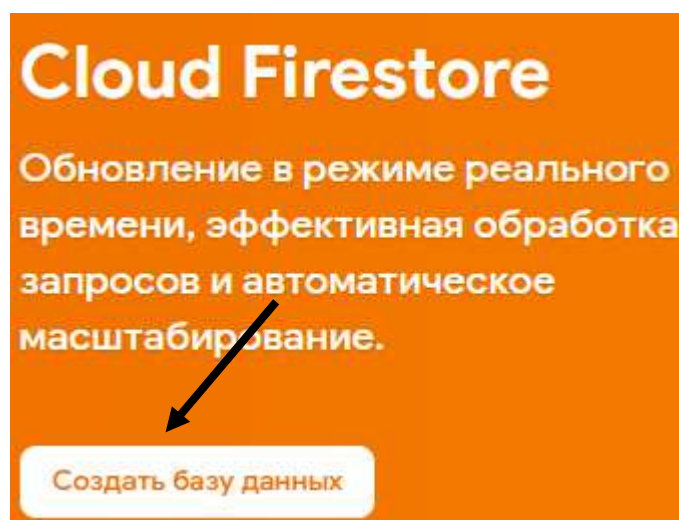


Рисунок 23 – База данных Cloud Firestore

Появится следующее окно, в котором нужно нажать «Дальше», затем «Готово».

Кнопка создания базы данных представлена на рисунке 24.

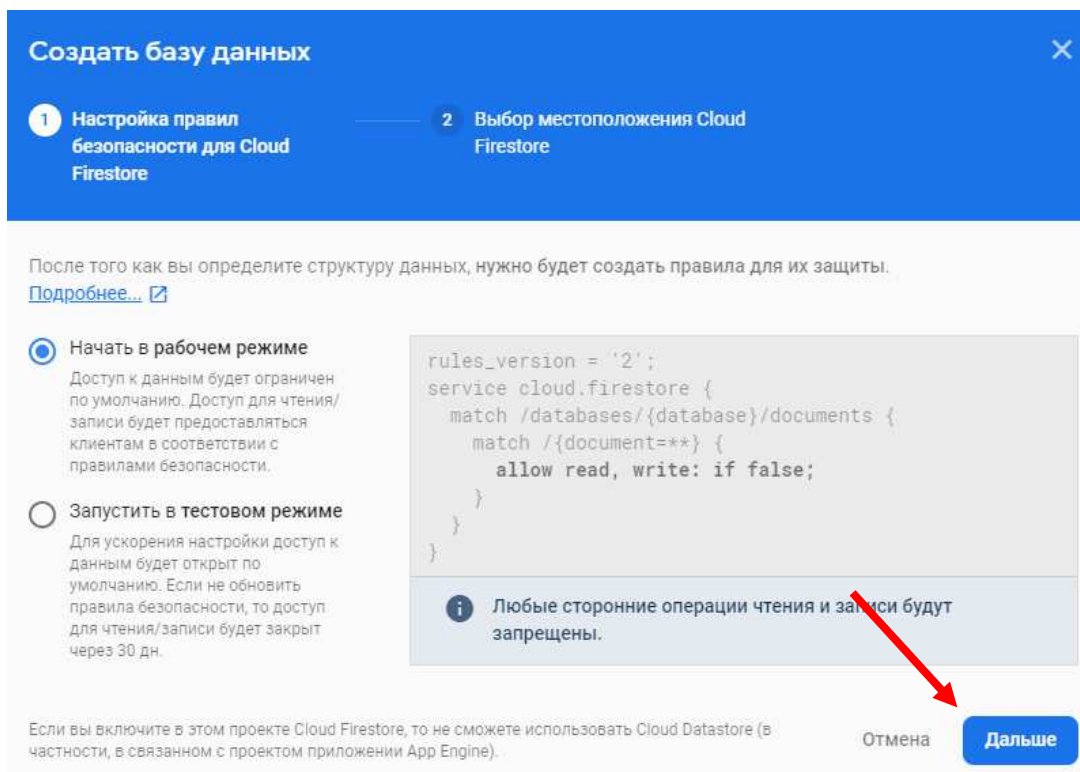


Рисунок 24 – Настройка правил безопасности в базе данных

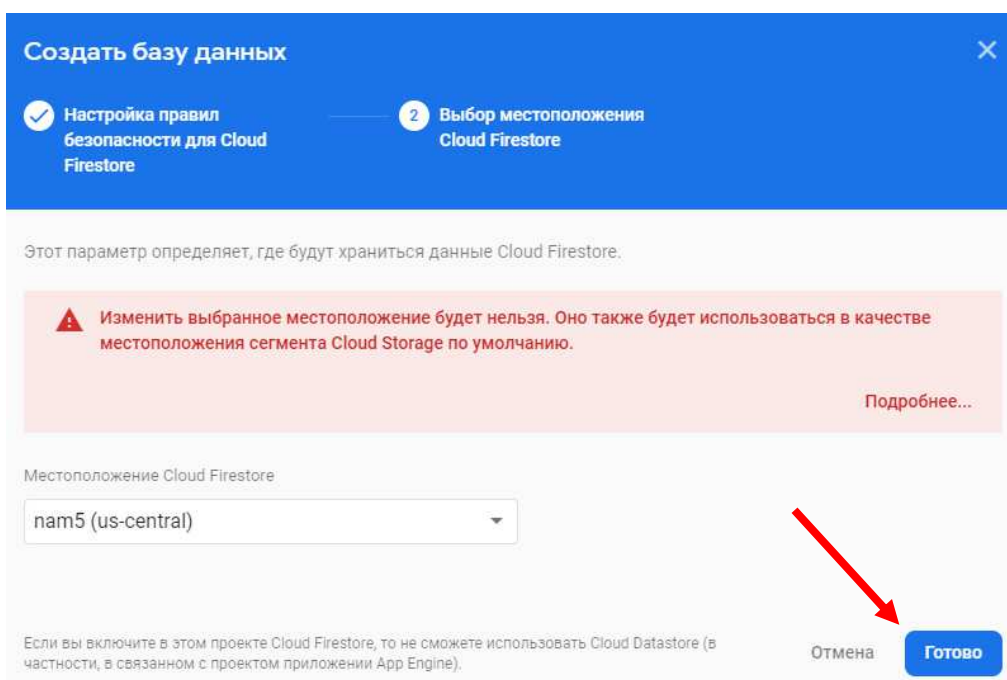


Рисунок 25 – Выбор местоположения в базе данных

После инициализации и настройки откроется пустая база данных. Так как на сайте будут аккаунты, необходимо добавить новую коллекцию, в которую будут заполняться профили пользователей. Для этого нужно кликнуть на кнопку «Добавить коллекцию».

Добавление новой коллекции представлено на рисунке 26.

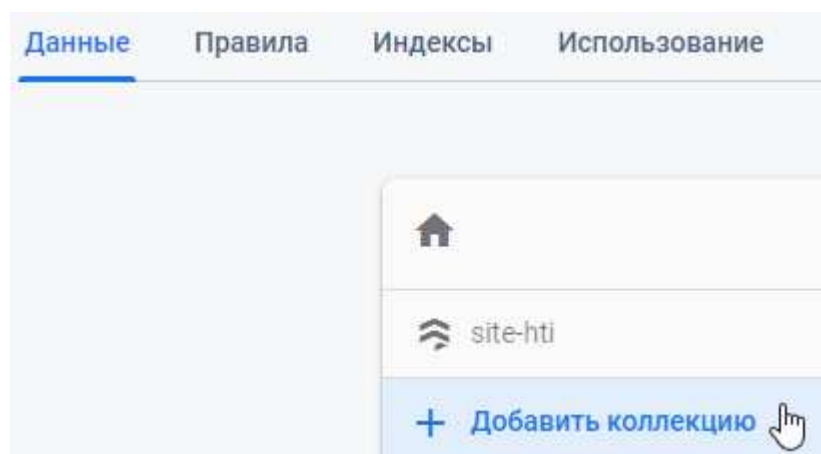


Рисунок 26 – Добавление новой коллекции в БД

Откроется окно, в котором нужно ввести идентификатор коллекции, это будет «profiles», и нажать на кнопку «Далее».

Добавление идентификатора коллекции представлено на рисунке 27.

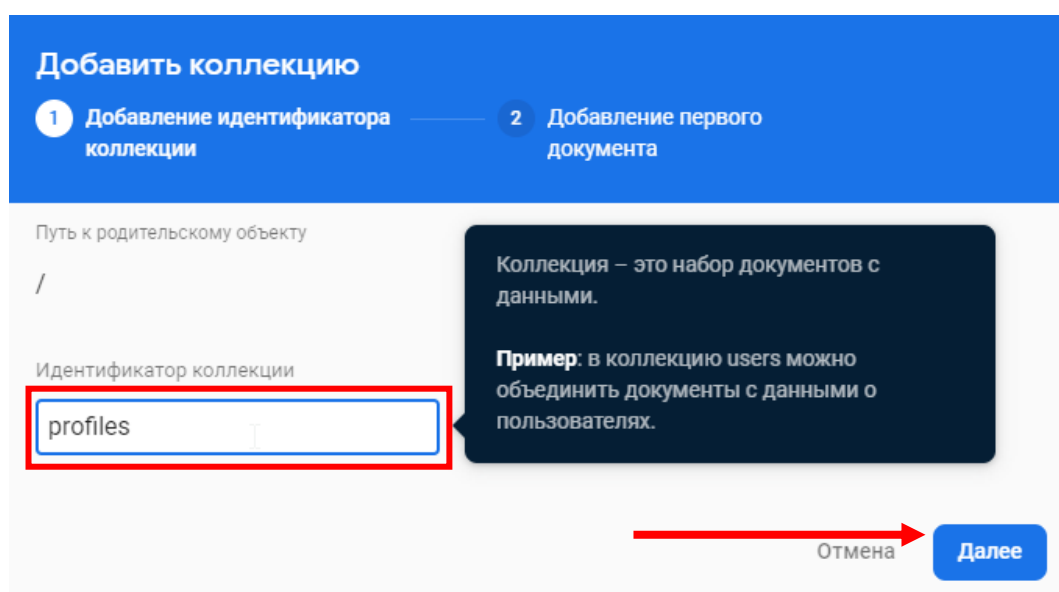


Рисунок 27 – Добавление идентификатора коллекции

Следующим пунктом стоит добавление документа. У каждого документа имеется свой идентификатор. Его можно ввести вручную или кликнуть по кнопке «Сгенерировать» для генерации. Второй вариант является предпочтительней, так как идентификатор создается полностью уникальным, а пользователей много, соответственно для каждого пользователя разный идентификатор.

Добавление первого документа представлено на рисунке 28.

Добавить коллекцию

1. Добавление идентификатора коллекции

2. Добавление первого документа

Путь к родительскому объекту документа

/profiles

Идентификатор документа

Сгенерировать

Поле	Тип	Значение
	string	

Отмена Сохранить

Рисунок 28 – Добавление первого документа

Далее нужно записать, какие поля будут у пользователя. «avatar» – это картинка в профиле пользователя, которая будет отображаться на сайте. «email» – это почта, которую будет вводить пользователь при регистрации. «fullName» – это имя пользователя, которое он будет вводить при регистрации. «uid» – это уникальный идентификатор пользователя. «last-changed» – это дата последнего входа пользователя в аккаунт. «state» – состояние аккаунта, оно

может быть онлайн или оффлайн. Все поля имеют тип «string», кроме поля «last-changed», оно имеет тип данных «timestamp».

Добавление полей в документ коллекции «profiles» представлено на рисунке 29.

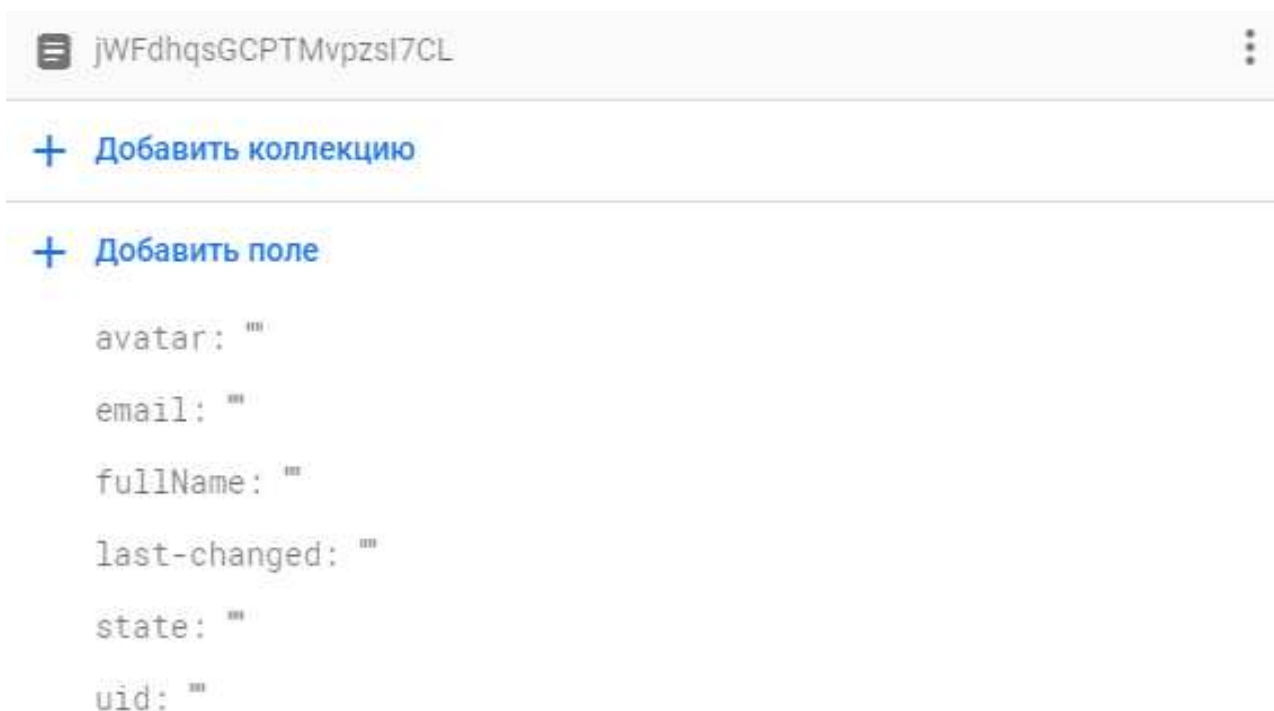


Рисунок 29 – Поля в коллекции «profiles»

Следующей коллекцией будут услуги, которые располагаются на сайте. Таким же способом создаётся новая коллекция, которая называется «services». Каждая услуга располагает пятью полями. «category» – это категория, в которую входит услуга. «description» – это описание услуги. «image» – это изображение услуги. «title» – это название услуги. «user» – это идентификатор пользователя, чья это услуга. Все поля имеют тип данных «string», кроме «user», он имеет тип данных «reference». Так как услуг насчитывается шесть штук, то документов, соответственно такое же количество.

Добавление коллекции «services» представлено на рисунке 30.

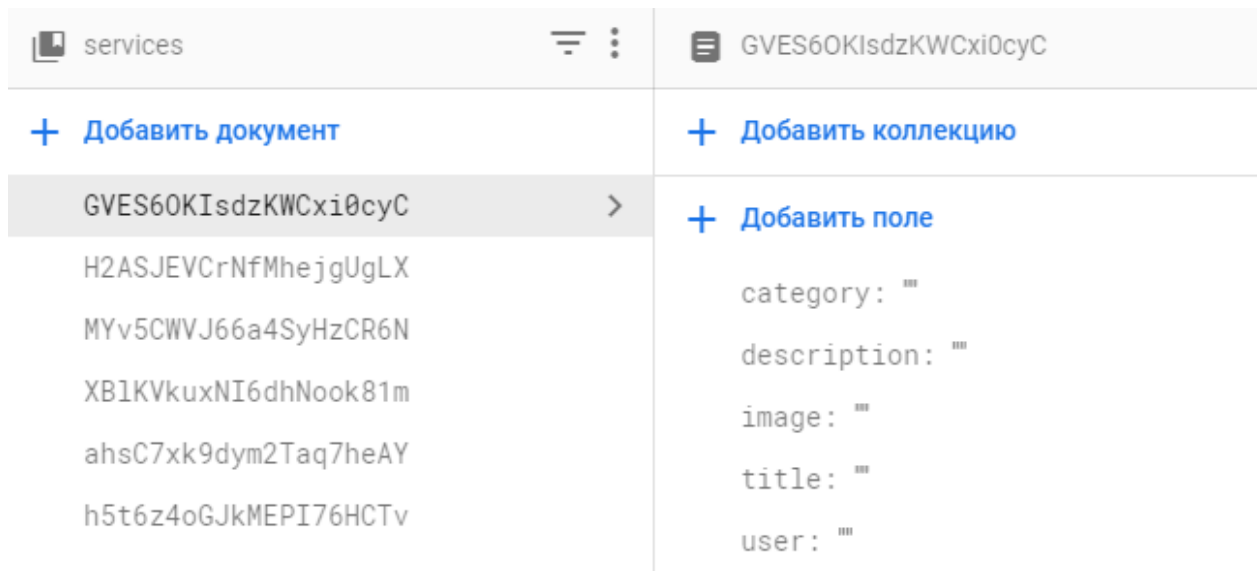


Рисунок 30 – Поля и документы в коллекции «services»

Третьей коллекцией предстают предложения, то есть «offers». Сюда входят документы всех заказов на услуги. «fromUser» – это ссылка на идентификатор пользователя, который взял услугу. «toUser» – это ссылка на идентификатор пользователя, который создал услугу. «service» – это ссылка на идентификатор услуги. «status» – это статус услуги после её заказа, который может быть в режиме ожидания, отклонения или принятия в зависимости от того, что сделает администратор этой услуги. «phone» – это телефон пользователя, который он вводит при заказе услуги. «note» – это примечание, где пользователь описывают свою проблему во время заказа услуги. «calendarDate» – это дата для оказания услуги пользователю, которую он вводит во время заказа. «calendarTime» – аналогично дате, но выдаёт время. Все поля имеют тип данных «string», кроме полей «fromUser», «toUser» и «service», они имеют тип данных «reference».

Добавление коллекции «offers» представлено на рисунке 31.

+ Добавить коллекцию

+ Добавить поле

```
calendarDate: ""
calendarTime: ""
fromUser: ""
note: ""
phone: ""
service: ""
status: ""
toUser: ""
```

Рисунок 31 – Поля в коллекции «offers»

В четвёртую коллекцию входят отзывы пользователей, она называется «rating». Сюда входит оценка и мнение пользователя о сайте. «fullName» – это имя пользователя, который написал отзыв. «rate» – это оценка, которую поставил пользователь. «wish» – это комментарий пользователя в отзыве.

Добавление коллекции «rating» представлено на рисунке 32.

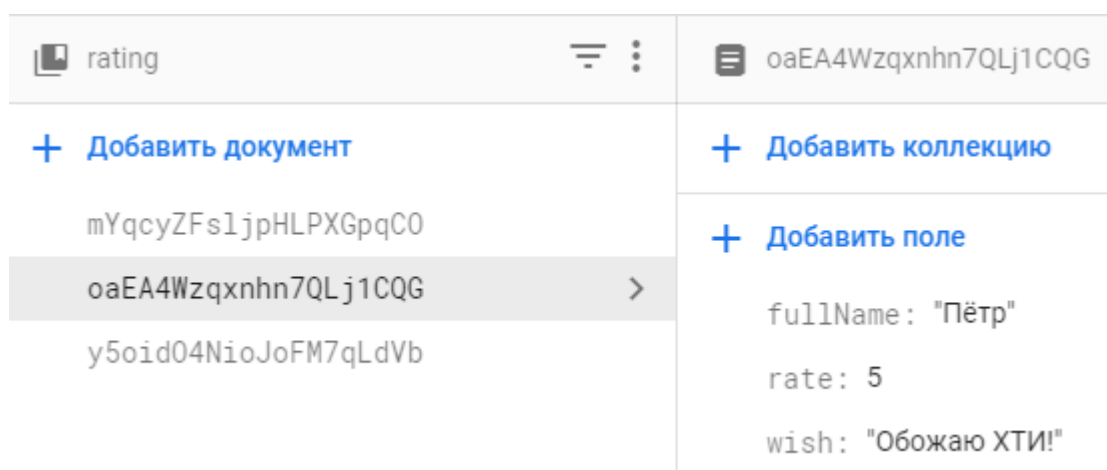


Рисунок 32 – Поля в коллекции «rating»

Последняя коллекция, это «collaborations» является коллаборацией между пользователем и администратором. Коллаборации позволяют общаться в чате, где могут находиться только заказчик услуги и администратор. Первое поле «allowedPeople» состоит из массива, тип данных которого «array», в которое входят 2 поля с типом данных «string» с идентификаторами заказчика и администратора. «createdAt» – это поле даты, когда была создана коллаборация, с типом данных «timestamp». «fromOffer» – это идентификатор заказа. «fromUser» – это идентификатор заказчика. «image» – это изображение услуги. Второй массив «joinedPeople» состоит из пользователей, которые находятся в чате. В нём находятся поля с типом данных «reference», это идентификаторы заказчика и администратора. «serviceId» – поле с типом данных «string», включающее идентификатор услуги. Дальше располагаются «status» и «title». Последнее поле это «toUser», то есть идентификатор администратора. Также в документе находится коллекция сообщений, это «messages».

Добавление коллекции «collaborations» представлено на рисунке 33.

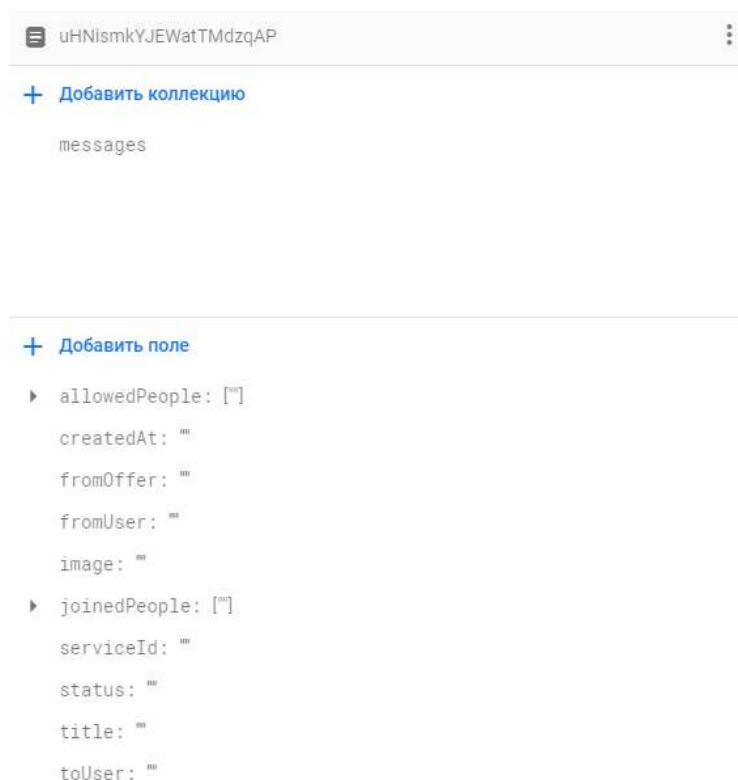


Рисунок 33 – Поля в коллекции «collaborations»

Коллекция «messages» сохраняет сообщения пользователей из чата. В коллекции генерируются документы, в которых находится поле «content» с типом данных «string», которое содержит в себе переданное сообщение. А также находится поле «user» с типом данных «map». Массив map позволяет изменять идентификатор массива, что делает обращение к данным из массива более понятным. В данный массив входят поля «avatar», «name» и «uid» с типами данных «string», которые хранят изображение, имя и идентификатор отправителя сообщения соответственно.

Коллекция «messages» представлена на рисунке 34.

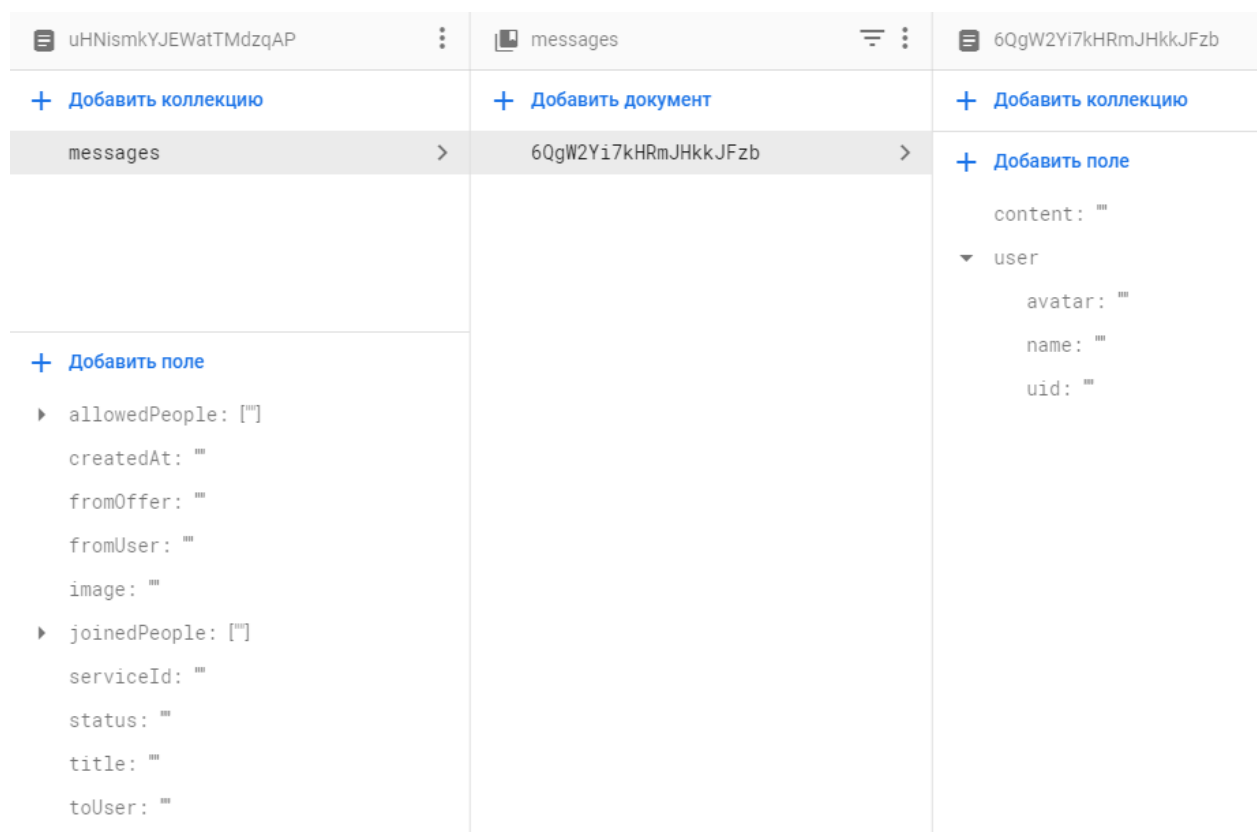


Рисунок 34 – Поля и документы в коллекции «messages»

Выше перечислена вся база данных со всеми, включающими в неё, данными. Так как база работает в режиме реального времени, то записи обновляются без перезагрузки страницы. Также можно вручную, через базу, добавлять услуги или другие какие-либо данные.

Cloud Firestore — это база типа document-model. Это означает, что данные хранятся в объектах, называемых документами, в виде ключ-значение, где значение может быть чем угодно, от строковой переменной до бинарного файла.

База данных представлена на рисунке 35.

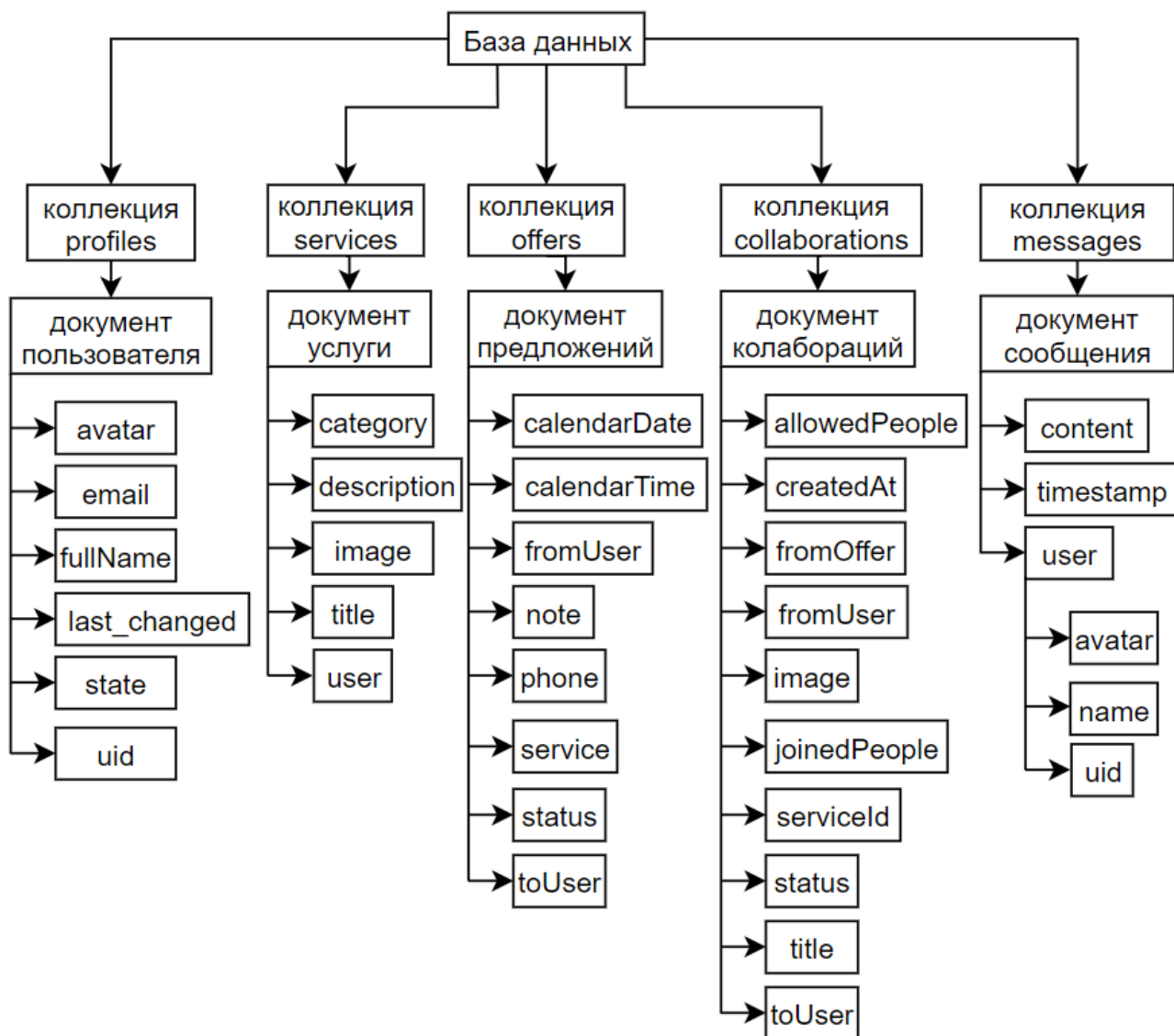


Рисунок 35 – База данных сайта

На сайте реализована функция смены изображения пользователя. Так как Cloud Firebase не может работать с изображениями, а только с файлами типа данных «string», то необходимо было создать Cloud Storage, которое обеспечивает хранение файлов.

Для этого на панели слева в разделе «Storage», необходимо нажать на кнопку «Начать».

Раздел «Storage» представлен на рисунке 36.

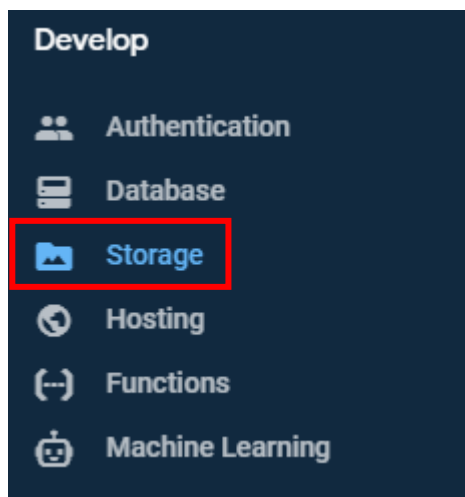


Рисунок 36 – Раздел «Storage»

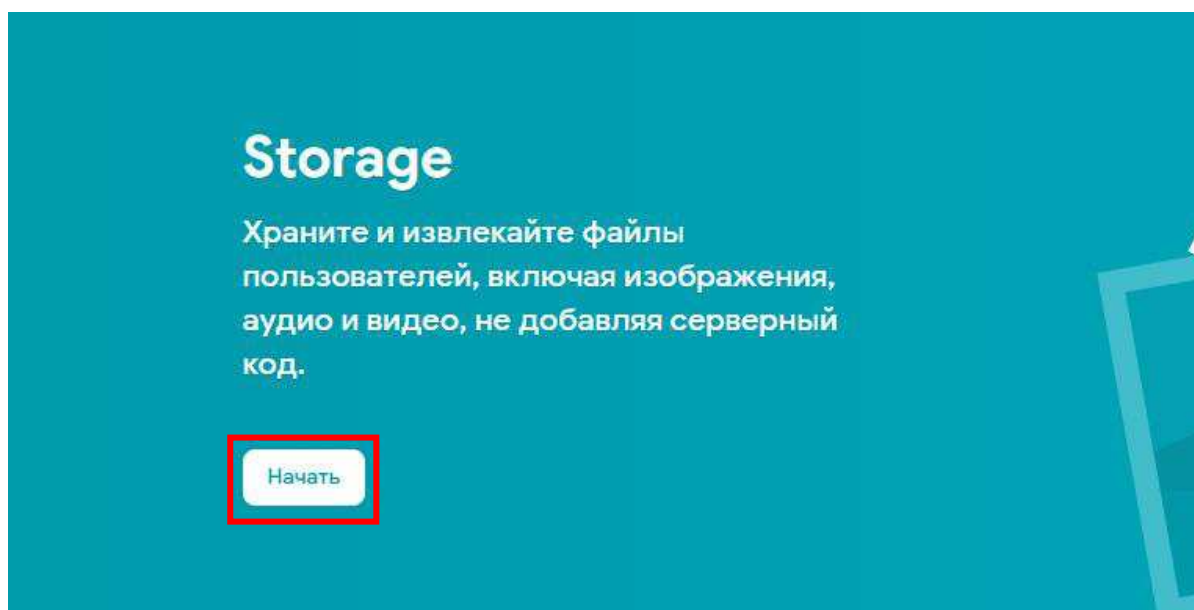


Рисунок 37 – Создание «Storage»

Затем последовательно нажать на «Далее» и «ОК». Начнётся создание хранилища в Firebase, в котором можно будет хранить фото, видео и другие файлы.

Настройка и создание «Storage» представлено на рисунках 38 и 39.

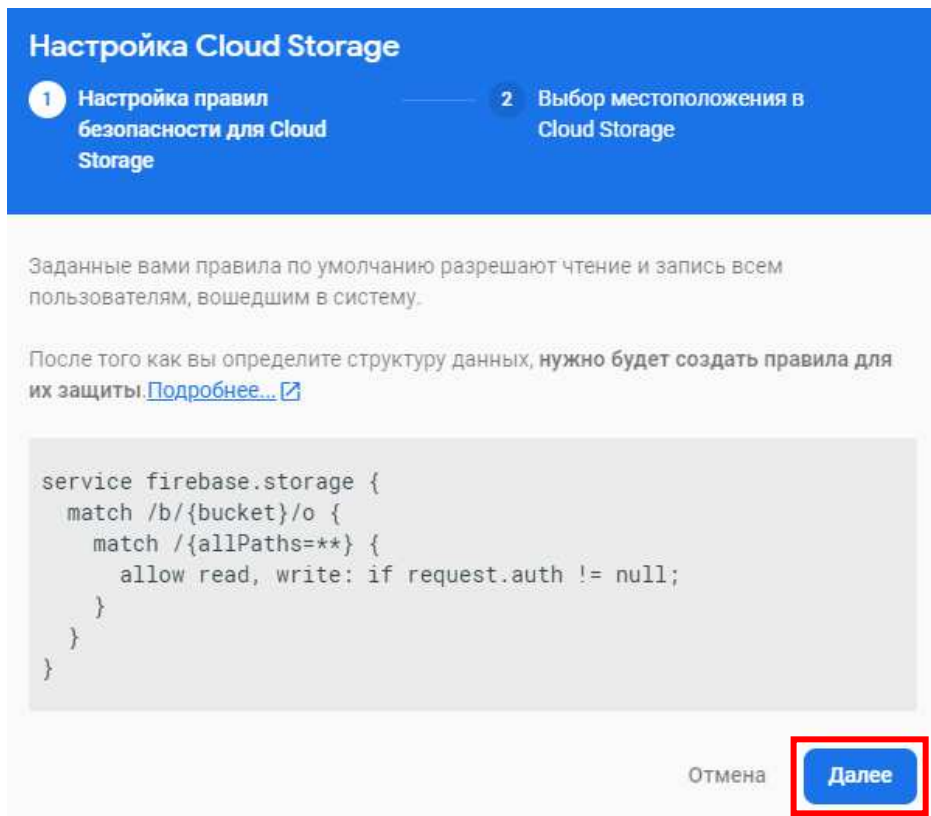


Рисунок 38 –Настройка «Storage»

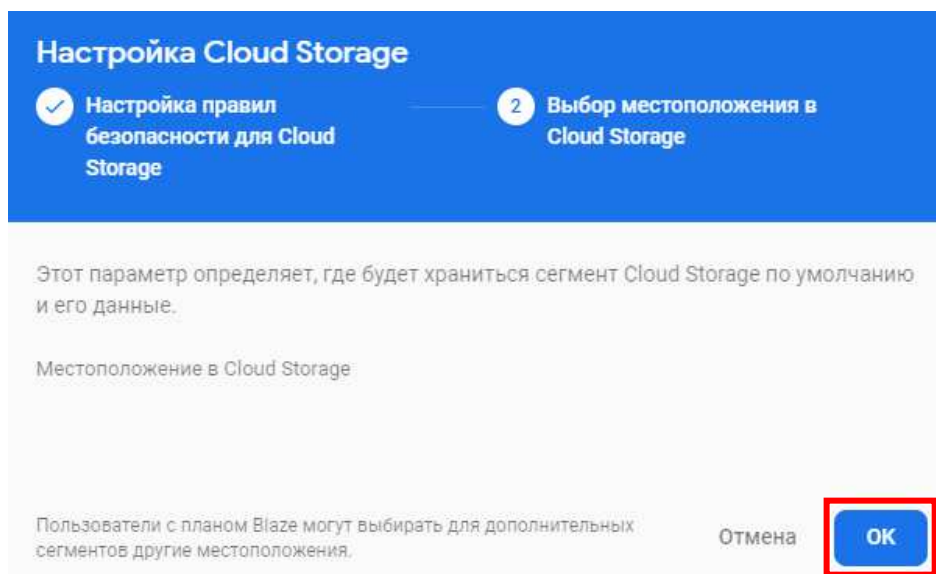


Рисунок 39 – Создание «Storage»

Сейчас в Storage Firebase пусто, но когда будет происходить загрузка изображения с сайта, сюда загрузится это изображение.

Storage Firebase представлен на рисунке 40.

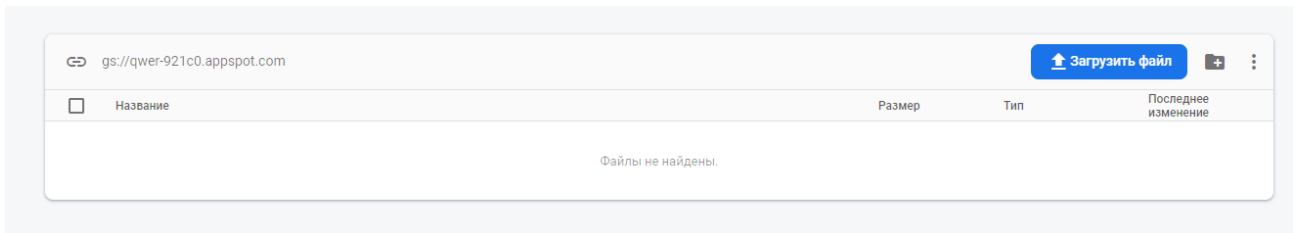


Рисунок 40 – Пустое хранилище Firestore

Сначала изображение загружается в Storage Firebase, у которого создаётся собственный токен доступа, то есть ссылка, по которой изображение будет доступно любому пользователю в интернете.

Токен представлен на рисунке 41.

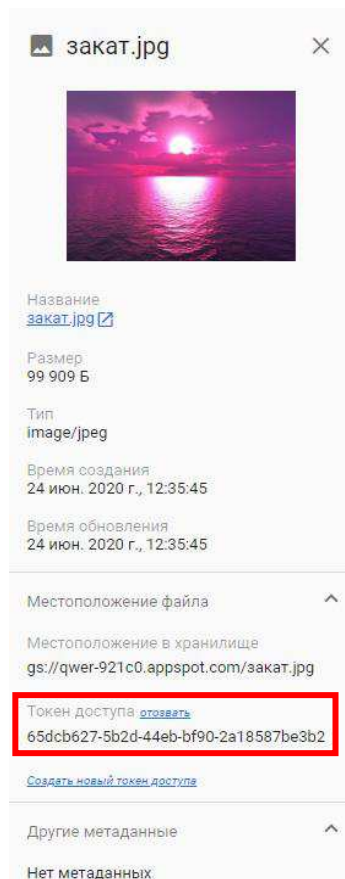


Рисунок 41 – Токен доступа изображения

Так как токен имеет тип данных «string», то он заменяет в Cloud Firestore прошлую ссылку изображения в коллекции «profile» в поле «avatar». Тем самым происходит смена изображения.

Остальная часть работы в разработке заключается в написании кода, где прописывается логика приложения, и связывании базы с кодом через консоль.

Когда проект сделан, его нужно опубликовать в сеть. Вначале нужно установить дополнительные модули в проект при помощи команды «firebase init function».

Чтобы установить firebase CLI, необходимо ввести команду «npm install -g firebase-tools».

Для того, чтобы у firebase был доступ к аккаунту, необходимо ввести команду «firebase login» и выбрать нужный аккаунт, нажав кнопку «Разрешить».

Запрос разрешения приложения Firebase CLI на доступ представлен на рисунке 42.

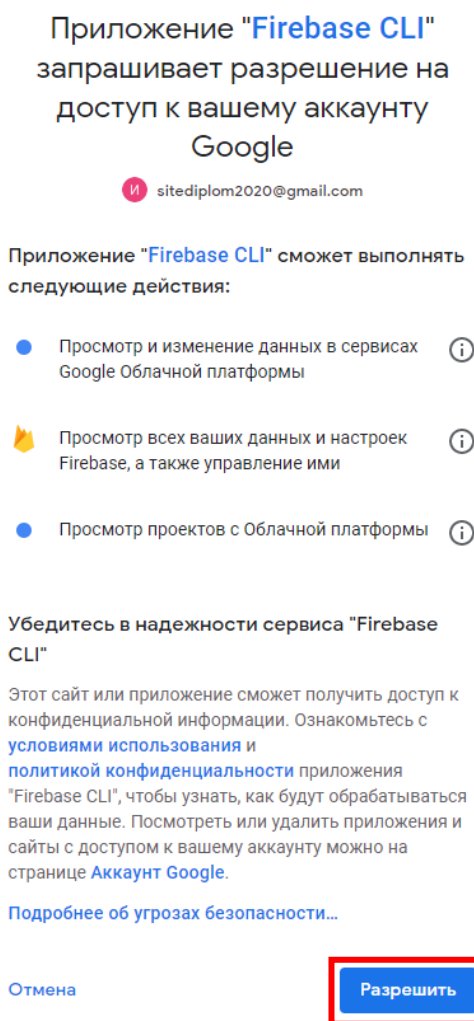


Рисунок 42 – Разрешение для приложения Firebase CLI

Можно просмотреть все созданные проекты в firebase при помощи команды «firebase projects:list». Там также указывается ID проекта.

Созданные проекты представлены на рисунке 43.

Project Display Name	Project ID	Project Number	Resource Location ID
khti	khti-6573d	649157652139	us-central
site-hti	site-hti (current)	454492947087	us-central

Рисунок 43 – Просмотр созданных проектов в firebase

Важным шагом является команда «npm init», которая создаёт файл package.json. В этот файл записываются названия пакетов с нужными версиями, необходимых для проекта. Это нужно для того, чтобы при команде «npm install» – установка пакетов, устанавливались пакеты только те, которые прописаны в файле package.json. После ввода команды «npm init», нужно попросят указать данные, такие как: «package name:», «version: (0.1.0)», «git repository:», «keywords:», «author:» и «license: (ISC)». В этом случае просто достаточно нажимать клавишу «Enter» в каждом пункте, чтобы согласиться, приняв изначальные условия. В конце спросят о правильности введённой информации, также нужно нажать на «Enter».

Далее необходимо написать команду в консоли «npm run build». Будет сгенерирована папка build, содержащая в себе статические файлы, которые будут использоваться на сервере. Данная команда при помощи Babel преобразует код React в код, который понимает браузер. При помощи оптимизации для лучшей производительности все файлы JavaScript объединяются в один файл. Этот файл будет минимизирован для сокращения времени загрузки. Babel – это транспайлер, переписывающий код современного стандарта Javascript на более поздний.

В папке проекта появится файл firebase.json, который представлен на рисунке 44.

📁 .firebase	11.06.2020 1:58
📁 build	11.06.2020 2:02
📁 node_modules	11.06.2020 1:30
📁 public	10.06.2020 1:25
📁 src	11.06.2020 1:35
📄 .firebaseconfig	11.06.2020 1:58
📄 .gitignore	26.10.1985 16:15
📄 firebase.json	11.06.2020 2:06
📄 package.json	11.06.2020 2:01
📄 package-lock.json	11.06.2020 1:30
📄 README.md	26.10.1985 16:15

Рисунок 44 – Файл firebase.json

Здесь важна строка «public». Она представлена на рисунке 45

```

{
  "hosting": {
    "public": "public",
    "ignore": [
      "firebase.json",
      "**/*.*",
      "**/node_modules/**"
    ],
    "rewrites": [
      {
        "source": "**",
        "destination": "/index.html"
      }
    ]
  }
}

```

Рисунок 45 – Строка «public»

Её нужно изменить на «build», так как преобразованный готовый код лежит в папке «build». Изменения представлены на рисунке 46.

```
{
  "hosting": {
    "public": "build",
    "ignore": [
      "firebase.json",
      "**/.*",
      "**/node_modules/**"
    ],
    "rewrites": [
      {
        "source": "**",
        "destination": "/index.html"
      }
    ]
  }
}
```

Рисунок 46 – Строка «build»

Чтобы загрузить проект на хостинг, необходимо ввести команду «firebase init» и выбрать четвёртую строку «Hosting: Configure and deploy Firebase Hosting sites».

Команда представлена на рисунке 47.

```
C:\Users\2018\Desktop\qweqwe\qweqwe\src>firebase init

#####
##
#####
##
##

You're about to initialize a Firebase project in this directory:

C:\Users\2018\Desktop\qweqwe\qweqwe\src

> Are you ready to proceed? Yes
> Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices.
  ( ) Database: Deploy Firebase Realtime Database Rules
  ( ) Firestore: Deploy rules and create indexes for Firestore
  ( ) Functions: Configure and deploy Cloud Functions
  > (*) Hosting: Configure and deploy Firebase Hosting sites
  ( ) Storage: Deploy Cloud Storage security rules
  ( ) Emulators: Set up local emulators for Firebase features
```

Рисунок 47 – Выбор хостинга для загрузки проекта

Далее выбрать «Use an existing project», который означает выбор из существующих проектов. И выбрать свой проект.

Выбор проекта представлен на рисунке 48.

```
? Please select an option:  
> Use an existing project  
Create a new project  
Add Firebase to an existing Google Cloud Platform project  
Don't set up a default project
```

Рисунок 48 – Выбор из существующих проектов

Далее нужно указать директорию, из которой будет идти инициализация.

Выбор директории представлено на рисунке 49.

```
? What do you want to use as your public directory? (public)
```

Рисунок 49 – Выбор директории

Написать букву «у», что значит «yes» и нажать «Enter», чтобы приложение SPA записалось в файл «index.html».

Согласие на запись файла представлено на рисунке 50.

```
? Configure as a single-page app (rewrite all urls to /index.html)? (y/N)
```

Рисунок 50 – Соглашение с перезаписью файла

После этого можно написать команду «firebase deploy», которая загрузит файлы с папки «build» в firebase. В аккаунте firebase на панели слева необходимо нажать на «Hosting».

Раздел хостинга представлен на рисунке 51.

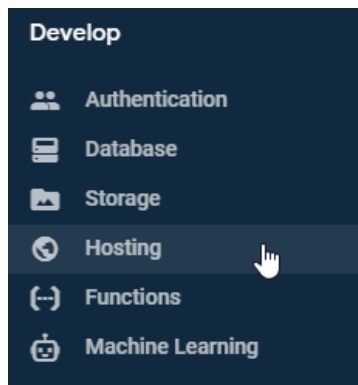


Рисунок 51 – Раздел «Hosting»

Первая панель, которая предстала перед глазами, это домены сайта. То есть на готовый сайт можно перейти по ссылкам «<https://site-hti.web.app/>» и «<https://site-hti.firebaseio.com/>».

Домены сайта представлены на рисунке 52.

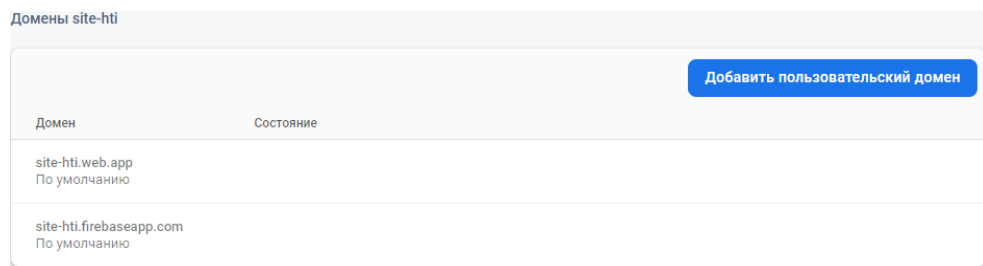


Рисунок 52 – Домены сайта

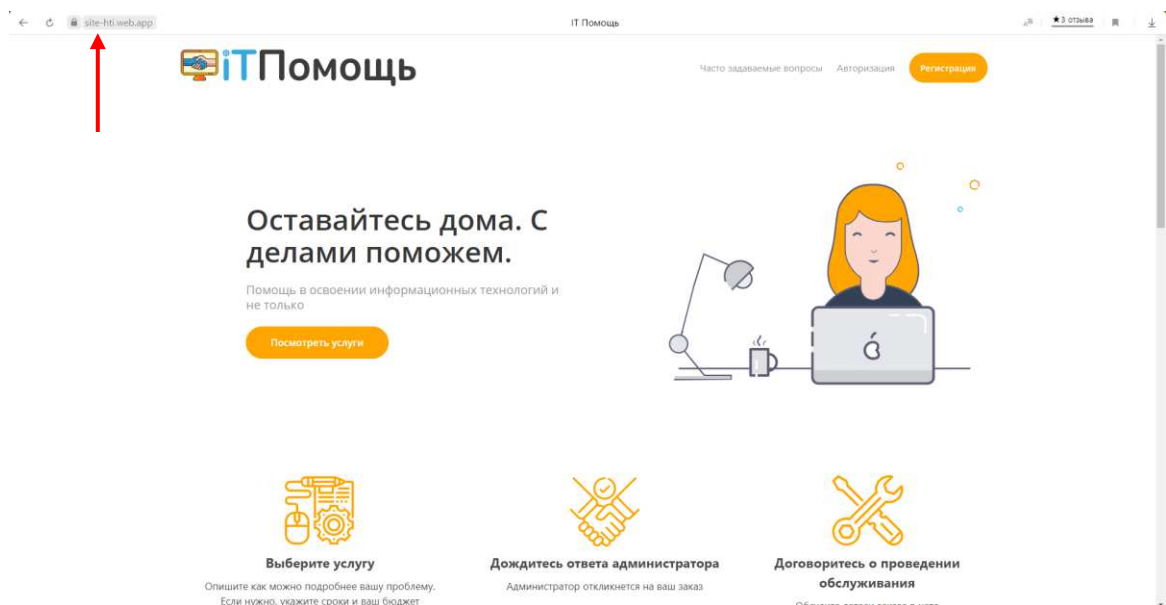
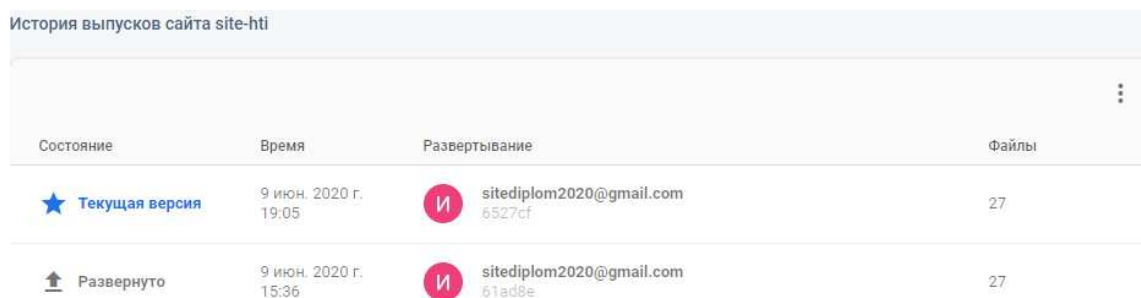


Рисунок 53 – Готовый сайт с доменом

Вторая панель это история развёртывания файлов на хостинг. Здесь можно увидеть, сколько файлов загрузилось на хостинг, дату развёртывания, а также есть возможность вернуться к прошлым развёртываниям, чтобы увидеть предыдущую версию сайта.

Версии проектов Firebase представлены на рисунке 54.



История выпусков сайта site-hti



Состояние	Время	Развертывание	Файлы
★ Текущая версия	9 июн. 2020 г. 19:05	 sitediplom2020@gmail.com 6527cf	27
↑ Развернуто	9 июн. 2020 г. 15:36	 sitediplom2020@gmail.com 61ad8e	27

Рисунок 54 – Версии проектов в Firebase

Код, прописывающий логику приложения, лежит в папке «src». Все файлы сайта отсортированы и лежат в разных папках. Файлы, которые отвечают за работу сайта, находятся в корне папки «src». Так как react это javascript-библиотека, то все файлы имеют расширение «.js», кроме файлов SCSS, которые добавляют стиль сайту.

Файлы в папке «src» представлены на рисунке 55.

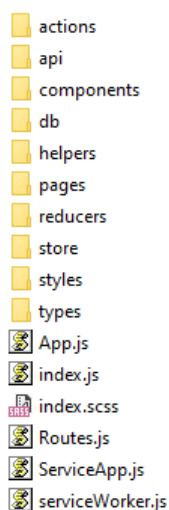


Рисунок 55 – Файлы, отвечающие за работу сайта

Сайт состоит из страниц, по которым можно переходить, поэтому в папке «pages» располагаются файлы, отвечающие за эти страницы, а также три папки.

Файлы в папке «pages» представлены на рисунке 56.

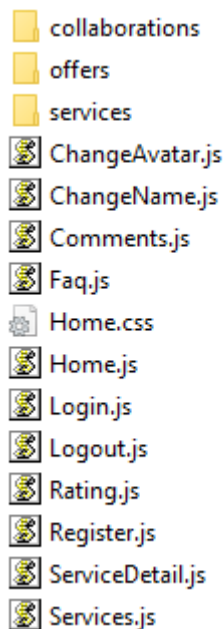


Рисунок 56 – Файлы в папке «pages»

В папке «collaboration» находятся файлы «CollaborationDetail.js» и «ReceivedCollaborations.js», которые отвечают за чат после коллаборации и список коллабораций соответственно.

Файлы в папке «collaboration» представлены на рисунке 57.

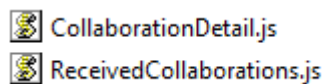


Рисунок 57 – Файлы в папке «collaboration»

Была создана папка «offers», содержащая файлы «ReceivedOffers.js» и «SentOffers.js», которые отвечают за отправленные предложения и принятые предложения соответственно.

Файлы в папке «offers» представлены на рисунке 58.

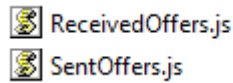


Рисунок 58 – Файлы в папке «offers»

Папка «services» содержит файлы «ServiceCreate.js» и «UserServices.js», в которых прописывается создание услуги и все услуги пользователя соответственно.

Файлы в папке «services» представлены на рисунке 59.

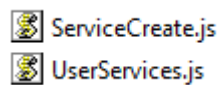


Рисунок 59 – Файлы в папке «services»

Остальные файлы, находящиеся в папке «pages», отвечают за создание страниц: часто задаваемых вопросов, оценок услуг и просмотр этих оценок, учетной записи, домашней страницы, страниц с авторизацией, регистрацией и выхода из аккаунта, а также просмотр конкретной услуги.

Папка «components» содержит следующие файлы, которые отвечают за главную страницу, навигационную панель и значок загрузки.

Файлы в папке «components» представлены на рисунке 60.

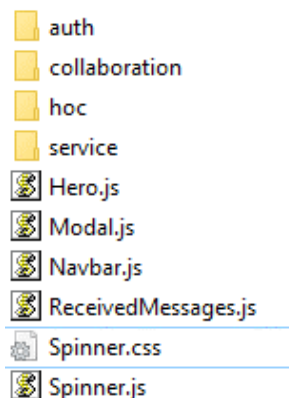


Рисунок 60 – Файлы в папке «components»

В отдельной папке «auth» была создана форма с регистрацией. Файл, которой носит имя «RegisterForm.js», проверяет на валидацию имя пользователя, почту и пароль.

Файл в папке «auth» представлен на рисунке 61.

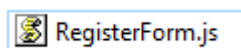


Рисунок 61 – Файл в папке «auth»

В папке «collaboration» содержатся файлы «ChatMessages.js» и «JoinedPeople.js», в которых прописываются чат и профили пользователей в чате.

Файлы в папке «collaboration» представлены на рисунке 62.

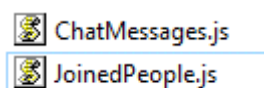


Рисунок 62 – Файлы в папке «collaboration»

В папке «hoc» содержатся файлы «onlyGuest.js» и «withAuthorization.js», в которых прописываются настройка для зарегистрированных пользователей и тех, кто зашёл в качестве гостя.

Файлы в папке «hoc» представлены на рисунке 63.

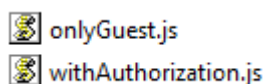


Рисунок 63 – Файлы в папке «hoc»

В папке «service» содержатся файлы «OfferModal.js» и «ServiceItem.js», в которых прописываются окна услуг и окно заказа услуги.

Файлы в папке «service» представлены на рисунке 64.

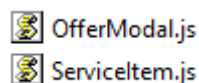


Рисунок 64 – Файлы в папке «service»

В папке «helpers» находятся файлы «offers.js» и «validators.js». В файле «validators.js», прописаны настройки для полей на правильность их заполнения при регистрации или авторизации. В файле «offers.js» содержится сообщение в качестве уведомления, когда пришла коллаборация.

Файлы в папке «helpers» представлены на рисунке 65.

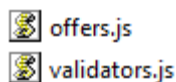


Рисунок 65 – Файлы в папке «helpers»

В папке «store» содержится файл «index.js», который нужен для асинхронных запросов к внешней API и отладки в коде.

Файл в папке «store» представлены на рисунке 66.

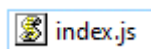


Рисунок 66 – Файл в папке «store»

В папке «db», как уже упоминалось ранее, содержится файл «index.js», в котором происходит связь проекта с firebase.

Файл в папке «db» представлен на рисунке 67.

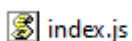


Рисунок 67 – Файл в папке «db»

В папке «styles» содержатся три файла со стилями «auth.scss», «collaborations.scss» и «services.scss», в которых описывается внешний вид документа. В файлах типа «scss» в отличие от файлов типа «css», присутствуют такие инструменты как переменные, наследование, логические операции, математические функции, разделение на модули. Данные файлы отвечают за внешний вид страниц авторизации, чата и услуг.

Файлы в папке «styles» представлены на рисунке 68.

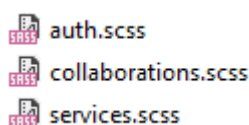


Рисунок 68 – Файлы в папке «styles»

Для того, чтобы взаимодействовать с api firebase, была создана папка «api», в которой находятся шесть файлов, отвечающие за запись в базу данных информации при регистрации, авторизации, создании услуг и статус пользователя в сети.

Файлы в папке «api» представлены на рисунке 69.



Рисунок 69 – Файлы в папке «api»

В файл папки «actions» импортируется файл из папки «types». В файлах прописываются запросы, в том числе и асинхронные. Это необходимо для того, чтобы сайт работал с данными через базу данных без перезагрузки страницы.

Файл в папке «api» представлен на рисунке 70.

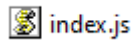


Рисунок 70 – Файл в папке «types»

Файлы в папке «actions» представлены на рисунке 71.

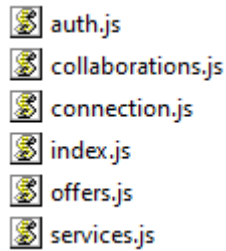


Рисунок 71 – Файлы в папке «actions»

Во время регистрации и авторизации. выполняется определённый код, данный код показан на скриншоте ниже.

Регистрация и авторизация представлены на рисунках 72 и 73.

```
export const register = async ({email, password, fullName, avatar}) => {
  try {
    const res = await firebase.auth().createUserWithEmailAndPassword(
      email, password)
    const { user } = res
    const userProfile = { uid: user.uid, fullName, email, avatar,
      services: [], description: ''}
    await createUserProfile(userProfile)
    return userProfile
  } catch(error) {
    return Promise.reject(error.message)
  }
}
```

Рисунок 72 – Регистрация

```
export const login = ({email, password}) =>
  firebase.auth().signInWithEmailAndPassword(email, password)
    .catch(error => Promise.reject(error.message))
```

Рисунок 73 – Авторизация

2.7 Выводы по разделу «Проектирование ИС»

Была построена модель ИС в нотации IDEF0 и её декомпозиция. Описаны входы, выходы, механизмы и управление ИС. В развёрнутой диаграмме подробно описаны механизмы и управление. А также по шагам расписана разработка сайта. Была построена модель ИС в нотации IDEF3, в которой по пунктам описано взаимодействие пользователя с сайтом. Была построена модель ИС в нотации DFD. В ней описаны все этапы по обслуживанию клиента. После проектирования моделей ИС в таких нотациях как IDEF0, IDEF3 и DFD, строится определённый план, который подробнее описывает будущую эксплуатацию проекта. Разработанный сайт создан через SPA – одностраничное приложение. Установка приложения выполняется при помощи предварительной загрузки и настройки Node.js. Создание базы данных происходит путём регистрации на Google аккаунте с последующим созданием приложения на платформе Firebase. Далее в Cloud Firestore идёт создание коллекций с документами. Затем создаётся Storage Firestore для хранения изображений.

Так как работа выполнялась совместно с Волковым Андреем Валерьевичем из группы 56-1, то данный раздел представлен не полностью. Описание эксплуатации сайта описано в работе партнёра.

3 Экономическое обоснование реализации проекта

3.1 Расчет затрат реализации проекта создания ИС

Для начала требуется подготовиться к проекту. В этот этап входит сбор всей необходимой информации для того, чтобы понять, зачем нужен проект и как он будет разрабатываться. Далее идёт разработка, это самый большой этап проекта. В нём идёт практическая работа над ним и его параллельное документирование. Третьим этапом располагается тестирование. Здесь

происходит тест завершённой работы, выявление ошибок в процессе и исправление этих ошибок. Внедрение в эксплуатацию это последний пункт в проекте. В нём нужно спроектированную работу ввести в полное рабочее состояние. То есть проект становится полностью функционирующим. Здесь понадобится совершить все условия для его правомерного использования. А также, возможно, будут исправляться некоторые ошибки.

В таблице 3 показаны плановые сроки начала и окончания работ.

Таблица 3 – Плановые сроки начала и окончания работ по внедрению проекта

Подготовка к проекту	Разработка	Тестирование	Внедрение в эксплуатацию
1 неделя	2 недели	3 дня	2 дня

Для расчёта затрат реализации проекта выбрана методика ТСО. Она рассчитывается по формуле 1.

$$ТСО = DE + IC_1 + IC_2, \quad (1)$$

где DE – прямые затраты;

IC₁ – косвенные расходы первой группы;

IC₂ – косвенные расходы второй группы.

Работа производилась двумя разработчиками, соответственно, капитальные затраты у каждого разные. Каждый осуществлял свои собственные затраты лично, которые впоследствии складывались. Работа происходила совместно с Волковым Андреем Валерьевичем студентом группы 56-1.

3.2 Капитальные затраты

Капитальные затраты на информационную систему носят разовый характер. Свою стоимость они переносят на продукцию по частям за счёт амортизационных отчислений.

Вычисление капитальных затрат по формуле 5:

$$K=K_{\text{пр}}+K_{\text{тс}}+K_{\text{лс}}+K_{\text{по}}+K_{\text{ио}}+K_{\text{об}}+K_{\text{оэ}}, \quad (5)$$

где $K_{\text{пр}}$ – затраты на проектирование информационной системы;

$K_{\text{тс}}$ – затраты на технические средства управления;

$K_{\text{лс}}$ – затраты на создание линий связи локальных сетей;

$K_{\text{пс}}$ – затраты на программные средства;

$K_{\text{ио}}$ – затраты на формирование информационной базы;

$K_{\text{об}}$ – затраты на обучение персонала;

$K_{\text{оэ}}$ – затраты на опытную эксплуатацию.

Формула вычисления затрат на проектирование информационной системы:

$$K_{\text{пр}}=K_{\text{зп}}+K_{\text{ипс}}+K_{\text{свт}}+K_{\text{проч}}, \quad (6)$$

где $K_{\text{зп}}$ – затраты на заработную плату проектировщиков;

$K_{\text{ипс}}$ – затраты на инструментальные программные средства для проектирования;

$K_{\text{свт}}$ – затраты на средства вычислительной техники для проектирования;

$K_{\text{проч}}$ – прочие затраты на проектирование.

Для того, чтобы узнать заработную плату разработчиков нужно сначала высчитать сумму заработной платы с районным и северным коэффициентами. Районный и северный коэффициенты составляют 30% от заработной платы. Доходы от коэффициентов складывают с зарплатой. В итоге получается общая затрата на зарплату разработчику. От всей заработной платы разработчика отчисляется НДФЛ в размере 13%.

Доходы и расходы представлены в таблице 4.

Таблица 4 – Затрата на зарплату разработчика

Доходы		Расходы	
Заработная плата	8 000 рублей		
Районный коэффициент	30% (+2400руб.)		
Северный коэффициент	30% (+2400 руб.)		
Сумма	12 800 руб.	НДФЛ	13% (1664руб.)
Всего на руки	11 136 руб.		

В фонд оплаты труда входят обязательные отчисления в фонды пенсионный, медицинский и социального обеспечения. Пенсионный фонд 22%, фонд медицинского страхования 5,1%, фонд социального обеспечения 2,9%, фонд социального страхования от несчастных случаев и профессиональных заболеваний 0,2%. ФОТ = затрата на зарплату разработчика + 30,2% (от его зарплаты).

$$K_{зп} = 12800 * 1,302 = 16\ 666 \text{ рублей}$$

Для разработки проекта необходимо специальное оборудование. Нужно посчитать стоимость на все элементы оборудования, чтобы узнать общие затраты.

В таблице 5 показаны характеристики компьютера, который использовался для работы с проектом.

Таблица 5 – Технические характеристики оборудования и его стоимость для разработки проекта

Комплектующие	Название элементов оборудования	Кол-во единиц	Стоимость единицы, руб.	Стоимость всего, руб.
Процессор	AMD Ryzen 5 2600 BOX	1	10 699 руб.	10 699 руб.

Продолжение таблицы 5

Видеокарта	MSI GeForce GTX 1660 ARMOR OC	1	19000 руб.	19000 руб.
Материнская плата	ASRock B450 STEEL LEGEND	1	7 999 руб.	7 999 руб.
Блок питания	HKC-ATX700-6070	1	4000 руб.	4000 руб.
Оперативная память	Kingston HyperX Predator [HX430C15PB3K2/16]	2	6 999 руб.	6 999 руб.
Хранение данных HDD	WD Blue [WD20EZRZ]	1	4 099	4 099
Хранение данных SSD	WD Green [WDS240G2G0A]	1	2 799	2 799
Корпус	AirTone K5-9388	1	3000 руб.	3000 руб.
Охлаждение процессора	DEEPCOOL GAMMAXX 400	1	1 999 руб.	1 999 руб.
Клавиатура	BenQ 6511-VA	1	1000 руб.	1000 руб.
Мышь	A4Tech Bloody A6/A60 Blazing черный	1	1 599 руб.	1 599 руб.
Монитор	Samsung S19A100	1	6000 руб.	6000 руб.
Всего:	69193 руб.			

Для одного разработчика требуется 1 компьютер. Значит 69193 руб. это минимальная необходимая сумма для технического оборудования.

По статистике, средний срок службы компьютерных комплектующих составляет:

- процессор – около 10 лет;
- материнская плата – около 7 лет;
- видеокарта – от 3 до 8 лет;
- модуль оперативной памяти (ОЗУ) – около 10 лет;
- жёсткий диск (HDD) – около 10 лет;
- твердотельный накопитель (SSD) – около 3 лет;
- блок питания – от 5 до 8 лет;
- корпус – до 30 лет;

- ЖК-монитор – около 8 лет;
- клавиатура – от 10 лет;
- мышь – от 3 до 6 лет.

У каждого оборудования есть своя стоимость. Для расчёта их затрат нужно узнать амортизацию по формуле 2.

$$A_{\text{год}} = C_{\text{б}} * N_{\text{ам}}, \quad (2)$$

где $A_{\text{год}}$ – амортизация за год использования;

$C_{\text{б}}$ – балансовая стоимость;

$N_{\text{ам}}$ – норма амортизации.

$$A_{\text{пр}} = \frac{A_{\text{год}}}{K_{\text{рдг}}} * K_{\text{дэ}}, \quad (3)$$

где $A_{\text{пр}}$ – проектная амортизация;

$K_{\text{рдг}}$ – количество рабочих дней в 2020 году;

$K_{\text{дэ}}$ – количество дней эксплуатации.

$$N_{\text{ам}} = \frac{100\%}{T_{\text{эк}}}, \quad (4)$$

где $T_{\text{эк}}$ – срок эксплуатации компьютера.

Так как срок эксплуатации некоторых элементов может быть меньше 5 лет, нужно предусмотреть возможность их замены. Средний срок эксплуатации компьютеров составляет 5 лет. Для поддержания комплектующих в рабочем состоянии необходимы дополнительные затраты в размере 3000 рублей.

$$T_{\text{эк}} = 5$$

$$N_{\text{ам}} = \frac{100\%}{5} = 0,2$$

$$A_{\text{год}} = 69193 * 0,2 + 3000 = 16839$$

$$K_{\text{рдг}} = 248$$

$$K_{\text{дэ}} = 14$$

$$A_{\text{пр}} = \frac{16839}{248} * 14 = 951 \text{ руб.}$$

$$K_{\text{свт}} = 951 \text{ руб.}$$

Расчёт стоимости программного обеспечения представлен в таблице 6.

Таблица 6 – Расчёт стоимости программного обеспечения для создания информационной системы

Наименование программного обеспечения	Стоимость, руб.	Срок использования	Количество единиц
Firestore	Бесплатно	Бессрочно	1
Microsoft VS Code	Бесплатно	Бессрочно	1
Node.js	Бесплатно	Бессрочно	1
Microsoft Windows 10 Home	8350 руб.	Бессрочно	1
Google Chrome	Бесплатно	Бессрочно	1
Всего:	8350 руб.		

Microsoft Windows 10 Home (x64) стоит 8350 рублей и обновляется примерно раз в 3 года. Стоимость операционной системы делим на 8, так как время разработки на этой ОС выходит 4,5 месяца.

$$K_{\text{инс}} = 8350 / 4,5 = 1856$$

Затраты на инструментальные программы средства для проектирования составляют 1856 руб.

Технические характеристики аппаратного обеспечения во время эксплуатации остаются такими же, так как для того, чтобы пользоваться сайтом не требуются мощные технические средства.

Платформа Firebase является относительно бесплатной. Но бесплатного тарифного плана на небольшой проект хватит с запасом. В противном случае существует возможность платить за фактическое пользование.

В таблице 7 показано программное обеспечение, которое требуется, чтобы пользоваться сайтом.

Таблица 7 – Требуемое программное обеспечение во время эксплуатации

Наименование программного обеспечения	Стоимость, руб.	Срок использования	Количество единиц
Google Chrome, Internet Explorer, Opera, Mozilla Firefox, Яндекс.Браузер и др.	Бесплатно	Бессрочно	1
Всего:	0 руб.		

Оно составляет 0 рублей.

Затраты на технические средства управления включают амортизацию компьютеров нужных при создании сайта и его эксплуатации составляют 951 руб.

Прочие затраты равны 3% от всей суммы затрат.

Затраты на технические средства управления 0 рублей, так как эти затраты включены в эксплуатационные затраты.

Стоимость подключения интернета 500 рублей, хостинг предоставляется бесплатно.

$K_{лс}=500$ руб.

Чтобы следить за стабильной работой сайта, необходим браузер, который доступен бесплатно.

Обучение персонала не требуется. Созданием сайта будут заниматься разработчики, имеющие необходимый опыт в создании веб-страниц.

Для системы не нужна опытная эксплуатация, так как все недоработки будут исправляться по мере разработки.

Капитальные затраты представлены в таблицах 8 и 9.

Так как работа выполнялась двумя разработчиками, то капитальные затраты рассчитывались совместно.

Таблица 8 – Капитальные затраты

Затраты	Состав затрат	Планируемая сумма
К _{пр} (затраты на проектирование информационной системы)	Затраты на заработную плату разработчика	12800+7733=20533 руб.
	Затраты на инструментальные программы средства для проектирования	1856 руб.
	Затраты на средства вычислительной техники для проектирования	951 руб.
	Прочие затраты на разработку	701 руб.
К _{тс} (Затраты на технические средства управления)		0
К _{лс} (Затраты на создание линий связи локальных сетей)		500
К _{по} (Затраты на программные средства)		0
К _{но} (Затраты на формирование информационной базы)		0
К _{об} (Затраты на обучение персонала)		0
К _{оэ} (Затраты на опытную эксплуатацию)		0

Таблица 9 – Капитальные затраты

Капитальные затраты	Сумма руб.
К _{пр}	24041
К _{тс}	0
К _{лс}	500
К _{по}	0
К _{но}	0
К _{об}	0
К _{оэ}	0

Капитальные затраты составляют:

$$K = 24041 + 500 = 24541 \text{ руб.}$$

На рисунке 74 показаны капитальные затраты в виде круговой диаграммы.

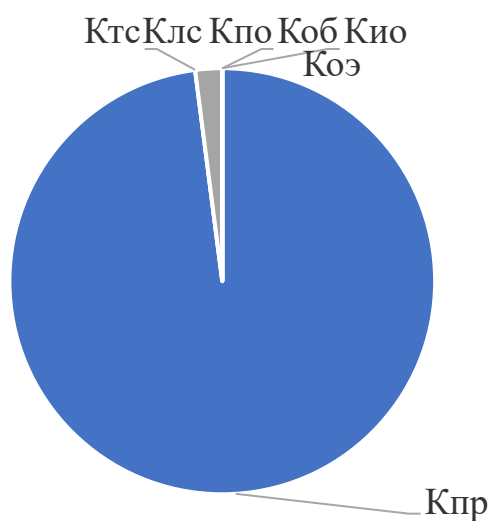


Рисунок 74 – Капитальные затраты

Так как работа выполнялась совместно, то капитальные затраты следует сложить. У второго разработчика капитальные затраты составили:

$$K = 56710 + 500 + 2813 = 60023 \text{ руб.}$$

Общие капитальные затраты составили:

$24541+60023= 84564$ руб.

3.3 Расчёт экономической эффективности проекта

Поскольку проект подразумевает оказание в основном безвозмездных услуг и его реализация не имеет прямых финансовых выгод, для определения экономической эффективности проекта решено сравнить предложенное проектное решение с вариантом наем оператора, собирающего заявки на обслуживание населения.

Необходимо провести расчет экономической эффективности с учётом коэффициента комплексного качества.

Для обобщающей характеристики эксплуатационного-технического уровня системы (ЭТУ) используем адаптивно-мультипликативный показатель «значимость технического решения» (ЗТР) для решения технической задачи. В общем виде рассчитывается по формуле:

$$\text{ЗТР} = K_a + K_{\text{п}} + K_c + K_m + K_o + K_{\text{ш}}, \quad (9)$$

где K_a – коэффициент актуальности;

$K_{\text{п}}$ – коэффициент соответствия программам важнейших работ научно-технического прогресса;

K_c – коэффициент сложности;

K_m – коэффициент места использования;

K_o – коэффициент объёма использования;

$K_{\text{ш}}$ – коэффициент широты охвата охраняемыми мероприятиями.

В таблице 10 показаны коэффициенты и ЗТР базового и разрабатываемого вариантов проекта.

Таблица 10 – Коэффициенты и ЗТР базового и разрабатываемого вариантов проекта

Коэффициенты	Базовый вариант	Разрабатываемый вариант
K_a	1	2
K_{Π}	1	1
K_c	1	3
K_M	1	1
K_o	1	1
$K_{\text{ш}}$	1	1
ЗТР	2	9

$$K_{\text{эту}} = \frac{\text{ЗТР}_{\text{пр}}}{\text{ЗТР}_{\text{баз}}}, \quad (10)$$

где $\text{ЗТР}_{\text{пр}}$ – значимость технического решения для проекта;

$\text{ЗТР}_{\text{баз}}$ – значимость технического решения для базового варианта.

$$K_{\text{эту}} = \frac{9}{2} = 4,5$$

Если $K_{\text{эту}} > 1$, то разработка проекта является оправданной с технической точки зрения.

Вычислим комплексный показатель качества проекта по формуле 11:

$$I_{\text{эту}} = \sum b_i * X_i, \quad (11)$$

где b_i – коэффициент весомости i -го показателя;

X_i – относительный показатель качества, устанавливаемый экспертным путём по выбранной шкале оценивания.

В таблице 11 показаны оценки показателя качества разрабатываемого и базового проектов.

Таблица 11 – Расчёт показателя качества

Показатель качества	Весовой коэффициент, b_i	Оценка X_i	
		Разрабатываемый проект	Базовый проект
Удобство работы (пользовательский)	0,2	5	3
Надёжность (защита данных)	0,2	5	3
Функциональные возможности	0,2	4	3
Временная экономичность	0,3	5	4
Время обучения персонала	0,1	1	2
Комплексный показатель качества $I_{эту}$	4,4	3,2	

$$I_{этупр} = 0,2*5+0,2*5+0,2*4+0,3*5+0,1*1 = 4,4$$

$$I_{этубаз} = 0,2*3+0,2*3+0,2*3+0,3*4+0,1*2 = 3,2$$

$$K_T = \frac{I_{этупр}}{I_{этубаз}}, \quad (12)$$

где $I_{этупр}$ – базовый проект;

$I_{этубаз}$ – разрабатываемый проект;

$$K_T = \frac{4,4}{3,2} = 1,375;$$

Так как работа выполнялась совместно, то капитальные затраты следует сложить. У второго разработчика капитальные затраты составили:

$$K = 14582 + 500 = 15082 \text{ руб.}$$

Общие капитальные затраты составили:

$$24541 + 15082 = 39623 \text{ руб.}$$

Для расчёта экономического эффекта рассчитываем приведённые затраты на единицу работ, выполняемые по базовому и разрабатываемому варианту по формуле 13:

$$Z_i = C_i + E_n * Z_{\text{пр}i}, \quad (13)$$

где C_i – текущие эксплуатационные затраты единицы i -го вида работ, р;

$E_n = 0,33$ – нормативный коэффициент экономической эффективности;

$Z_{\text{пр}i}$ – суммарные затраты, связанные с внедрением проекта.

Возьмём проектные затраты на зарплату персонала, обслуживающих сайт, которые составляют 6000 рублей. Эксплуатационные затраты составят 9600 руб.

$$Z_{\text{пр}} = 9600 + 0,33 * 39623 = 22676$$

Для базового варианта возьмём затрату на зарплату оператора. Он обслуживает сайт и работает 6 часов в сутки. Затраты на его зарплату составят 32604 руб. Эксплуатационные затраты составят 39626 рублей.

$$Z_{\text{баз}} = 39626 + 0,33 * 39623 = 52702$$

Экономический эффект от использования разрабатываемой системы определяется по формуле 14:

$$\mathcal{E} = (Z_{\text{баз}} * K_T - Z_{\text{пр}}) * V, \quad (14)$$

где $Z_{\text{баз}}$ и $Z_{\text{пр}}$ – это приведённые затраты на единицу работ, выполненные с помощью базового и проектируемого варианта работ.

V – это объём работ, выполняемых с помощью разрабатываемого проекта. Так как работа не является автоматизированной, то посчитаем затраты, которые компенсирует разрабатываемый проект.

Для начала узнаем, сколько времени тратит человек на поиск необходимой услуги. Поиск объявлений составляет примерно 20 минут. К этому добавляется ещё 10 минут на их сортировку для наиболее выгодного решения. Ещё 5 минут прибавляются на то, чтобы написать или позвонить и обговорить выполнение услуги. Ждать исполнителя услуги потребуется несколько часов или дней в связи с плотным графиком работы. Возьмём в среднем 6 часов. Выходит необходимо $360+20+10+5=395$ минут или чуть больше 6 часов необходимо для проведения помощи.

Для того, чтобы воспользоваться сервисом ИТ-Помощь ХТИ, достаточно перейти на официальный сайт ХТИ и перейти на сайт сервиса, посмотрев услуги. Так как студентов насчитывается большое количество, то исполнителей на заказ найдётся в среднем за 3 часа.

Посчитаем соотношение времени у аналогов и сервиса, который предлагает ХТИ – филиал СФУ. $100\% - (180 / 385 * 100\%) = 53,3\%$. На 53% работа выполнится быстрее. В итоге получается, что заказчик, который воспользовался сервисом ХТИ – филиал СФУ, сэкономит время. Можно сказать, что проект оказался на 53% автоматизируемым, поэтому $53\% / 100 = 0,53$

$$\text{Э} = (52702 * 1,375 - 22676) * (0,53) = 26389$$

Срок окупаемости затрат на разработку проекта рассчитывается по формуле 15:

$$T_{\text{ок}} = \frac{3\text{пп}}{\text{Э}}, \quad (15)$$

где $T_{\text{ок}} = \frac{39623}{26389} = 1,5$

Коэффициент фактической экономической эффективности рассчитывается по формуле 16:

$$E_{\phi} = \frac{1}{T_{\text{ок}}}, \quad (16)$$

где $E_{\phi} = \frac{1}{1,5} = 0,67$

Поскольку $E_{\phi}=0,67>E_n$, то разработка и внедрение информационной системы является эффективной. А именно эффект от её использования окупает все затраты.

3.4 Выводы по разделу «Экономическое обоснование реализации проекта»

Были определены сроки начала и окончания работ по внедрению проекта. Всего требуется 2 недели на разработку. Посчитана стоимость оборудования для разработки, которая составила 69193 руб. Посчитана стоимость программного обеспечения для разработки, которая составила 8350 руб. Также посчитаны затраты на ПО, чтобы пользоваться сайтом. Они составили 0 рублей, так как чтобы пользоваться сайтом нужен браузер, который доступен бесплатно. Посчитана амортизация в размере 951 руб. Определены капитальные затраты у первого и второго разработчиков. Они составили 24541 руб. и 15082 руб. Определены эксплуатационные затраты, которые составили 9600 руб. Составлена 1 диаграмма по капитальным затратам. Рассчитан показатель качества. Базовый проект составил 4,4, разрабатываемый проект 3,2. На основе них была определена экономическая эффективность проекта, которая составила 0,67. Таким образом, разработка и внедрение ИС оказались эффективными.

ЗАКЛЮЧЕНИЕ

В ходе выполнения ВКР была поставлена цель и выполнены задачи:

1. проанализирована предметная область;
2. описана информационная система графическим методом по нотации IDEF0;
3. описана информационная система графическим методом по нотации IDEF3;
4. описаны потоки данных информационной системы графическим методом по нотации DFD;
5. выбран жизненный цикл для информационной системы;
6. разработан сайт, который позволяет обслуживать клиентов;
7. посчитаны общие затраты;
8. рассчитана экономическая эффективность проекта;

Сделан анализ предметной области. Доказана положительная актуальность ИС. Сравнены средства разработки и выбраны те, которые лучше всего подходят для создания данной ИС.

Составлена диаграмма IDEF0 графическим методом. Описаны входы, выходы, управление и механизмы этой диаграммы. Составлены диаграммы IDEF3 и DFD. Описана их работа.

Выбрана спиральная модель жизненного цикла информационной системы и объяснено обоснование этого выбора.

Рассчитана экономическая эффективность проекта.

Подсчитаны капитальные затраты и амортизация. Также доходы системы. Все затраты при создании и эксплуатации системы показали, что разработка ИС является обоснованной.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. 6 редакторов и IDE для работы с JavaScript [Электронный ресурс]. – Режим доступа: https://skillbox.ru/media/code/6_redaktorov_i_ide_dlya_raboty_s_javascript/
2. Датацентр+ [Электронный ресурс]. – Режим доступа: <http://datacenterplus.ru/>
3. Знакомство с нотацией IDEF0 и пример использования [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/trinion/blog/322832/>
4. Как установить NodeJS и NPM в Windows [Электронный ресурс]. – Режим доступа: https://zen.yandex.ru/media/id/5a16a1eee86a9e56a71cf8ef/kak-ustanovit-nodejs-i-npm-v-windows-5dc2915034808200b20fb55a?utm_source=serp
5. Компания "Орион телеком" [Электронный ресурс]. – Режим доступа: <https://www.google.ru/maps/place/ул.+Карла+Маркса,+44,+Абакан,+Хакасия+Республика,+655017/@53.729282,91.4360853,680m/data=!3m2!1e3!4b1!4m5!3m4!1s0x5d2c17fe3dd21039:0x6f815cf40dc9e585!8m2!3d53.729282!4d91.438274?hl=ru>
6. Местам инфо [Электронный ресурс]. – Режим доступа: <https://mestam.info/>
7. Методология описания бизнес-процессов IDEF3 [Электронный ресурс]. – Режим доступа: https://studopedia.ru/4_131349_metodologiya-opisaniya-biznes-protsessov-IDEF.html
8. Мультимедиа [Электронный ресурс]. – Режим доступа: <http://deltatrade.ru/>
9. Одностраничные (spa) и многостраничные (pwa) веб-приложения [Электронный ресурс]. – Режим доступа: <https://vc.ru/seo/108149-odnostranichnye-spa-i-mnogostranichnye-pwa-veb-prilozheniya>
10. Организационная структура ХТИ — филиала СФУ [Электронный ресурс]. – Режим доступа: <http://khti.sfu-kras.ru/institute/struktura/>
11. Орион телеком [Электронный ресурс]. – Режим доступа: <https://orionnet.ru/abk/>

12. Руководство пользователя Xiaomi [Электронный ресурс]. – Режим доступа: <https://www.mi.com/ru/service/support/userguide.html>
13. Сервисный центр ПК сервис [Электронный ресурс]. – Режим доступа: <http://pk-service19.ru/>
14. Сколько времени россияне сидят в интернете [Электронный ресурс]. – Режим доступа: https://www.gazeta.ru/tech/2020/02/12/12956929/we_are_social.shtml
15. Установка | 1.2.3.2 Создание React-приложения [Электронный ресурс]. – Режим доступа: <https://learn-reactjs.ru/introduction/installation>
16. Федеральный закон от 29 декабря 2012 г. № 273-ФЗ [Электронный ресурс]. – Режим доступа: <http://xn--273--84d1f.xn--plai/zakonodatelstvo/federalnyy-zakon-ot-29-dekabrya-2012-g-no-273-fz-ob-obrazovanii-v-rf>
17. Хакасский технический институт – История [Электронный ресурс]. – Режим доступа: <http://khti.sfu-kras.ru/institute/istoria.php>
18. Что такое DFD (диаграммы потоков данных) [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/trinion/blog/340064/>
19. Add Firebase to your JavaScript project [Электронный ресурс]. – Режим доступа: <https://firebase.google.com/docs/web/setup>
20. Firebase Pricing – История [Электронный ресурс]. – Режим доступа: <https://firebase.google.com/pricing?hl=ru>

Выпускная квалификационная работа выполнена мной самостоятельно.
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 20 наименований.

Один экземпляр сдан на кафедру.

С.В. Кокшаров

Подпись, дата

Студенту Кокшарову Сергею Валерьевичу

Группа ХБ 16-03

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: «Разработка информационной системы сервиса по оказанию услуг программного и аппаратного обеспечения»

Утверждена приказом по институту № 216 от 06.04.2020

Руководитель ВКР: Е. Н. Скуратенко, зав. кафедрой «Прикладная информатика, математика и естественно-научные дисциплины», кандидат технических наук, ХТИ – филиал СФУ

Исходные данные для ВКР: создать он-лайн сервис, позволяющий проводить коммуникацию с потребителями услуг программного и аппаратного обеспечения.

Перечень разделов ВКР:

1. Анализ предметной области информационной системы.
2. Практическая реализация сервиса по оказанию услуг программного и аппаратного обеспечения.
3. Оценка экономической эффективности внедрения информационной системы.

Перечень графического материала: нет

Руководитель ВКР


подпись

Е.Н. Скуратенко

Задание принял к исполнению


подпись

С.В. Кокшаров

«06» 04 2020 г.