

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

---

Кафедра «Прикладная математика и информатика»  
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем  
(код и наименование направления подготовки / специальности)

---

Мобильные и сетевые технологии  
(направленность (профиль) / специализация)

---

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка онлайн сервиса «Перезачет внешнего курса / курса ДПО»

Обучающийся

А.М. Соломанидина

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд. пед. наук, доцент, Т.А. Агошкова

(ученая степень, ученое звание, Инициалы Фамилия)

Консультант

А.В. Москалюк

(ученая степень, ученое звание, Инициалы Фамилия)

Тольятти 2023

## Аннотация

Название бакалаврской работы: «Разработка онлайн сервиса «Перезачет внешнего курса / курса ДПО»».

Целью работы - разработка онлайн сервиса для предоставления студентам возможности подавать заявление о перезачете онлайн-курсов, и для предоставления студентам возможности просматривать результат поданного заявления.

Объектом бакалаврской работы является процесс перезачета внешнего курса / курса ДПО.

Предметом бакалаврской работы являются этапы проектирования, разработки и тестирования онлайн сервиса «Перезачет внешнего курса / курса ДПО».

Выпускная работа может быть разделена на следующие логически взаимосвязанные части: исследование предметной области, проектирование онлайн сервиса, разработка сервиса и тестирование разработанного сервиса.

Первая глава работы посвящена анализу предметной области и постановкой задачи. Определили, что такое онлайн сервис и что можно делать с их помощью. Также в первой главе определили главную задачу онлайн сервиса.

Во второй главе определили основную функциональность онлайн сервиса и разработали прототип пользовательского интерфейса.

В третьей главе выбрали язык программирования, выбрали средство разработки для реализации онлайн сервиса, а также разработали и протестировали онлайн сервис.

Бакалаврская работа состоит из пояснительной записки на 41 страниц, введения на 2 страницы, включая 26 рисунков, списка из 20 источников, в том числе 5 источников на иностранном языке.

## **Abstract**

The aim of the work is to develop an online credit transfer service for external courses / further vocational education courses which will provide students with the opportunity to apply for the credit transfer and view the result of the submitted application.

The object of the work is the process of credit transfer.

The subject of the work is the stages of design, development and testing of the online service.

The work can be divided into the following logically interconnected parts: research of the subject area, online service design, development and testing.

The first chapter of the work is devoted to the analysis of the subject area and the formulation of the problem. We determined what an online service is and what can be done with the help of it. Also in the first chapter, the main task of the online service was determined.

In the second chapter, we defined the main functionality of the online service and developed a prototype user interface.

In the third chapter, we chose a programming language, chose a development tool for implementing an online service, and developed and tested an online service.

The graduation work consists of an explanatory note of 41 pages, an introduction of 2 pages, and includes 26 figures and a list of 20 references, including 5 sources in a foreign language.

## Оглавление

Введение.....	5
Глава 1 Анализ предметной области.....	7
1.1 Характеристика предметной области .....	7
1.2 Постановка задачи .....	8
Глава 2 Проектирования сервиса.....	14
2.1 Определить цель сервиса и его основную функциональность ....	14
2.2 Разработка диаграмм логической модели .....	15
2.3 Разработка диаграммы последовательности.....	16
2.4 Разработка пользовательского интерфейса.....	18
Глава 3 Реализация онлайн сервиса .....	22
3.1 Выбор языка программирования.....	22
3.2 Выбор средств разработки для реализации онлайн сервиса .....	23
3.3 Разработка страницы авторизации.....	25
3.4 Разработка главной страницы.....	26
3.5 Разработка страницы подачи запроса .....	28
3.6 Разработка страницы “Мои запросы” .....	32
3.7 Тестирование сервиса.....	34
Заключение .....	38
Список используемых источников.....	39

## Введение

Онлайн сервисы играют огромную роль в современном мире. Они предоставляют широкий спектр услуг, которые мы можем получать через интернет:

- Обмена информацией и коммуникаций между людьми;
- Покупки товаров и услуг онлайн;
- Поиска и сравнения товаров и услуг;
- Обмена документами и информацией в рамках бизнес-процессов;
- Облачного хранения и обмена файлами;
- Создания онлайн-образовательных курсов и учебных ресурсов;
- Онлайн-банкинга и финансовых услуг;
- Игр и развлечений;
- Онлайн-бронирования и покупки билетов на различные мероприятия;
- Организации виртуальных встреч и конференций.

«На данный момент существует тысячи всевозможных сервисов, которые помогают нам в решении различных проблем» [8]. В общем, онлайн-сервисы облегчают жизнь людей и обеспечивают быстрый и удобный доступ к различным видам информации и услуг. Онлайн сервисы сделали нашу жизнь более удобной и доступной, что делает их незаменимыми в современном мире.

Целью данной работы является разработка онлайн сервиса «Перезачет внешнего курса / курса ДПО».

Для достижения цели данной работы необходимо составить список задач:

- исследовать предметную область;
- спроектировать онлайн сервис;
- разработать онлайн сервис;
- протестировать онлайн сервис.

Структура работы представлена тремя главами.

В первой главе исследовали предметную область.

Во второй главе определили требования к разрабатываемому сервису и спроектировали онлайн сервис.

В третьей главе разработали и протестировали онлайн сервис.

## **Глава 1 Анализ предметной области**

### **1.1 Характеристика предметной области**

«Онлайн сервис — это программа, выполняющая определенные действия по запросам пользователей через интернет, без необходимости скачивания и установки на локальный компьютер» [12]. Онлайн-сервисы позволяют решать проблемы и вопросы пользователей в режиме онлайн. Примеры онлайн-сервисов включают онлайн-банкинг, образовательные услуги, новости, покупки, знакомства, сайты компьютерной помощи, социальные сети и электронную почту. Онлайн-сервисы обеспечивают удобство и доступность для пользователей, позволяя им получать необходимые услуги и информацию в любое время и из любого места, где есть доступ к интернету. Они упрощают выполнение различных задач, ускоряют обмен информацией и обеспечивают удобство использования для пользователей.

С помощью онлайн-сервисов пользователь может, например, общаться, выполнять вычисления, нарисовать рисунок или схему, хранить, редактировать и публиковать информацию, смотреть фильм, слушать музыку и многое другое.

Онлайн-сервисы могут быть простыми до сложными. Базовый онлайн-сервис может помочь пользователям получить необходимые данные через поисковую систему, в то время как сложным может отправить онлайн-заявку на ипотеку в банке.

Онлайн-сервисы также могут быть бесплатными или платными. Например, онлайн-сервисы, такие как Netflix и Microsoft Office, взимают абонентскую плату, но многие другие бесплатны, такие как Wikipedia и Computer Hope. Онлайн-сервисы сейчас настолько распространены, широко распространены и часто даже бесплатны, что большинство абонентов даже не осознают, что они ими пользуются.

Разработкой собственного онлайн-сервиса может быть заинтересован представитель любой сферы. Они могут являться как вполне самостоятельными единицами, так и дополнять основной сайт. К примеру, на сайте можно разместить страницу с онлайн-сервисом по заказу услуги или покупке продукции данной компании.

## **1.2 Постановка задачи**

Главная задача состоит в разработке онлайн сервиса «Перезачет внешнего курса / курса ДПО». Сервис должен предоставлять возможность загружать сертификат о прохождении онлайн-курса и заявление о перезачете онлайн-курсов, после этого у сотрудника появляется заявление и сертификат в Битрикс24, также студент имеет возможность в личном кабинете просматривать результат поданной заявки.

Битрикс24 — это онлайн-сервис, в котором собраны все нужные инструменты для работы компании и управления бизнесом. Перенесите в единое пространство и автоматизируйте все рабочие коммуникации, продажи, проекты и бизнес-процессы. «Битрикс24 – это набор полезных инструментов, которые помогают бизнесу работать: CRM, Видеозвонки, Документы Онлайн, Задачи и проекты, Контакт-центр, конструктор сайтов и Интернет-магазины» [2].

«Онлайн-курсы — это формат обучения, который позволяет получать знания и выполнять домашние задания через интернет» [7]. Представляет собой совокупность графической, текстовой, цифровой, звуковой, видео-, фото- и другой информации по соответствующей научно-практической области знаний, обеспечивающая обучаемым активное овладение знаниями / умениями / навыками в данной области, в соответствии с требованиями ФГОС (Федеральный государственный образовательный стандарт).

Сертификат – это документ, подтверждающий оценку результатов обучения обучающегося на онлайн-платформе и количество полученных



зачетных единиц или академических часов. Сертификатом может служить справка об обучении, подтвержденный сертификат или другой документ, официально оформленный и подтверждающий факт и результаты обучения на онлайн-платформе.

Для учета результатов обучения на онлайн-курсах обучающийся подает заявление о перезачете. К заявлению прикладывается сертификат.

Перезачет - зачет определенного объема ОПОП (основная профессиональная образовательная программа) (зачетных единиц или академических часах) вместе с результатами промежуточных аттестаций. Зачет результатов обучения в форме перезачета полностью освобождает обучающегося от необходимости повторного изучения (прохождения промежуточной аттестации) соответствующей дисциплины. Объем онлайн-курса может и не совпадать с объемом дисциплины в ОПОП (основная профессиональная образовательная программа).

Платформами, на которых студент может пройти открытые онлайн-курсы, являются: free Росдистант, Национальная платформа открытого образования, Stepik, Универсариум.

Росдистант — федеральный образовательный проект, в рамках которого студентам предоставляется возможность получить высшее профессиональное образование в дистанционной форме. Проект реализуется на базе Тольяттинского государственного университета. «Наряду с самими образовательными продуктами, которые предлагаются в рамках дистанционного онлайн-обучения, одним из ключевых конкурентных преимуществ Росдистанта является современная клиенто-ориентированная система сопровождения студентов, обеспечивающая их индивидуальное консультирование» [3]. Главная страница платформы Росдистант продемонстрирована на рисунке 1.

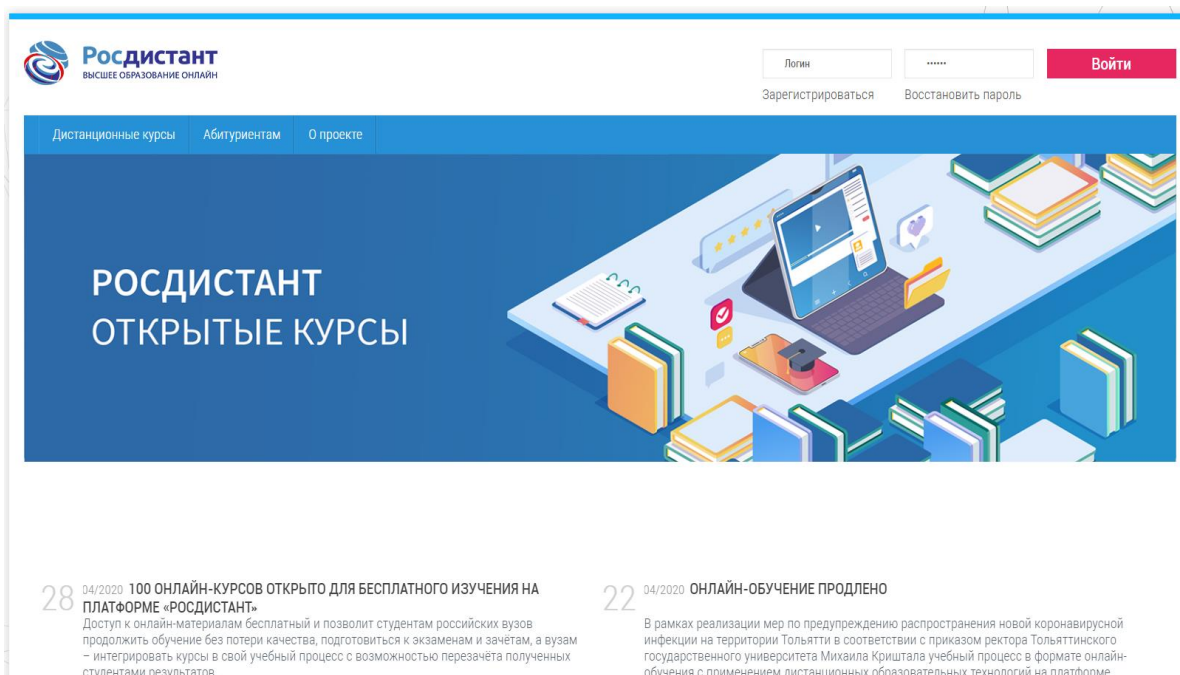


Рисунок 1 – Платформа “Росдистант”

Национальная платформа открытого образования (НПОО, Openedu.ru) — это образовательная платформа, предлагающая массовые онлайн-курсы ведущих российских вузов. Главная страница национальной платформа открытого образования продемонстрирована на рисунке 2.

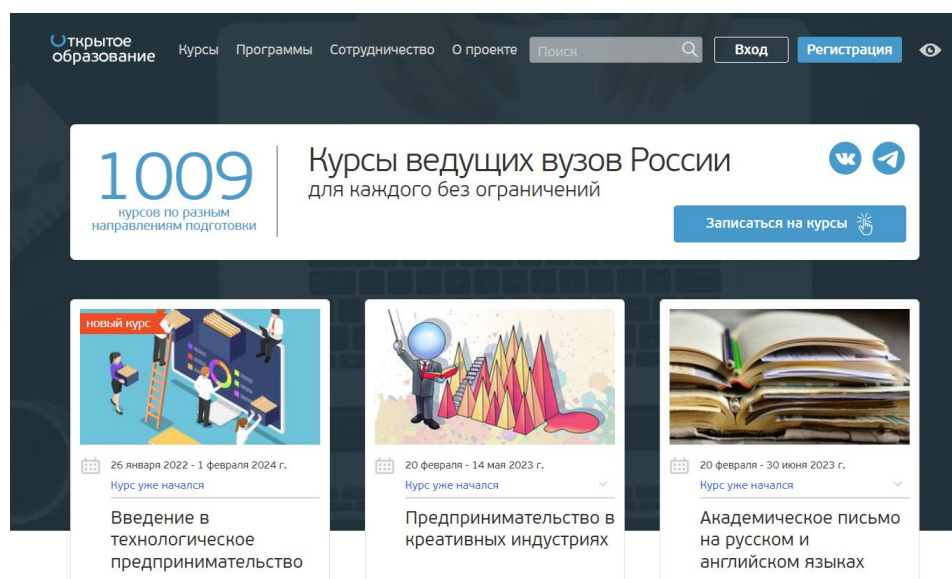


Рисунок 2 – Платформа “Открытое образование”

«Stepik — образовательная платформа и конструктор онлайн-курсов» [10]. Позволяет любому зарегистрированному пользователю создавать интерактивные обучающие уроки и онлайн-курсы, используя видео, тексты и разнообразные задачи с автоматической проверкой и моментальной обратной связью. В процессе обучения студенты могут вести обсуждения между собой и задавать вопросы преподавателю на форуме. Основные охватываемые курсами дисциплины — программирование, математика, биоинформатика и биология, экономика; основной язык курсов — русский, есть курсы на английском языке. По состоянию на 2020 год на платформе зарегистрировано 5 миллионов пользователей. Целевые аудитории — школьники, студенты, начинающие специалисты. Главная страница платформы Stepik продемонстрирована на рисунке 3.

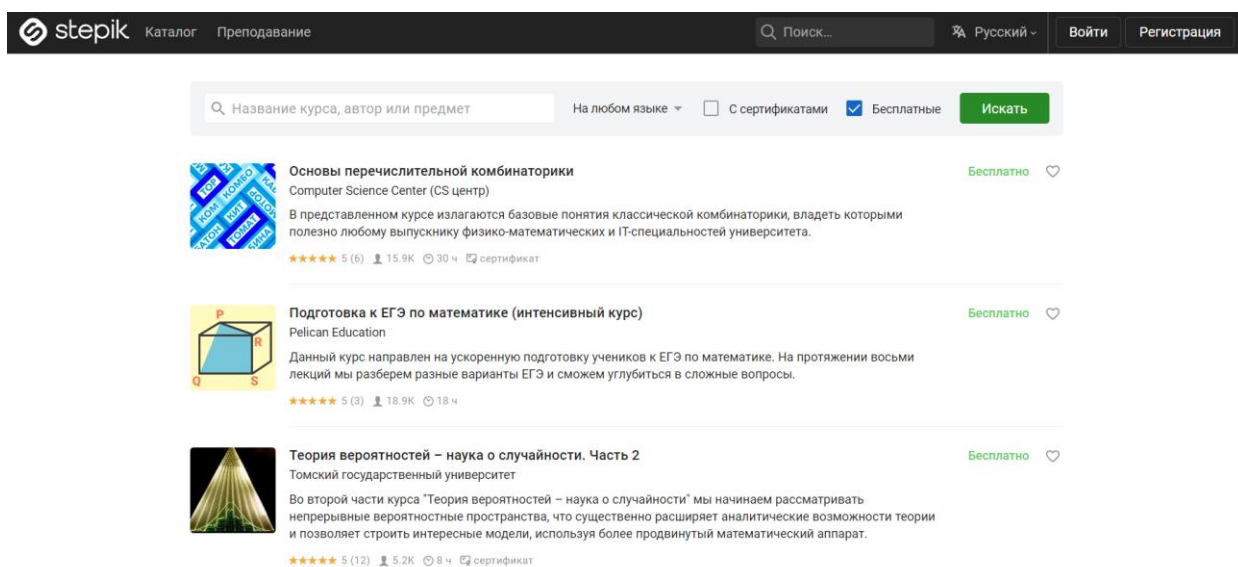


Рисунок 3 – Платформа “Stepik”

Универсариум — российская система электронного онлайн-образования, построенная по технологии массовых открытых онлайн-курсов. В системе представлены бесплатные образовательные курсы преподавателей ряда университетов страны, а также российских научных центров. Основные направления, по которым создаются и будут разрабатываться курсы — химия,

физика, математика, экономика, программирование, астрономия, биология. Особое внимание уделено межфакультетским курсам, и курсам, находящимся «на стыке» дисциплин. На первом этапе проекта преподавание предполагается на русском языке, на следующих этапах планируется развитие проекта за рубежом и перевод курсов на основные европейские языки. «Курсы и образовательные программы, представленные в «Универсариуме», создаются совместно с лучшими преподавателями ведущих вузов страны, вошедших в проект» [9].

В отличие от систем дистанционного обучения, использующихся в российских университетах, Универсариум предлагает полноценные бесплатные курсы, выполненные по образовательным стандартам электронного обучения, которые включают видеолекции, самостоятельные задания, домашние задания, тесты, групповую работу и итоговую аттестацию. Проект реализуется при поддержке РИА Наука и Агентства стратегических инициатив.

Главная страница платформы Универсариум продемонстрирована на рисунке 4.

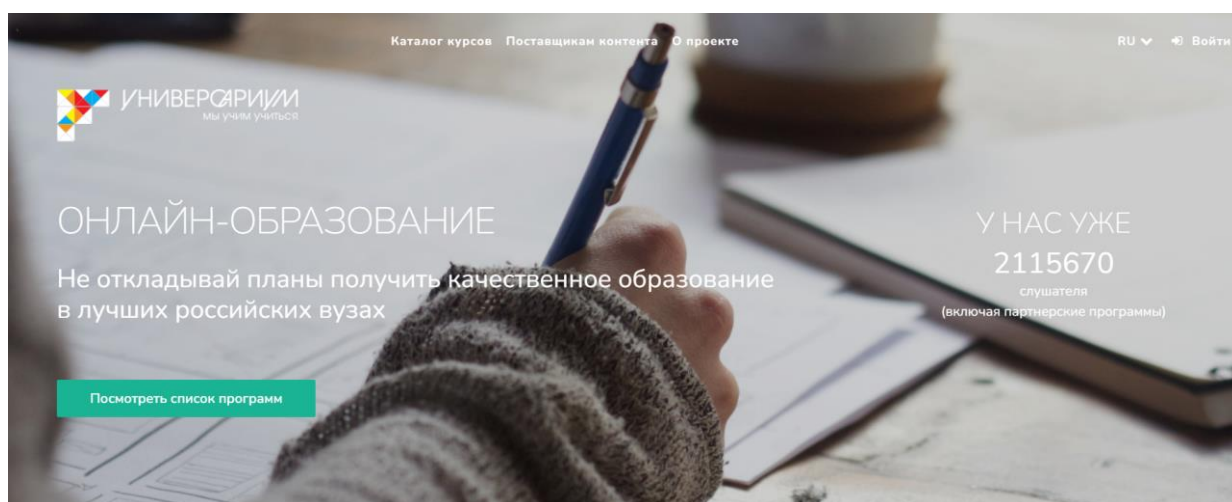


Рисунок 4 – Платформа “Универсариум”

Подытожим, цель работы – создание онлайн-сервиса «Перезачет внешнего курса / курса ДПО».

Для достижения поставленной цели необходимо:

- спроектировать сервис;
- подобрать необходимые средства разработки и выбрать язык программирования;
- разработать сервис;
- протестировать работу сервиса.

Также уделим особое внимание дизайну интерфейса, он должен быть интуитивно понятным и удобным для пользователей.

#### Выводы по главе 1

В данной главе были сформированы требования к создаваемому онлайн сервису, а также был проведен анализ предметной области.

## Глава 2 Проектирования сервиса

### 2.1 Определить цель сервиса и его основную функциональность

Целью сервиса является перезачет внешнего курса / курса ДПО, студент загружает сертификат, свидетельствующий о прохождении какого-либо курса, полученный на платформе для онлайн-образования и заявления о перезачете онлайн-курсов. Сертификат и заявление попадает сотруднику в сервис Битрикс24, где он с ними работает. Студент также может в личном кабинете отслеживать результаты поданной заявки.

Требования к разрабатываемому сервису можно разделить на функциональные и нефункциональные.

Функциональные требования описывают, что система должна делать, то есть их цель - определить, какие функциональные возможности должны быть доступны в системе. Например, какие действия должна выполнять система, какой функционал должен быть реализован и т.д.

К функциональным требованиям относятся:

- Пользователь должен иметь возможность авторизации в личном кабинете;
- Пользователь должен иметь возможность отправлять свой сертификат и заявление;
- Пользователь должен иметь возможность просматривать состояние дела.

«Если функциональные требования определяют, что должна делать система, то нефункциональные требования описывают, как она будет это делать» [14]. К ним относятся требования к производительности, надежности, безопасности, удобству использования, масштабируемости и т.д.

К нефункциональным требованиям относятся:

- Сервис должен быть стабильным и надежным;

- Сервис должен быть удобным для использования и интуитивно понятным;
- Сервис должен быть безопасным: уровень защиты хранимых данных и доступа к ним;
- Сервис должен быть легальным: соответствие правовым и юридическим требованиям, включая права интеллектуальной собственности, защиту персональных данных и прочее.

## **2.2 Разработка диаграмм логической модели**

Логическое моделирование - это аналитический подход, при котором строится логическая структура системы, исключая детали реализации. Логическая модель используется для описания того, как система должна работать, определяя связи между объектами и процессами в рамках этой системы. Логическое моделирование позволяет увидеть систему в абстрактной форме, сосредоточиваясь на ее аспектах, которые важны для решения конкретных задач. Оно может использоваться для разработки новых систем или для модернизации и оптимизации существующих.

В этом разделе для описания деятельности используем диаграмму вариантов использования.

Диаграмма вариантов использования (Use Case Diagram) – это модель, которая используется в разработке программного обеспечения для описания функциональных требований к системе. Она показывает, как различные типы пользователей (акторы) взаимодействуют с системой, какие действия они выполняют в рамках системы и возвраты, которые могут быть получены после выполнения этих действий. «При этом актеры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой» [11]. Диаграмма вариантов использования помогает определить границы системы, а также

выявить соответствие функциональных требований и предложений пользователям.

Ниже на рисунке 5 предоставлена диаграмма вариантов использования.

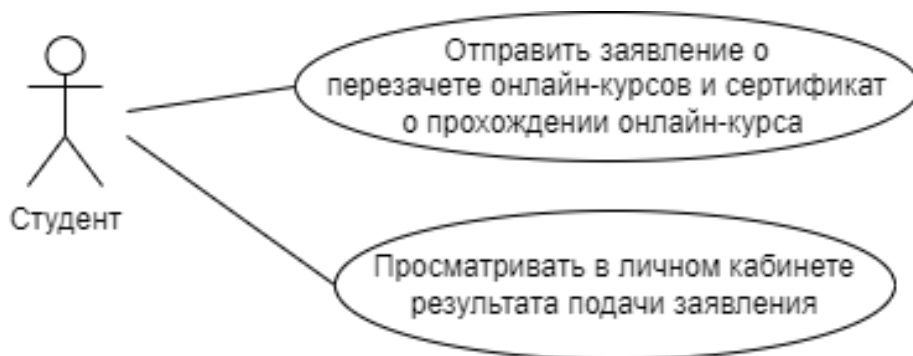


Рисунок 5 – Диаграмма вариантов использования

В этом сервисе присутствует тип пользователей “Студент”.

Студенты могут пользоваться нашей системой следующим образом:

- Загружать заявление о перезачете онлайн-курсов, прилагая к нему сертификат;
- Отслеживать в личном кабинете результата поданной заявки.

### 2.3 Разработка диаграммы последовательности

«Диаграмма последовательности — это тип диаграммы, который позволяет описать взаимодействие между объектами в системе в виде последовательности сообщений, действий и операций, отображая порядок выполнения действий и обмена информацией между объектами во времени» [5]. Она помогает понимать временные зависимости между объектами и процессами, а также отображает сообщения и вызовы функций между ними. Диаграмма последовательности особенно удобна для моделирования процессов с большим количеством этих взаимодействий, таких как разработка программного обеспечения или бизнес-процессы.



В соответствии с функциями сервиса необходимо разработать диаграмму последовательности.

Ниже на рисунке 6 предоставлена диаграмма последовательности.

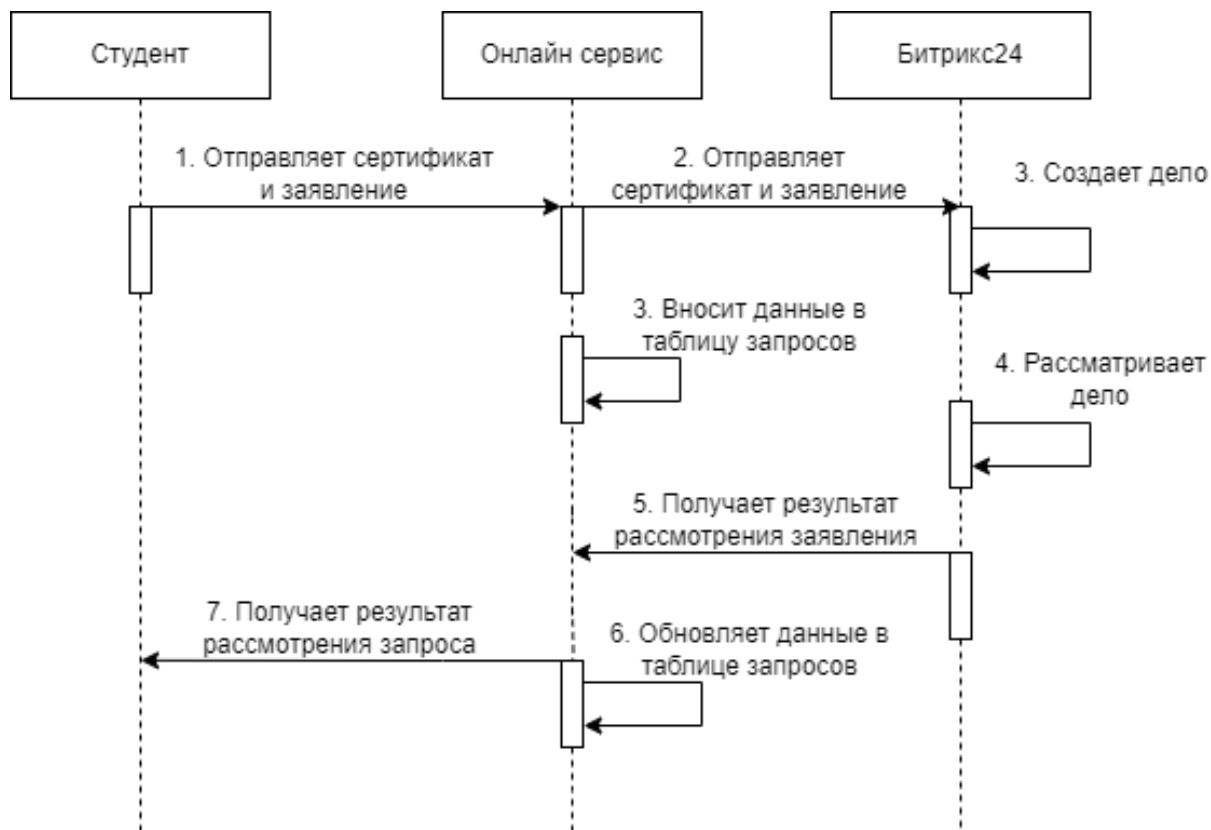


Рисунок 6 – Диаграмма последовательности

Для разработки диаграммы последовательности необходимо выполнить ряд действий, которые описаны ниже:

- Студенту нужно отправить свое заявление о перезачете онлайн-курсов и сертификат о прохождении онлайн-курса;
- Онлайн сервис отправляет заявление и сертификат сотруднику в Битрикс24;
- После того как сотрудник рассмотрит заявление, сервис получит результат рассмотрения;
- Студент получает результат о перезачете курса.

## 2.4 Разработка пользовательского интерфейса

Пользовательский интерфейс (UI) - это совокупность элементов и функций, предназначенных для совместной работы пользователя с компьютерной программой или устройством. Он представляет собой графическое отображение программы, которое пользователь видит на экране. Пользовательский интерфейс включает в себя окна, кнопки, меню, панели инструментов, диаграммы, графики, изображения и другие элементы, которые помогают пользователям взаимодействовать с приложением. Оно обычно разрабатывается с учетом потребностей и привычек пользователей с целью облегчения доступа к функционалу программы и улучшения удобства и эффективности использования. «Интерфейс оказывает значительно влияние на скорость и качество выполнения работы пользователем» [1].

Разрабатывая пользовательский интерфейс, необходимо придерживаться основных принципов, определяющие взаимодействие с пользователем и саму структуру приложения в целом.

Рассмотрим основные принципы:

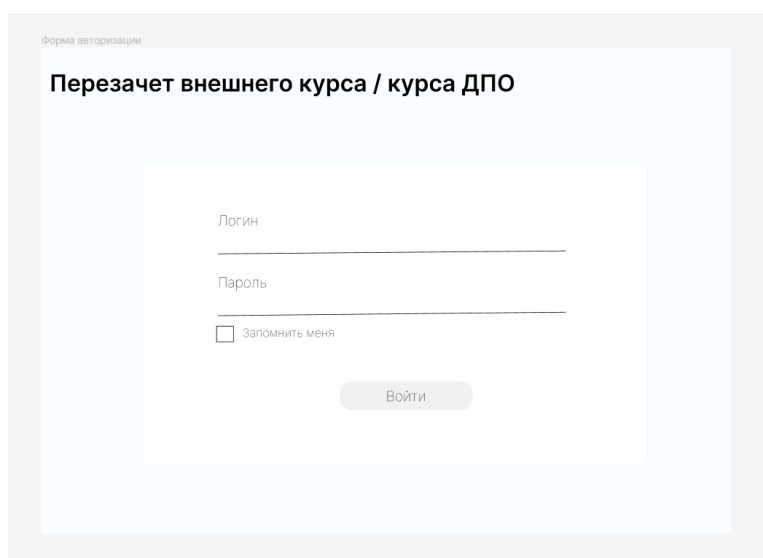
- Принцип простоты и понятности: пользователь должен быстро понимать, как пользоваться интерфейсом и выполнять необходимые действия без лишних затруднений.
- Принцип понятности языка и символов: используйте ясные и понятные слова и символы, которые легко запомнить и не вызывают путаницы.
- Принцип консистентности: интерфейс должен быть предсказуемым и последовательным, чтобы пользователь мог легко ориентироваться и не переживать, что на следующей странице он столкнется с новыми элементами.
- Принцип удобства и скорости: интерфейс должен быть удобным для пользователей, чтобы они могли быстро и легко находить нужный контент и выполнять действия.

– Принцип гибкости и настроенности: учтите, что пользователи могут иметь разные потребности и установленные предпочтения, поэтому должны существовать возможности настройки интерфейса.

– Принцип визуальной и информационной ясности: интерфейс должен быть привлекательным и простым в восприятии, а информация должна подаваться иерархически и легко читаться.

Среди современных средств разработки дизайна выделяется онлайн-сервис Figma. Figma - это онлайн-инструмент для дизайна интерфейсов, который позволяет создавать, редактировать и совместно работать над графическими проектами «В Фигме можно отрисовать элементы интерфейса, создать интерактивный прототип сайта и приложения, иллюстрации, векторную графику» [4]. Figma также значительно упрощает совместную работу, позволяя пользователям работать одновременно над одним проектом и делиться своими работами и мнениями.

В соответствии с запланированными экранами, используя онлайн-инструмент Figma, разработаем прототип формы авторизации. Важно понимать, что конечный вид экрана может отличаться от прототипа. Прототип формы авторизации представлен на рисунке 7.



Форма авторизации

**Перезачет внешнего курса / курса ДПО**

Логин

Пароль

Запомнить меня

Войти

Рисунок 7 – Форма авторизации

Предполагаемый алгоритм действий студента таков. Студент вводит данные для входа и после нажатия на кнопку войти, если вводимые данные верны, происходит переход на главную страницу. Экран главной страницы показан на рисунке 8.

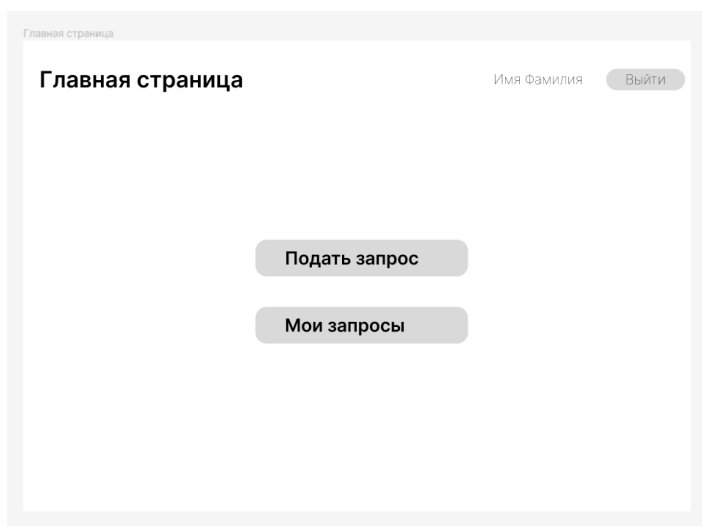


Рисунок 8 – Главная страница

При нажатии на кнопку “Подать запрос” происходит переход на страницу подачи запроса. Форма подачи запроса представлена на рисунке 9.

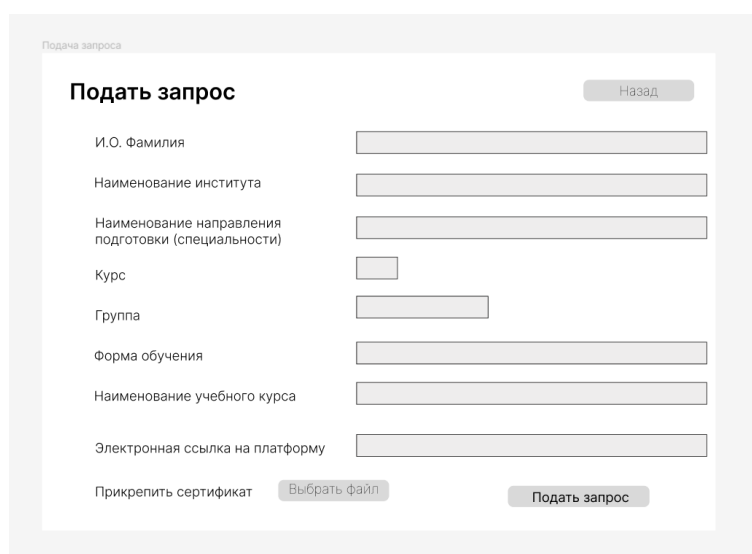
The image displays a form titled 'Подать запрос'. At the top right of the form is a 'Назад' button. The form contains several input fields: 'И.О. Фамилия', 'Наименование института', 'Наименование направления подготовки (специальности)', 'Курс', 'Группа', 'Форма обучения', and 'Наименование учебного курса'. Below these is a field for 'Электронная ссылка на платформу'. At the bottom left, there is a section 'Прикрепить сертификат' with a 'Выбрать файл' button. A 'Подать запрос' button is located at the bottom right of the form.

Рисунок 9 – Форма подачи запроса

При нажатии на главной странице на кнопку “Мои запросы” происходит переход на страницу запросов. Страница запросов показана на рисунке 10.

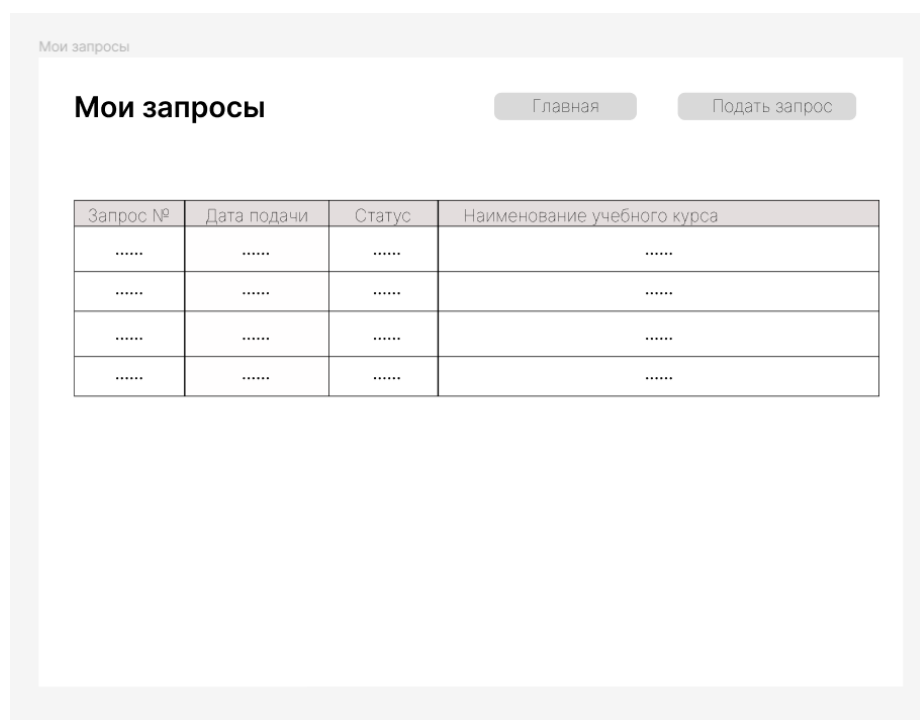


Рисунок 10 – Страница запросов

## Выводы по главе 2

В ходе данной главы были рассмотрены основные требования, используемые для разработки онлайн сервиса. Требования были составлены и разделены на две категории – функциональные и нефункциональные. С использованием диаграммы вариантов использования и последовательности построили логическая модель разрабатываемого онлайн сервиса. Также в ходе данной главы был разработан пользовательский интерфейс.

## Глава 3 Реализация онлайн сервиса

### 3.1 Выбор языка программирования

Первым делом при разработке онлайн сервиса мы должны определиться с языком программирования.

Для создания онлайн сервиса может потребоваться несколько языков программирования, в зависимости от функциональности и технологий, используемых в сервисе. Некоторые из таких языков программирования могут включать:

- JavaScript - для создания интерактивных пользовательских интерфейсов и разработки клиентской части веб-приложения.
- HTML и CSS - для разметки веб-страниц и оформления их визуального стиля соответственно.
- PHP, Python, Ruby - для разработки серверной части веб-приложения и обеспечения взаимодействия между клиентом и сервером.
- SQL - для работы с базами данных, которые могут использоваться в сервисе для хранения информации о пользователях и их действиях.
- Java - используется для разработки масштабируемых веб-приложений и сервисов.
- Node.js - для разработки серверной части веб-приложений на базе JavaScript.
- Ruby on Rails - фреймворк для разработки веб-приложений на основе языка Ruby.

Для проектирования онлайн сервиса будет использовать JavaScript, HTML и CSS так как:

- С помощью JavaScript, HTML и CSS можно создавать не только веб-страницы, а также визуальные эффекты, анимацию, интерактивность и многое другое.

– Языки JavaScript, HTML и CSS являются очень популярными технологиями и имеют большое сообщество разработчиков, которые создают инструменты и библиотеки для их использования.

– Совместимость с другими языками программирования и инструментами - JavaScript, HTML и CSS могут легко взаимодействовать с другими языками программирования и инструментами, такими как SQL и PHP.

### **3.2 Выбор средств разработки для реализации онлайн сервиса**

«Среда разработки или IDE (сокр. англ. Integrated Development Environment – интегрированная среда разработки) — специальный программный комплекс, предназначенный для полного цикла написания и тестирования программ на определенном языке» [13]. Среда разработки можно использовать для написания кода, редактирования исходного текста, работе с компиляторами и отладчиками, а также для выполнения других задач, связанных с разработкой программного обеспечения. Она объединяет в себе множество инструментов и функций, которые упрощают и ускоряют процесс разработки, а также позволяют дополнительно проверять код на ошибки и недочеты. В среде разработки можно настроить цветовую схему, шрифты, расстановку пробелов и другие параметры, чтобы сделать работу более удобной и комфортной.

Существует множество сред разработки для создания онлайн-сервисов, некоторые из них:

- Visual Studio Code
- Sublime Text
- Atom
- Brackets
- WebStorm
- Adobe Dreamweaver

- PHPStorm
- Eclipse
- NetBeans
- IntelliJ IDEA.

Кроме того, есть также некоторые онлайн-среды разработки, такие как Cloud9, CodePen, Repl.it и другие, которые позволяют разрабатывать, тестировать и запускать приложения без необходимости установки программного обеспечения на локальном компьютере.

В качестве средства разработки будем использовать Visual Studio Code так как среда разработки:

- Бесплатная: Visual Studio Code бесплатна и доступна для загрузки и использования без ограничений.
- Кроссплатформенная: Visual Studio Code доступна для Windows, Linux и macOS.
- Расширяемая: В Visual Studio Code есть возможность установки дополнительных расширений для улучшения функциональности.
- Интегрируема с Git: Visual Studio Code предоставляет нативную поддержку Git, что делает работу с этой системой контроля версий гораздо проще и удобнее.
- Редактирование кода: Редактор кода в Visual Studio Code мощный и автоматически подсвечивает синтаксис, что делает процесс написания кода более продуктивным.
- Отладка: Visual Studio Code предоставляет возможность отладки кода, что позволяет быстро находить и устранять ошибки.
- Поддержка различных языков программирования: Visual Studio Code поддерживает множество языков программирования, включая Python, Java, JavaScript, C++, PHP и другие.



### 3.3 Разработка страницы авторизации

Первым что будет видеть пользователь это странице авторизации, на ней студент должен будет ввести логин и пароль от образовательного портала ТГУ, если данные авторизации верны студент попадет в личный кабинет на главную страницу.

Для этого создадим HTML код. Где добавим название сервиса и составим форму для ввода логина и пароля, а также добавим кнопку для авторизации. Для определения внешнего вида и стиля страницы авторизации используем язык CSS. «Это язык таблиц стилей, который используется для описания внешнего вида и форматирования документа, написанного на языке разметки» [20]. Он позволяет устанавливать цвета, шрифты, размеры, расстояния между элементами, размещение объектов на странице, а также создавать анимации и эффекты перехода. HTML код страницы авторизации представлен ниже на рисунке 11.

```
<body>
  <p class="name">Перезачет внешнего курса / курса ДПО</p>
  <div class="form">
    <div class="title">Авторизация</div>
    <label id="email">
      <input type="text">
      <span>Login</span>
    </label>
    <label id="password">
      <input type="password">
      <span>Password</span>
    </label>
    <div class="button"><a href = "main-page.html">Войти</a></div>
  </div>
  <script src="scripts/log-in.js"></script>
</body>
```

Рисунок 11 – HTML код страницы авторизации

Итоговый вид страницы авторизации представлен на рисунке 12.

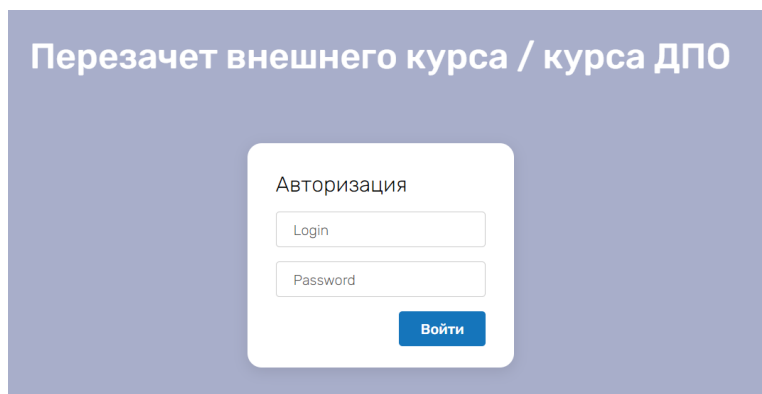


Рисунок 12 – Конечный вид страницы авторизации

Тольяттинский государственный университет для авторизации своих пользователей использует LDAP MS Active Directory.

«Active Directory (AD) – проприетарная реализация от компании Microsoft службы каталогов – совокупности программных сервисов и баз данных для иерархического представления информационных ресурсов в сети (компьютеров, принтеров, сетевых дисков и пр.) и настройки доступа к ним» [15]. Active Directory поддерживает протоколы LDAP, Kerberos и другие стандартные протоколы для обмена информацией между клиентами и серверами. Он является ключевым компонентом при создании и управлении сетью Windows.

Обмен информацией между клиентами и сервером в LDAP MS Active Directory происходит по протоколу LDAP (Lightweight Directory Access Protocol). Клиенты отправляют запросы на сервер Active Directory, используя стандартный набор LDAP-операций, таких как поиск, добавление, изменение и удаление записей в каталоге.

### **3.4 Разработка главной страницы**

На главной странице будет присутствовать информация о том для чего создан данный сервис и как им пользоваться. HTML код главной страницы представлен ниже на рисунке 13.

```
<body>
  <nav class="navbar">
    <div class="container">
      <a href="main-page.html" class="navbar-brand">Перезачет внешнего курса / курса ДПО</a>
      <div class="navbar-wrap">
        <ul class="navbar-menu">
          <li><a href="main-page.html">Главная страница</a></li>
          <li><a href="add-request.html">Подать запрос</a></li>
          <li><a href="my-request.html">Мои запросы</a></li>
        </ul>
        <h5 class="callback">Имя Фамилия</h5>
      </div>
    </div>
  </nav>
  <div class="info-inner">
    <h2 class="section-title">Добро пожаловать!</h2>
    <p class="info-text">
      Здесь вы можете подать запрос  перезачете открытых онлайн курсов.
      Также отследить ход работы уже поданных запросов.
    </p>
    <h3 class="info-title">Как работает сервис?</h3>
    <ol class="info-list">
      <li class="info-item">
        Вы нажимаете на кнопку "Подать запрос".  открывшейся форме пишете наименование учебного курса, по которому хотите
      </li>
      <li class="info-item">
        Информация  поданных вами запросах отображается в разделе "Мои запросы". Если в статусе подданной заявки указано
      </li>
    </ol>
  </div>
</body>
```

Рисунок 13 – HTML код главной страницы

Как и при разработке страницы авторизации для внешнего вида и стиля страницы использовали формальный язык CSS.

Итоговый вид главной страницы представлен на рисунке 14.

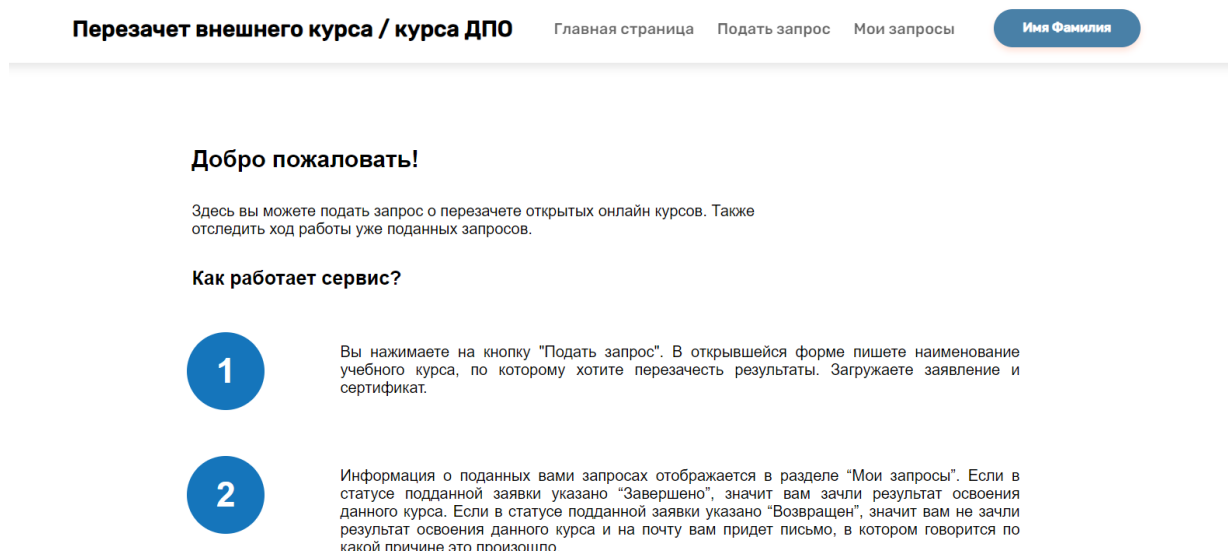


Рисунок 14 – Конечный вид главной страницы

В правом верхнем углу будем выводить имя и фамилия студента. На главной странице также будут присутствовать три кнопки “Главная страница”, “Подать запрос”, “Мои запросы”, первая для перехода на главную страницу, вторая кнопка для подачи запроса и третья для того чтобы просмотреть состояния уже поданных запросов.

### 3.5 Разработка страницы подачи запроса

На странице подачи запроса студент вводит наименование учебного курса, загружает заявление о перезачете и сертификат. Все эти данные должны отправиться сотруднику в Битрикс24. Для этого составим алгоритм работы. Алгоритм отправки запроса показан на рисунке 15.

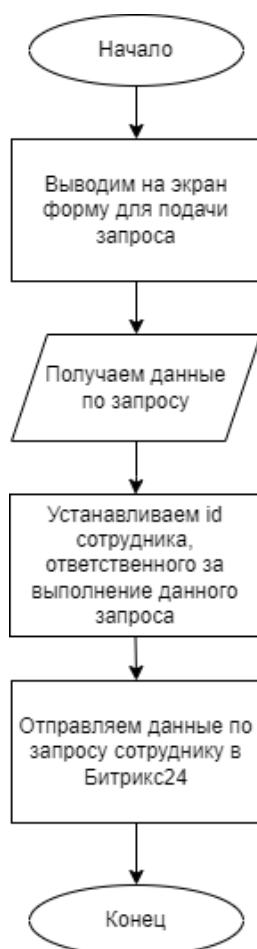


Рисунок 15 - Алгоритм отправки запроса

Для отправки данных в Битрикс24 необходимо использовать вызовы API (Application Programming Interface).

«Интерфейс прикладного программирования (API) - это интерфейс, который позволяет взаимодействовать двум программам» [17]. API обеспечивает доступ к определенной функциональности или данным, которые могут быть использованы для создания новых приложений, интеграции с другими сервисами или автоматизации определенных процессов.

Войдем в аккаунт Битрикс24. В меню сверху выберите "Настройки" и перейдите на вкладку "Интеграции" и выберем "REST API". Создайте новый вебхук с необходимыми правами доступа к модулям, которые хотим использовать. Для этого в настройке прав выберем пункт CRM, ведь именно туда и оттуда будем передавать, и получать данные. Полученные адрес вебхука для вызова rest api, будем использовать для отправки данных.

Вебхук (Webhook) — это механизм, который позволяет приложению отправлять автоматические уведомления о событиях другому программному обеспечению через HTTP-запросы. Вебхуки часто используются в веб-приложениях, чтобы отправлять уведомления о событиях, таких как изменения в базе данных, добавление новых пользователей и т.д. «При настройке webhook сгенерирует уникальный URL-адрес» [19]. «URL (Uniform Resource Locator) - это уникальный идентификатор, используемый для определения местоположения ресурса в Интернете» [18].

Для совершения HTTP-запросов будем использовать метод Fetch. Fetch - это API в JavaScript, который используется для выполнения запросов к серверу и получения данных в формате JSON, текста, массива байтов и других форматах. «Он является более современным и удобным способом работы с запросами, чем старый XMLHttpRequest» [16]. Fetch позволяет создавать асинхронные запросы, такие как GET, POST, PUT, DELETE и др., и обрабатывать результаты запроса в удобном формате при помощи promise.

На рисунке 16 представлена отправка запроса в формате JSON с применением Fetch API.

```

var requestData = {
  "fields": {
    "TITLE": "Перезачет внешнего курса / курса ДПО по курсу: " + name_course,
    "RESPONSIBLE_ID": 1,
    "UF_CRM_1686041295": {
      "fileData":
        ["application.docx", string_base64_1]
    },
    "UF_CRM_1686041344": {
      "fileData":
        ["certificate.pdf", string_base64_2]
    }
  }
};

fetch('https://b24-3jtsga.bitrix24.ru/rest/1/ucpnvcvdajw1ozud/crm.deal.add', {
  method: "POST",
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(requestData)
})
.then(res => res.json())
.then(data => console.log(data))
.catch(err => console.log(err));

```

Рисунок 16 – Отправки POST-запроса в формате JSON с применением Fetch API

Для создания новой сделки использовали функцию `crm.deal.add`.

«Параметр `fields` – это набор полей, содержащий значения полей запроса» [6]. Он содержит название “TITLE” к которому добавляется наименование курса, `id` сотрудника, ответственного за выполнение данного запроса, и два пользовательских поля с типом файл, для передачи заявления и сертификата, первым аргументом мы передаем название самого файла с расширением, вторым аргументом мы передаем сам файл в кодировке Base64.

Так как файлы попадают на портал с помощью вебхуков в кодировке Base64, напишем функцию для конвертирования файла в Base64. На рисунке 17 представлена функция для конвертирования в Base64.

```
const base64Encoder = file => new Promise((resolve, reject) => {
  const reader = new FileReader();
  reader.readAsDataURL(file);
  reader.onload = () => resolve(reader.result);
  reader.onerror = error => reject(error);
});
```

Рисунок 17 – Функция для конвертирования в Base64

Итоговый вид страница подачи запроса представление ниже на рисунке 18.

Перезачет внешнего курса / курса ДПО    Главная страница    Подать запрос    Мои запросы    [Имя Фамилия](#)

**Подать запрос**

Заявление

Сертификат

Рисунок 18 – Конечный вид страницы для подачи запроса

HTML код страницы для подачи запроса представлен ниже на рисунке 19.

```

<body>
  <nav class="navbar">
    <div class="container">
      <a href="main-page.html" class="navbar-brand">Перезачет внешнего курса / курса ДПО</a>

      <div class="navbar-wrap">
        <ul class="navbar-menu">
          <li><a href="main-page.html">Главная страница</a></li>
          <li><a href="add-request.html">Подать запрос</a></li>
          <li><a href="my-request.html">Мои запросы</a></li>
        </ul>
        <h5 class="callback">Имя Фамилия</h5>
      </div>
    </div>
  </nav>

  <div class="info-inner-2">
    <div class="form">
      <div class="title-2">Подать запрос</div>
      <label id="name-course">
        <input type="text">
        <span>Наименование учебного курса</span>
      </label>
      <label for="formFile" class="form-label"> Заявление </label>
      <input class="form-control-1" name="documen-1" type="file" id="file-1">

      <label for="formFile" class="form-label"> Сертификат </label>
      <input class="form-control-2" name="documen-2" type="file" id="file-2">

      <button class="button"><a href = "add-request.html">Отправить</a></div>
    </div>
  </div>
  <script src="scripts/add-req.js"></script>
</body>

```

Рисунок 19 – HTML код страницы для подачи запроса

На страницу мы добавим форму для подачи запроса, где студент сможет внести наименование учебного курса, а также загрузить два файла, заявление о перезачете и сертификат. Также добавим кнопку “Отправить” для отправки данных запроса в Битрикс24.

### 3.6 Разработка страницы “Мои запросы”

При переходе в раздел “Мои запросы” открывается таблица, где показан идентификатор уже поданного запроса, дата подачи, наименование учебного курса и статус запроса. Для этого составим алгоритм работы. Алгоритм получение данных о запросах показан на рисунке 20.





Рисунок 20 - Алгоритм получение данных о запросах

Для получения данных о запросах также, как и для отправки запроса будем использовать вызовы API (Application Programming Interface), адрес вебхука для получения данных и метод Fetch для совершения HTTP-запросов. На рисунке 21 представлено получение данных о запросах с применением Fetch API и запись данных в таблицу.

Для получения данных использовали функцию `crm.deal.list`. Данная функция возвращает список сделок, вместе со всей имеющейся информацией по каждой сделке.

```

fetch('https://b24-3jtsga.bitrix24.ru/rest/1/ucpnvcvdajw1ozud/crm.deal.list')
.then((data) => {
  return data.json();
}).then((objectData) => {
  let tableData="";
  objectData.result.map((values) => {
    tableData+= `<tr>
    <td>${values.ID}</td>
    <td>${values.DATE_CREATE}</td>
    <td>${values.TITLE}</td>
    <td>${values.STAGE_ID}</td>
    </tr>`;
  });
  document.getElementById("table_body").innerHTML=tableData;
})

```

Рисунок 21 – Получение данных о запросах с применением Fetch API и запись данных в таблицу

После получения данных мы перенесем эти данных в таблицу. Параметр values.ID содержит идентификатор запроса, параметр values.DATE\_CREATE содержит дату создания запроса, параметр values.TITLE наименование учебного курса, параметр values.STAGE\_ID содержит статус запроса.

### 3.7 Тестирование сервиса

Для тестирования онлайн сервиса будем использовать виртуальную машину VMBitrix. Исходя из требований, составленных во 2 главе, онлайн сервис должен отправлять запрос от студента на сервис Битрикс24, а также получать данные о ходе работы подданного запроса.

Отправим несколько запросов и проверим работает ли корректно сервис. На рисунке 22 показано как будет подаваться запрос.

После отправки данных в битрикс24 была создана сделка, где в названии указано наименование учебного курса, а в поля “Заявление о перезачете” и “Сертификат” добавленные соответствующие файлы. На рисунке 23 показана созданная сделка в битрикс24.

Подать запрос

Наименование учебного курса  
Учебный курс № 1

Заявление  
Выберите файл Заявление обуч... айн-курсов.docx

Сертификат  
Выберите файл сертификат.pdf

Отправить

Рисунок 22 – Отправка запроса

Перезачет внешнего курса / курса ДПО по курсу: Учебный курс № 1 Общая

РАСШИРЕНИЯ ☰ ☎ ✉ 💬 ⚙ ДОКУМЕНТ ▾ ПРЕДЛОЖЕНИЕ ▾

Новая ➤ Подготовка документов ➤ Счёт на предоплату ➤ В работе ➤ **Финальный счёт** ➤ Завершить сделку

Общие Товары Предложения Роботы Бизнес-процессы Счета Связи История Маркет Еще ▾

О СДЕЛКЕ ✎ AI24-скоринг изменить

Стадия  
Новая

Клиент  
не заполнено

Заявление о перезачете  
application.docx ( 12.61 КБ)

Сертификат  
certificate.pdf ( 355.51 КБ)

Выбрать поле Создать поле Удалить раздел

ДОПОЛНИТЕЛЬНО изменить

Дата начала  
11 июня 2023

Доступна для всех  
Нет

Ответственный  
👤 a.solomanidina@gmail.com

Дело Комментарий Задача Слоты СМС/WhatsApp Письмо Еще ▾

Что нужно сделать

👤+ Пригласить к обсуждению

Что нужно сделать

+ **Создайте дело**  
Запланируйте следующий шаг по сделке, чтобы не забыть о клиенте

Сегодня 🔍 ФИЛЬТР

📄 Создана сделка 17:48  
Перезачет внешнего курса / курса ДПО по курсу: Учебный курс № 1 📄

Рисунок 23 – Созданная сделка

После это мы переходим во вкладку “Мои запросы” и видим всю информацию по запросу в виде таблицы. На рисунке 24 показана вкладка “Мои запросы”, после подачи одного запроса.

Запрос №	Дата подачи	Наименование учебного курса	Статус
104	2023-06-11	Учебный курс № 1	В работе

Рисунок 24 – Вкладка “Мои запросы”

Если сотрудник завершит сделку, то статус запроса изменится на статус “Завершено”. На рисунке 25 показана вкладка “Мои запросы”, после завершения рассмотрения запроса.

Запрос №	Дата подачи	Наименование учебного курса	Статус
104	2023-06-11	Учебный курс № 1	Завершено

Рисунок 25 – Вкладка “Мои запросы” после завершения работы над запросом

Если студент, например, совершил ошибку в составлении заявления, то перезачесть данный курс студенту не могут, и во вкладке “Мои запросы”, в статус заявке будет указано “Возвращен” и студенту нужно будет подать запрос еще раз, но с уже корректными данными.

На рисунке 26 показана вкладка “Мои запросы”, после подачи двух запроса с разными результатами рассмотрения.

Запрос №	Дата подачи	Наименование учебного курса	Статус
104	2023-06-11	Учебный курс № 1	Завершено
106	2023-06-11	Учебный курс № 2	Возвращен

Рисунок 26 – Вкладка “Мои запросы” после завершения работы над двумя запросами

### Выводы по главе 3

В результате данной главы были описаны выбранные средства разработки и язык программирования, изучен программный интерфейс API, механизм вебхук, метод Fetch. Был разработан онлайн сервис. Тестирование показало, что онлайн сервис удовлетворяет всем поставленным требованиям.

## Заключение

В процессе выполнения выпускной квалификационной работы был проведен анализ предметной области. Анализ выявил проблему в отсутствии онлайн сервиса для перезачет внешнего курса / курса ДПО. Была выделена актуальность исследуемой темы, определены объект, предмет исследования, цели и задачи работы.

Была проведена разработка онлайн сервиса, в которую входило:

- исследовать предметную область и сформированы требования к создаваемому онлайн;
- спроектировать онлайн сервис, построили логическую модель разрабатываемого онлайн сервиса, разработали пользовательский интерфейс;
- выбрать средства разработки и язык программирования и разработать онлайн сервис и протестировать онлайн сервис.

Результатом выпускной квалификационной работы является работающий онлайн сервис для перезачета внешнего курса / курса ДПО который позволяет:

- подавать запрос на перезачет, загружая сертификат о прохождении онлайн-курса и заявление о перезачете онлайн-курсов;
- просматривать результат поданных студентом запросов.

Для того, чтобы проверить правильности логического и функционального моделирования, были построены: диаграмма вариантов использования и диаграмма последовательности. Для реализации онлайн сервиса был выбран язык программирования JavaScript, для разметки веб-страниц и оформления их визуального стиля соответственно языки HTML и CSS, среда разработки Visual Studio Code.

За время написания выпускной квалификационной работы мной были изучены различные аспекты, связанные с разработкой сервисов, а также применены полученные знания, что бесспорно будет полезно в последующей профессиональной деятельности.

## Список используемых источников

1. Автоматизированное проектирование пользовательских интерфейсов : сайт. - URL: <https://cyberleninka.ru/article/n/avtomatizirovannoe-proektirovanie-polzovatelskih-interfeysov> (дата обращения: 17.02.2023).
2. Битрикс24: Что такое Битрикс24? : сайт. - URL: <https://www.bitrix24.ru/whatisthis/> (дата обращения: 10.02.2023).
3. Высшее образование дистанционно | Росдистант – высшее образование в России дистанционно : сайт. - URL: <https://www.rosdistant.ru/about/> (дата обращения: 20.01.2023).
4. Гид по Фигме для начинающих веб-дизайнеров : сайт. - URL: <https://tilda.education/articles-figma> (дата обращения: 05.03.2023).
5. Диаграмма последовательности. Руководство с примерами. : сайт. - URL: [https://itonboard.ru/analysis/394-diagramma\\_posledovatelnosti\\_sequence\\_diagrams\\_uml/](https://itonboard.ru/analysis/394-diagramma_posledovatelnosti_sequence_diagrams_uml/) (дата обращения: 10.02.2023).
6. Документация по REST : сайт. - URL: [https://dev.1c-bitrix.ru/rest\\_help/crm/cdeals/crm\\_deal\\_add.php](https://dev.1c-bitrix.ru/rest_help/crm/cdeals/crm_deal_add.php) (дата обращения: 10.04.2023).
7. Какие бывают онлайн-курсы и как правильно выбрать обучающий курс в интернете? : сайт. - URL: <https://externat.foxford.ru/polezno-znat/kak-vybrat-onlajn-kursy#:~:text=Онлайн-курсы%20/> (дата обращения: 10.02.2023).
8. Какие бывают онлайн-сервисы : сайт. - URL: [https://metallicheckiportal.ru/articles/raznoe/servisi/kakie\\_bivayt\\_onlain-servisi/](https://metallicheckiportal.ru/articles/raznoe/servisi/kakie_bivayt_onlain-servisi/) (дата обращения: 15.01.2023).
9. Межвузовская площадка электронного образования | Универсариум : сайт. - URL: <https://universarium.org/> (дата обращения: 20.01.2023).
10. О нас – Stepik : сайт. - URL: <https://welcome.stepik.org/ru/about/> (дата обращения: 20.01.2023).

11. Построение диаграммы вариантов использования как средство описания типичных взаимодействий системы и пользователя на основе сервиса по сдаче помещений в аренду : сайт. - URL: <https://cyberleninka.ru/article/n/postroenie-diagrammy-variantov-ispolzovaniya-kak-sredstvo-opisaniya-tipichnyh-vzaimodeystviy-sistemy-i-polzovatelya-na-osnove-servisa/> (дата обращения: 15.03.2023).

12. Что такое онлайн сервис — виды, перспективы, опасности : сайт. - URL: <https://dmitriyzhilin.ru/chto-takoe-onlajn-servis-vidy-perspektivy-opasnosti/> (дата обращения: 20.01.2023).

13. Что такое среда разработки (IDE) : сайт. - URL: <https://www.bestfree.ru/glossary/rus-s/sreda-razrabotki.php> (дата обращения: 10.04.2023).

14. Что такое функциональные требования: примеры, определение, полное руководство : сайт. - URL: <https://visuresolutions.com/ru/blog/functional-requirements/#:~:text=Если%20функциональные%20требования%20определяют%2С%20что,Это%20часть%20функциональных%20требований.> (дата обращения: 19.02.2023).

15. Active Directory/LDAP : сайт. - URL: [https://trueconf.ru/blog/wiki/active-directory-ldap#:~:text=Active%20Directory%20\(AD\)%20—%20проприетарная,и%20настройки%20доступа%20к%20ним.](https://trueconf.ru/blog/wiki/active-directory-ldap#:~:text=Active%20Directory%20(AD)%20—%20проприетарная,и%20настройки%20доступа%20к%20ним.) (дата обращения: 10.03.2023).

16. Using the Fetch API : сайт. - URL: [https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch) (дата обращения: 18.03.2023).

17. What is an API? (Application Programming Interface) : сайт. - URL: <https://www.techtarget.com/searcharchitecture/definition/application-program-interface-API> (дата обращения: 10.04.2023).

18. What is a URL? (Uniform Resource Locator) : сайт. - URL: <https://www.techtarget.com/searchnetworking/definition/URL> (дата обращения: 12.03.2023).



19. What is a webhook? : сайт. - URL: <https://www.redhat.com/en/topics/automation/what-is-a-webhook> (дата обращения: 20.03.2023).

20. What is CSS : сайт. - URL: <https://www.javatpoint.com/what-is-css> (дата обращения: 15.03.2023).