**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»**

**(ВлГУ)**

**Колледж инновационных технологий и предпринимательства**

КАФЕДРА ФИЗИКИ И ПРИКЛАДНОЙ МАТЕМАТИКИ

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ**

**ПО ДИСЦИПЛИНЕ «Технология разработки программного обеспечения»**

**на тему «Разработка веб-сервиса распознавания математических формул»**

Студента Башилова Максима Андреевича

Специальность: 09.02.03 "Программирование в компьютерных системах"

Группы ПКсп-120

*Руководитель*: доцент каф. ФиПМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Лексин А.Ю.

Владимир 2023

ВЛАДИМИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Зав. кафедрой

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

ЗАДАНИЕ

НА КУРСОВУЮ РАБОТУ

по дисциплине «Технология разработки программного обеспечения»

Выдано студенту Башилову Максиму Андреевичу группы ПКсп-120

Специальность подготовки СПО 09.02.03 «Программирование в компьютерных системах»

1. Тема работы «Разработка веб-сервиса распознавания математических формул»

2. Срок сдачи законченной работы 21.12.2023

3. Исходные данные к курсовой работе постановка задачи.

4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов)

Введение.

1. Анализ предметной области.
2. Постановка задачи
3. Проектирование приложения.
4. Разработка приложения.
5. Тестирование.

Заключение.

5. Постановка задачи

ЧАСТЬ 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1. Выполнить описание предметной области.
2. На основании описания предметной области провести обзор существующих аналогов.

ЧАСТЬ 2. ПОСТАНОВКА ЗАДАЧИ

1. Определить требования и задачи для достижения поставленной цели.
2. Определить минимальные требования для работы программы.

ЧАСТЬ 3. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

1. На основании описания предметной области выполнить разработку ER-модели, реляционной модели данных.
2. Составить словарь данных, в котором представлены все поля таблиц базы данных с указанием ключа, наименования, типа, обязательности заполнения и заметок.
3. Представить диаграмму прецедентов, диаграммы активностей.
4. Расписать основные методы, используемые в проектировании интерфейса.

ЧАСТЬ 4. РАЗРАБОТКА ПРИЛОЖЕНИЯ

1. Описать выбор инструментальных средств разработки приложения.
2. На основании ER-диаграммы базы данных, описать физическую реализацию базы данных. Если в работе с базой данных используются представления, то описать подробно каждое из них.
3. Описать все классы приложения с пояснением для каких целей они предназначены и методы с описанием: назначения, входных значений и выходных значений каждого метода.
4. Кратко описать структуру и компоненты, используемые при разработке интерфейса.

ЧАСТЬ 5. ТЕСТИРОВАНИЕ

1. Составить тест-требования, определяющие, что должно быть протестировано, но не определяющие, как это должно быть сделано.
2. По тест-требованиям разработать тест-план, в каждом тестовом примере которого обязательно перечислены все входные значения и ожидаемые выходные значения, а также сценарий, описывающий последовательность действий, которые необходимо выполнить для выполнения тестового примера.

Дата выдачи задания 05.10.2023

Руководитель Лексин А.Ю.

Задание принял к исполнению Башилов М.А.

содержание

[ВВЕДЕНИЕ 4](#_Toc153472147)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc153472148)

[1.1 Описание предметной области 5](#_Toc153472149)

[1.2. Анализ существующих аналогов 7](#_Toc153472150)

[1.2.1 Веб-сервис Google Docs 8](#_Toc153472151)

[1.2.2 Веб-сервис Detexify 10](#_Toc153472152)

[1.2.3 Веб-сервис Free Online OCR 10](#_Toc153472153)

[1.2.4 Платная программа Adobe Acrobat Pro DC 12](#_Toc153472154)

[1.2.5 Программа OmniPage Ultimate 13](#_Toc153472155)

[1.2.6 Программа ABBYY FineReader PDF 14](#_Toc153472156)

[1.2.7 Сравнительный анализ систем искусственного интеллекта 15](#_Toc153472157)

[2. ПОСТАНОВКА ЗАДАЧИ 18](#_Toc153472158)

[3. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ 19](#_Toc153472159)

[3.1 ER-диаграмма 19](#_Toc153472160)

[3.2 Словарь данных 20](#_Toc153472161)

[3.3 Диаграмма вариантов использования 21](#_Toc153472162)

[3.4 Диаграммы деятельности 23](#_Toc153472163)

[3.5 Проектирование макетов пользовательского интерфейса 27](#_Toc153472164)

[4. РАЗРАБОТКА ПРИЛОЖЕНИЯ 32](#_Toc153472165)

[4.1 Описание инструментальных средств разработки программного обеспечения 32](#_Toc153472166)

[4.2 Альтернативы и преимущества инструментальных средств разработки программного обеспечения 34](#_Toc153472167)

[4.3 Создание и обслуживание базы данных 36](#_Toc153472168)

[4.4 Решение административных функций в базе данных 36](#_Toc153472169)

[4.5 Разработка веб-сервиса 38](#_Toc153472170)

[5. ТЕСТИРОВАНИЕ 49](#_Toc153472171)

[5.1 Разработка тест-требований 49](#_Toc153472172)

[5.2 Разработка тест-плана 51](#_Toc153472173)

[Заключение 59](#_Toc153472174)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 60](#_Toc153472175)

[ПРИЛОЖЕНИЕ А 62](#_Toc153472176)

[ПРИЛОЖЕНИЕ Б 63](#_Toc153472177)

[ПРИЛОЖЕНИЕ В 64](#_Toc153472178)

# ВВЕДЕНИЕ

С учётом развития направления нейронных сетей в информационном окружении, тема распознавания математических формул достаточно актуальна и интересна, программное обеспечение систем распознавания или определения некоторых символьных наборов широко распространяется в общественных организациях, малых и больших бизнесах, в научных и учебных организациях.

Целью курсовой работы является проектирование и разработка веб-сервиса для распознавания математических формул, который с помощью обработки исходного изображения, содержащего математическую формулу или выражение преобразовывает его в форматы LaTeX, MathML и JSON.

В рамках курсовой работы будут определены основные проблемы существующих систем распознавания рукописных и печатных символов, разработаны функциональные требования к веб-сервису для распознавания математических формул, спроектирована структура базы данных и проведено тестирование разработанной системы.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Описание предметной области

В компьютерных технологиях выделяют отрасль искусственного интеллекта, в которой каждый тип системы определяет собой направление, в котором решаются определенные наборы задач.

Типами систем искусственного интеллекта являются:

* Логические выводы с использованием нечетких множеств - тип системы подразумевает логические выводы на основе заранее определенного понимания бытовых, повседневных, технических, логистических задач, а также помогает решить задачи характера, когда нужно отталкиваться от временной ситуации и окружающего момента. Примером могут служить системы умного дома, которые в зависимости от окружающего состояния, меняют настройки характеристик в доме.
* Машинное обучение - тип систем в большей степени направлен на то, чтобы машина с заранее заданным набором данных самостоятельно обучалась посредством решения нужных человеку задач.
* Эвристический поиск - тип систем направлен на решение задач путём перебора всех возможных вариантов решения задачи и выбором самого оптимального варианта, при котором конечное состояние системы, в которой решается задача, стане лучше, в сравнении с другими ответами. Примером может быть анализ хорошего хода в любой настольной игре.

Самым распространенным примером использования искусственного интеллекта является оптическое распознавание символов (англ. Optical Character Recognition - OCR). OCR – это особая система, способная проводить операции над образами рукописных символов, отсканированных файлов, изображений, фрагментов видео в текст, понятный машине и преобразованный для восприятия человеком. Данная технология применяется во многих отраслях жизни человека. В качестве примеров можно привести распознавание документов, поиск данных в потоке информации.

Для решения задачи оптического распознавания символов применяются искусственные нейронные сети. Искусственная нейронная сеть - это громадный распределенный параллельный процессор, состоящий из элементарных единиц обработки информации, накапливающих экспериментальные знания и предоставляющий их для последующей обработки.

Для получения необходимых выходных результатов при заданных входных данных искусственная нейронная сеть проходит процедуру обучения. Машинное обучение – подмножество искусственного интеллекта, подразумевающее создание машины, способной самообучаться и находить закономерности в данных, представляемых ей для обучения. Основной принцип машинного обучения – машины получают данные и на их основе обучаются. Такие системы позволяют быстро применять знания, полученные при обучении на больших наборах данных [1][2].

Разрабатываемое программное обеспечение предназначено для решения задачи распознавания математических формул и использует технологию OCR и машинное обучение.

Математическая формула - это всякая символическая запись в виде выражения, равенства или неравенства, содержащая какую-либо информацию. С помощью математической формулы довольно сложные предложения могут быть записаны в компактной и удобной форме.

Веб-сервис использует обученную модель нейронной сети для анализа входных данных и точного определения математических символов и формул. Он включает в себя алгоритмы машинного обучения и методы распознавания образов для интерпретации сложных математических выражений и преобразования их в цифровой формат, который можно в дальнейшем обрабатывать или манипулировать.

Пользователи веб-сервиса могут взаимодействовать с ним через веб-интерфейс, загружая изображение.

Входные данные могут быть представлены в виде изображения, содержащего текст с включенной в него математической формулой либо непосредственное изображение с математической формулой или выражением.

В качестве результата работы веб-сервис преобразует распознанную формулу или выражение в формат MathML и представляет ее пользователю.

Основная цель веб-сервиса - обеспечить точное и эффективное распознавание математических формул в тексте, тем самым сокращая ручные усилия, необходимые для расшифровки и интерпретации печатных математических формул. Он позволяет пользователям легко преобразовывать аналоговые математические выражения в формат MathML или в печатный текст.

В целом, веб-сервис распознавания математических формул с помощью нейронной сети решает задачу автоматизации и добавляет удобство в работу с математическими выражениями.

## 1.2. Анализ существующих аналогов

Программные продукты в области компьютерного зрения и искусственного интеллекта очень востребованы в разработке и реализации, так как способы их применения затрагивают множество задач, связанных с визуальными источниками информации.

Распознавание текста является одной из областей в программной реализации искусственного интеллекта в области компьютерного зрения. Большое количество языков, надписей систематического назначения, математических выражений и формул пользуются спросом в реализации. Математические формулы и выражения считаются достаточно сложной тематикой в компьютерном зрении, так как нужно распознать не только отдельные символы, но и саму структуру математической формулы.

В качестве бесплатных аналогов выделены следующие системы:

* Google Docs – это бесплатный онлайн-офис, разрабатываемый компанией Google для создания документов в браузере [3].
* Detexify – это – это веб-сервис для распознавания единичных математических символов и преобразования их в формат LaTeX [4].
* Free Online OCR – это веб-сервис для распознавания текста по изображению [5]. Поддерживает 106 языков и распознает текст с большинства самых популярных форматов файлов.

В качестве платных аналогов выделены следующие системы:

* Adobe Acrobat Pro DC – это программное обеспечение, позволяющее конвертировать файлы PDF в Word, Excel и другие форматы [6].
* OmniPage Ultimate – это одна из самых популярных OCR-программ, является мощной и простой в использовании программой, которая может сделать хорошие предположения о том, какой текст может быть в отсканированном документе [7].
* ABBYY FineReader PDF – это мощная и простая в использовании программа, которая может угадать, какой текст может быть в отсканированном документе [8].

### 1.2.1 Веб-сервис Google Docs

Google Docs – это веб-ориентированное программное обеспечение, то есть программа, работающая в рамках веб-браузера без установки на компьютер пользователя.

Google Docs имеют в своем составе OCR-систему по распознаванию и поиску отдельных символов, нарисованных в реальном времени. Интерфейс сервиса представлен на рисунке 1.

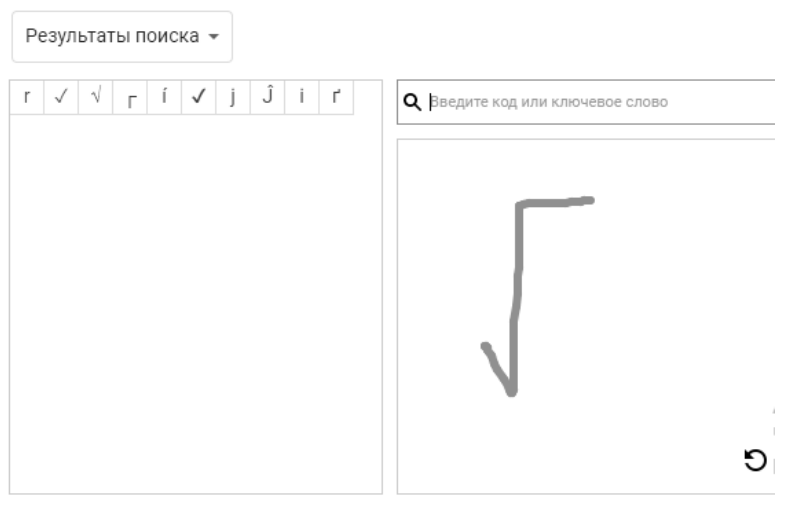


Рисунок 1 – Веб-сервис по распознаванию символов Google Docs

Документы и таблицы, создаваемые пользователем, сохраняются на специальном сервере Google, или могут быть экспортированы в файл. Это одно из ключевых преимуществ программы, так как доступ к введённым данным может осуществляться с любого компьютера, подключенного к интернету (при этом доступ защищён паролем).

В качестве преимуществ системы можно выделить:

* Бесплатный доступ к функционалу.
* Хранение информации в облаке.
* Возможность простого обмена файлами через отправления ссылок.
* Организация коллективной работы пользователей.
* Возможность редактировать историю изменений.
* Универсальность — ОС не имеет значения.

В качестве недостатков системы можно выделить:

* Работает медленнее. Особенно с объемными текстами при низкой скорости интернета.
* Зависит от аккаунта. Все документы хранятся на облаке, которое привязано к учетной записи.
* Зависит от разработчика.
* Зависит от интернета.

### 1.2.2 Веб-сервис Detexify

Detexify – это веб-сервис для распознавания единичных математических символов и преобразования их в формат LaTeX. Интерфейс сервиса представлен на рисунке 2.

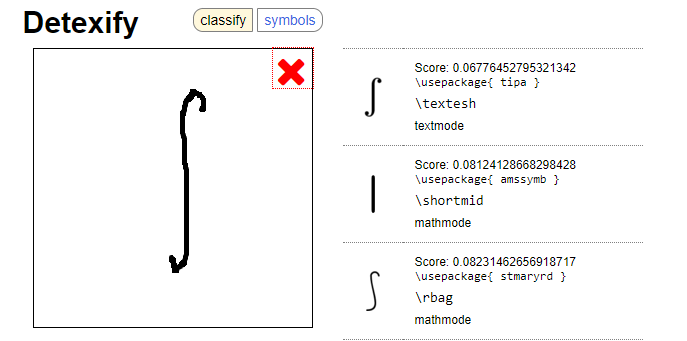


Рисунок 2 – Веб-сервис для распознавания единичных математических символов Detexify

В качестве преимущества сервиса можно выделить обширную базу символов для распознавания.

Недостатком системы является низкая точность распознавания символов в ряде случаев. Также в сервис нельзя загрузить изображения для распознавания содержимого на нем.

### 1.2.3 Веб-сервис Free Online OCR

Free Online OCR поддерживает 106 языков и распознает текст с большинства популярных форматов файлов: JPEG, PNG, GIF, BMP, TIFF, PDF, DjVu.

Имеет функционал для сохранения готовых документов не только в PDF, но и в стандартных форматах .doc и .txt. Кроме текста, может распознать математические уравнения, правильно форматировать текст в колонках и столбцах или обработать только выделенный фрагмент.

Интерфейс веб-сервиса представлен на рисунке 3.

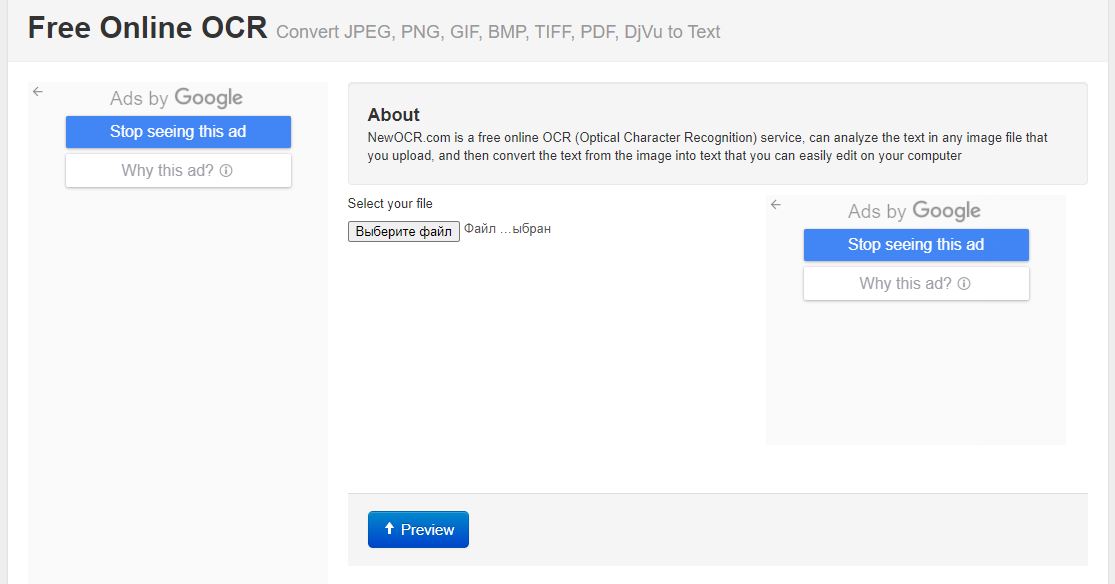


Рисунок 3 – Веб-сервис для распознавания символов Free Online OCR

Качество распознавания довольно высокое даже c картинок низкого качества.

В качестве преимуществ сервиса можно выделить:

* Удобство и простота.
* Нет лимитов по размеру исходников.
* Поддержка русского языка.
* Большой выбор форматов.

В качестве недостатка сервиса можно выделить долгую обработку файла при большом его размере.

### 1.2.4 Платная программа Adobe Acrobat Pro DC

Adobe Acrobat Pro DC – это программное обеспечение, позволяющее конвертировать файлы PDF в Word, Excel и другие форматы.

Он также имеет функцию OCR для преобразования отсканированных документов в редактируемый текст, что позволяет копировать/вставлять преобразованные тексты непосредственно в Office 365 или Google Docs.

Интерфейс программы представлен на рисунке 4.

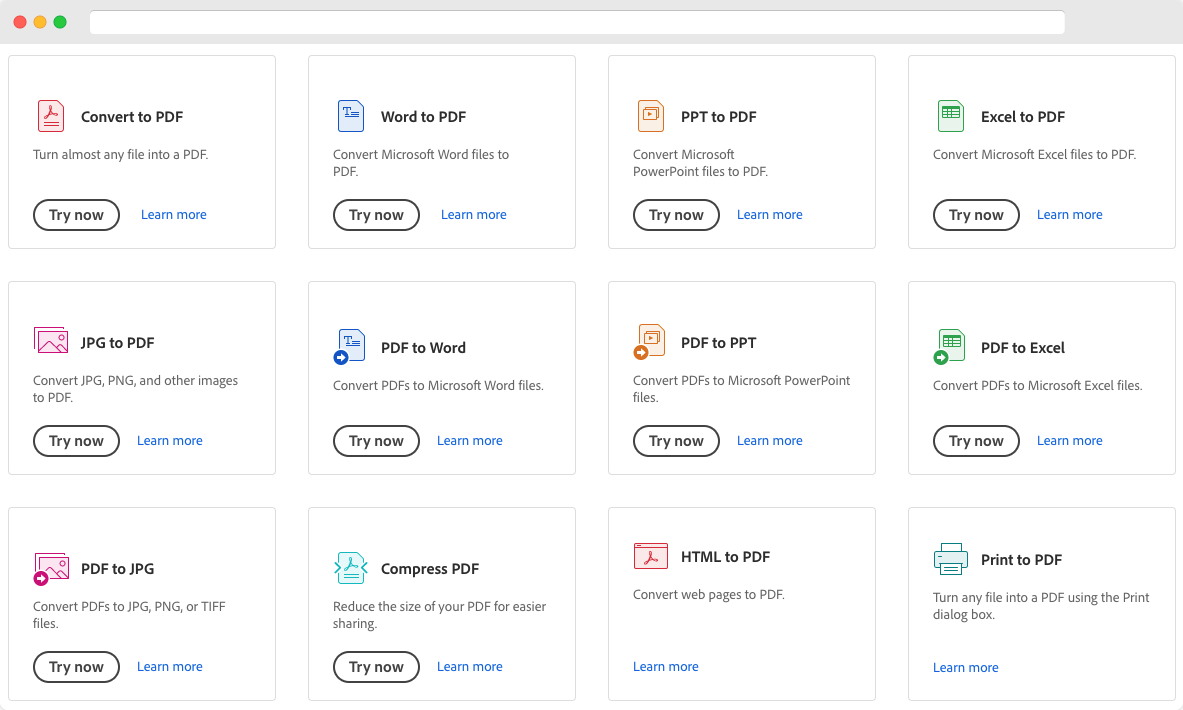


Рисунок 4 – Интерфейс программы Adobe Acrobat Pro DC

Adobe Acrobat Pro DC – не самый удобный вариант, поскольку это настольное приложение, но его функциональность компенсирует это.

В качестве преимуществ программного обеспечения можно выделить:

* Простой в использовании.
* Имеется возможность редактирования текста и изображений.
* Функционал для изменения порядка и удаление страниц в PDF.
* Легко конвертирует другие форматы файлов в PDF.
* Хороший инструмент для совместной работы.
* Возможность добавить безопасность и защитить PDF-файлы паролем.
* Функция подписи документов в цифровом формате своей собственной подписью.
* Удобные вкладки рабочего процесса.
* Простой пользовательский интерфейс.

В качестве недостатков программного обеспечения можно выделить:

* Функция редактирования текста неудобна.
* Модель подписки Adobe может отталкивать.
* Настольное приложение.

### 1.2.5 Программа OmniPage Ultimate

OmniPage Ultimate одна из самых популярных OCR, является мощной и простой в использовании программой, которая может сделать хорошие предположения о том, какой текст может быть в отсканированном документе.

Она также включает функции для редактирования документов PDF путем устранения ошибок печати. Программа поставляется с тремя лицензиями для отдельных компьютеров или устройств.

Интерфейс программы представлен на рисунке 5.

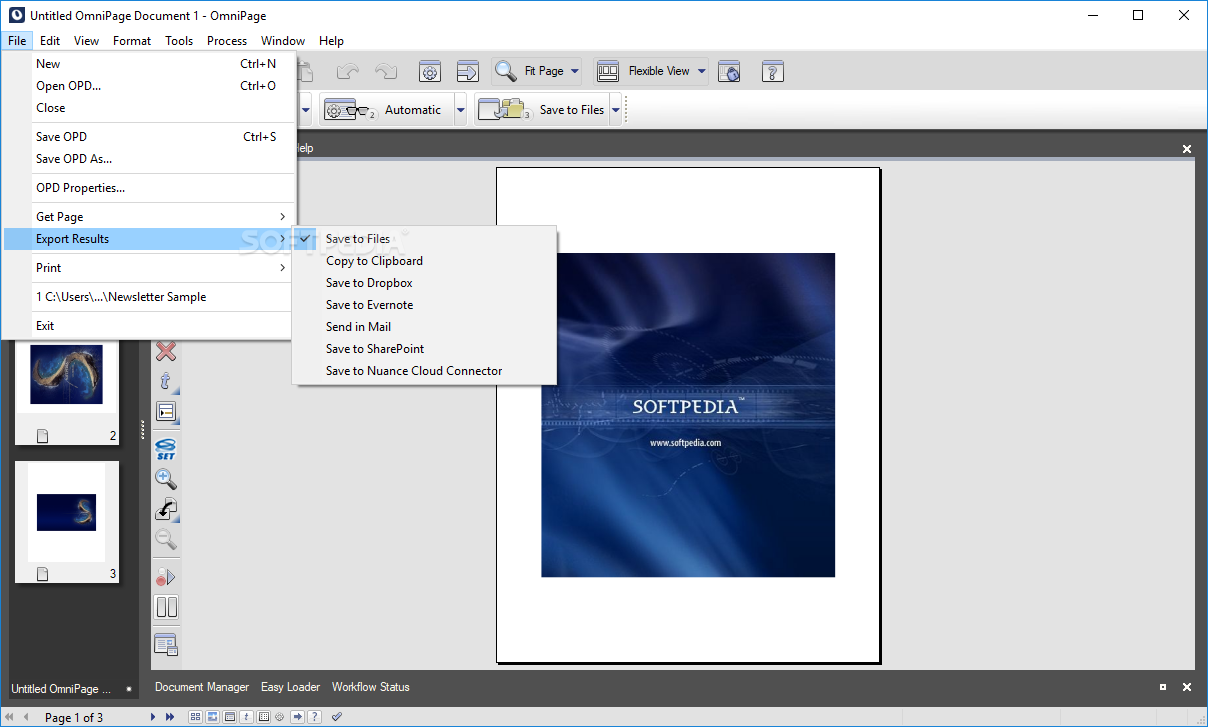


Рисунок 5 – Интерфейс программы OmniPage Ultimate

Программное обеспечение также поддерживает более 100 языков, включая французский, немецкий и испанский, и это лишь несколько примеров.

В качестве преимуществ программного обеспечения можно выделить:

* Дополнительные языки.
* Бесплатная пробная версия.
* Поддерживает более 100 языков.
* Простой в использовании.
* Устранение ошибок печати.
* Возможность сканирования и создания заполняемых документов.

В качестве недостатков программного обеспечения можно выделить интуитивно не понятный пользовательский интерфейс и большую цену за лицензию.

### 1.2.6 Программа ABBYY FineReader PDF

ABBYY FineReader – это мощная и простая в использовании программа, которая может угадать, какой текст может быть в отсканированном документе.

Интерфейс программы представлен на рисунке 6.

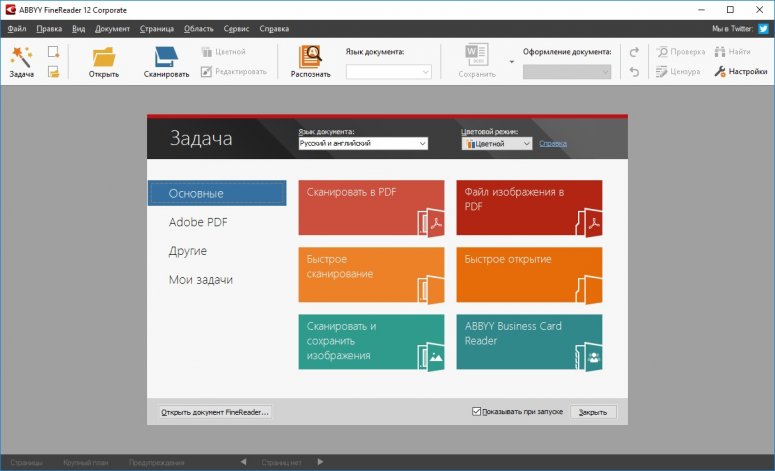


Рисунок 6 – Интерфейс программы ABBYY FineReader

Механизм основан на последних научных достижениях в области распознавания образов, компьютерного зрения и искусственного интеллекта. Он был разработан с использованием обширного набора данных отсканированных документов из различных источников: книг, журналов, дневников и тому подобных.

В качестве преимуществ программы можно выделить:

* Редактор для ручных исправлений.
* Простой интерфейс.
* Экспорт в различные форматы.
* Функция сравнения документов.
* Простой в использовании.

В качестве недостатка программы можно выделить отсутствие полнотекстового индексирования.

### 1.2.7 Сравнительный анализ систем искусственного интеллекта

Для сравнения систем искусственного интеллекта были выдвинуты следующие задачи:

* Возможность распознавания нескольких символов на изображении.
* Возможность распознавания нелинейных выражений.
* Доступность.
* Offline-реализации.
* Понятность и удобство интерфейса.

В качестве условных обозначений были выбраны следующие символы:

* «+» – реализация удовлетворяет заданному критерию.
* «–» – реализация не удовлетворяет заданному критерию.

Сравнение систем искусственного интеллекта по распознаванию изображений представлено в таблице 1.

Таблица 1 – Сравнение систем искусственного интеллекта

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название системы | Возможность распознавания нескольких символов | Возможность распознавания нелинейных выражений | Доступность | Offline-реализация | Понятность и удобство интерфейса |
| Google Docs | – | – | + | – | + |
| Detexify | – | – | + | – | + |
| OmniPage | + | – | – | + | – |
| Free Online OCR | + | – | + | – | – |
| Adobe Acrobat Pro DC | + | – | – | + | + |
| ABBYY FineReader PDF | – | + | – | + | + |

После сравнения программных продуктов был проведен анализ и сделаны несколько выводов.

Половина приложений в настоящее время представлены в формате онлайн, то есть в виде веб-сервисов. Такой формат реализации требует подключения к интернету и не может функционировать автономно.

Рассмотрев все представленные приложения, можно уверенно заявить, что три из пяти из них доступны для общественного пользования и предоставляются бесплатно.

Возможность распознавать множество символов отмечена в трех системах. Сложность некоторых распознаваемых формул и выражений весьма велика для получения корректного результата.

Удобность интерфейса и функционал в рассмотренных сервисах и приложениях достаточно просты и не выделяются. Весь функционал в основном распределяется на распознавание нарисованного символа или же распознавание по готовому изображению.

Сравнивая программные продукты, представленные в данном сравнительном анализе, можно сделать вывод, что задача распознавания математических формул является непростой и по-прежнему актуальна.

# 2. ПОСТАНОВКА ЗАДАЧИ

Цель работы – спроектировать и реализовать веб-сервис для распознавания математических формул, который будет иметь функционал для автоматической обработки изображения и предоставления результата распознавания в форматах LaTeX, MathML и JSON. Для достижения поставленной цели были выдвинуты следующие задачи:

* изучить существующие системы и технологии распознавания объектов, рукописных и печатных символов;
* проанализировать требования системы к формату исходного изображения для распознавания;
* определить структуру базы данных для веб-сервиса;
* реализовать и протестировать веб-сервис;
* провести анализ результатов тестирования.

При разработке были выделены минимальные требования для работы программы:

* загрузка исходного изображения для распознавания математической формулы;
* просмотр истории распознаваний для зарегистрированных пользователей;
* регистрация и авторизация пользователей;
* отображение результата распознавания в форматах LaTeX, MathML и JSON.

Эти задачи помогут определить, что требуется для разработки веб-сервиса и осуществления его успешной эксплуатации.

# 3. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

## 3.1 ER-диаграмма

ER-диаграмма (ER - Entity-Relationship – сущность-связь) представляет собой физическую модель взаимосвязей между объектами. Она позволяет визуализировать то, как объекты связаны друг с другом. Диаграмма представлена на рисунке 7.

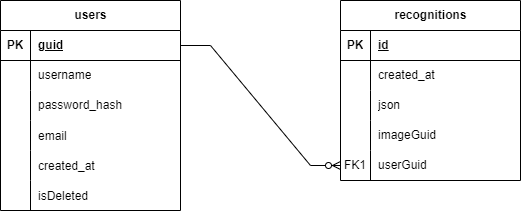


Рисунок 7 – ER-Диаграмма

На диаграмме отображено, какие связи установлены для таблиц и соответствующие им атрибуты. Между таблицами организована связь «один-ко-многим».

## 3.2 Словарь данных

Таблица 2 – Сравнение систем искусственного интеллекта

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Users (пользователи) | | | | | |
| Ключ | Поле | Перевод | Тип | Обязательное | Примечание |
| Первичный ключ | Guid |  | Nvarchar(128) | Да | Идентификационный номер пользователя |
|  | Username | Имя пользователя | Varchar (100) | Да | Имя пользователя |
|  | Password\_hash | Хэш пароля | Nvarchar(128) | Да | Пароль пользователя в виде хэша |
|  | Email | Адрес электронной почты | Varchar (100) | Да | Адрес электронной почты пользователя |
|  | Created\_at | Создано | Datetime | Да | Дата и время создания аккаунта |
|  | isDeleted | Удален | Boolean | Да | Показывает, удален ли аккаунт |
| Recognitions (распознавания) | | | | | |
| Первичный ключ | id |  | Bigint | Да | Идентификационный номер строки таблицы |
|  | Json |  | Varchar(MAX) | Нет | Представление распознанной формулы в формате json |
|  | imageGuid | Уникальный идентификатор изображения | Nvarchar(128) | Да | Уникальный идентификатор изображения |

Продолжение таблицы 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Внешний ключ | userGuid | Уникальный идентификатор пользователя | Nvarchar(128) | Да | Внешний ключ на таблицу Users |

Таблица Users (пользователи) базы данных имеет следующие атрибуты: guid - уникальный идентификатор пользователя, который представляет собой строку, состоящую из 38 символов, username – представляет собой имя пользователя, password\_hash – хэш пароля пользователя, email – электронная почта пользователя, isDeleted – показывает, удален ли аккаунт и created\_at – дата и время создания аккаунта.

Таблица Recognitions (распознавания) имеет следующий набор атрибутов: id – уникальный идентификатор строки, json – представление математической формулы в формате json, imageGuid – уникальный идентификатор загруженной картинки и userGuid – уникальный идентификатор пользователя, внешний ключ на таблицу Users (пользователи).

## 3.3 Диаграмма вариантов использования

Для описания функционала разрабатываемого веб-сервиса была создана диаграмма прецедентов, показанная на рисунке 8.

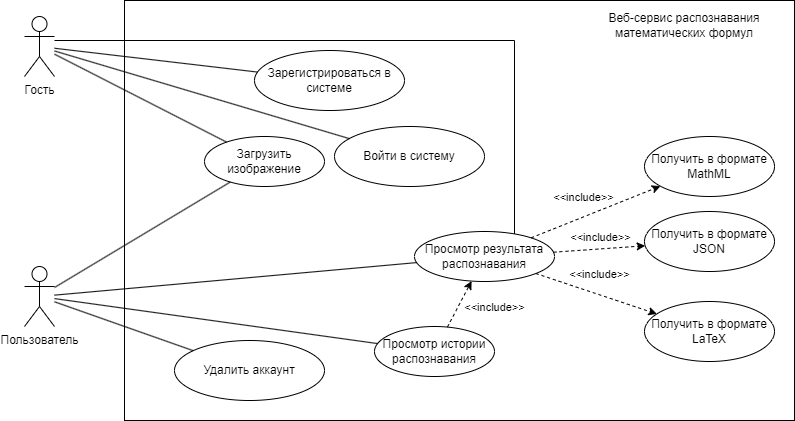


Рисунок 8 – Диаграмма вариантов использования

На диаграмме можно наблюдать, что в программе существует две роли: гость и пользователь.

Пользователь имеет полный доступ к веб-сервису, который выражается в следующих прецедентах:

* «Загрузить изображение» – пользователь может загрузить изображение в систему для последующего распознавания.
* «Просмотр результата распознавания» – пользователь может просмотреть результат распознавания загруженного изображения.
* «Просмотр истории распознавания» – пользователь может просмотреть историю распознаваний изображений.
* «Удалить аккаунт» - пользователь может удалить свой аккаунт.

Прецедент «просмотр результата распознавания» имеет отношение включения с прецедентами: «получить в формате MathML», «получить в формате LaTeX» и «получить в формате JSON», что означает, что пользователь имеет возможность получить результат распознавания в приведенных на диаграмме форматах.

Гость имеет ограниченный доступ к веб-сервису и имеет следующие прецеденты:

* «Зарегистрироваться в системе» - гость может пройти процедуру регистрации и перейти в роль пользователя.
* «Войти в систему» - гость может пройти процедуру авторизации и перейти в роль пользователя.
* «Загрузить изображение» - гость может загрузить изображение для распознавания.
* «Просмотреть результат распознавания» - гость может просмотреть результат распознавания загруженного изображения.

Ограниченность роли гостя выражается в отсутствии возможности просмотра истории запросов распознавания, но он не имеет ограничений на распознавание текущего загруженного изображения и так же как пользователь может получить результат распознавания математической формулы в форматах, представленных в диаграмме.

## 3.4 Диаграммы деятельности

Диаграмма деятельности, которая относится к прецеденту «зарегистрироваться в системе», представлена на рисунке 9.

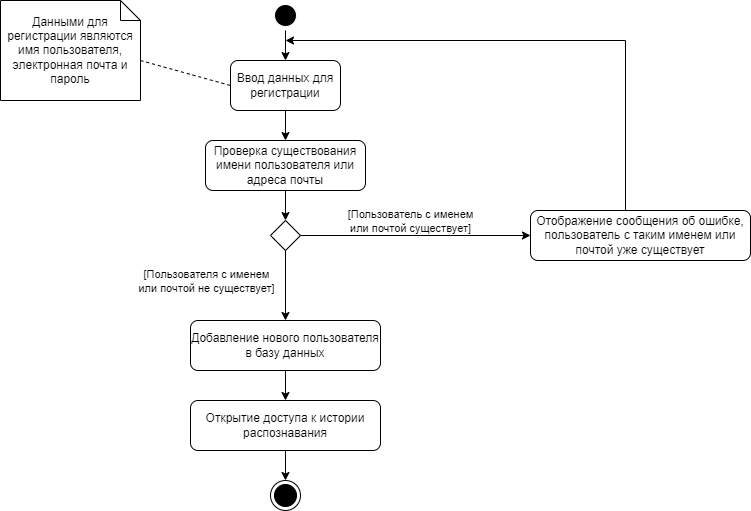


Рисунок 9 – Диаграмма деятельности для прецедента «зарегистрироваться в системе»

Диаграмма деятельности, которая относится к прецеденту «войти в систему», представлена на рисунке 10.

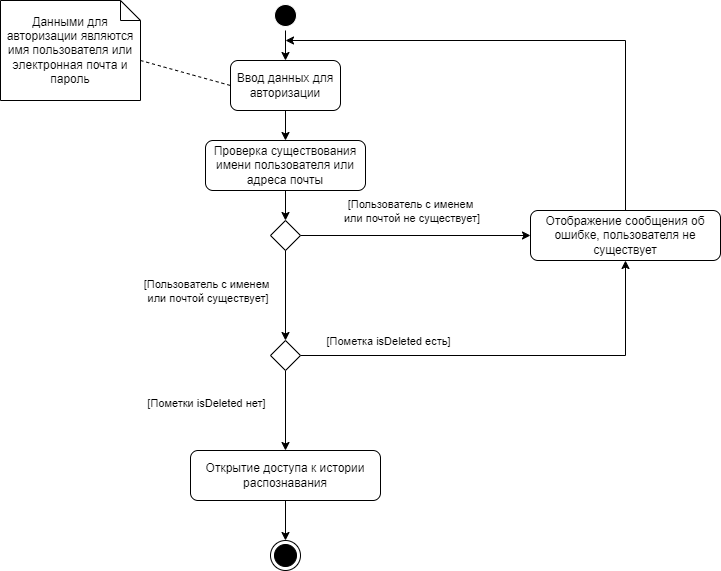


Рисунок 10– Диаграмма деятельности для прецедента «войти в систему»

Диаграмма деятельности, которая относится к прецеденту «загрузить изображение», представлена на рисунке 11.

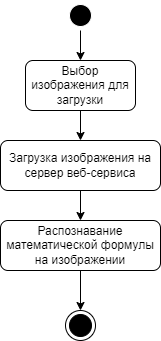


Рисунок 11 – Диаграмма деятельности для прецедента «загрузить изображение»

Диаграмма деятельности, которая относится к прецеденту «просмотреть результат распознавания», представлена на рисунке 12.

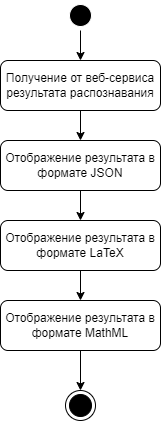


Рисунок 12 – Диаграмма деятельности для прецедента «просмотреть результат распознавания»

Диаграмма деятельности, которая относится к прецеденту «просмотреть историю распознавания», представлена на рисунке 13.

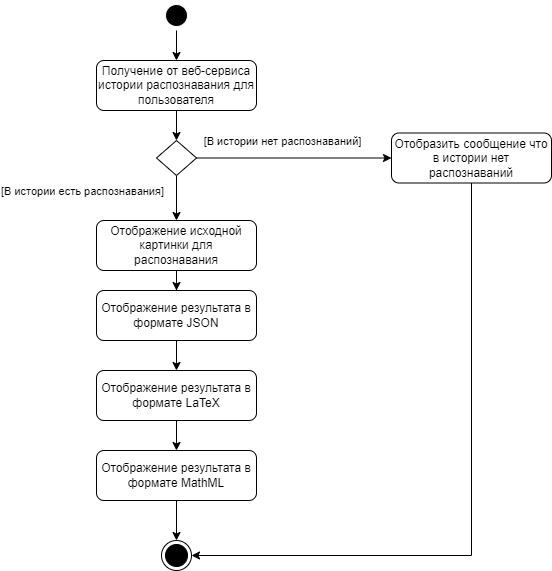


Рисунок 13 – Диаграмма вариантов деятельности «Завершить заказ»

Диаграмма деятельности, которая относится к прецеденту «удалить аккаунт», представлена на рисунке 14.

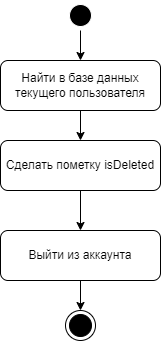


Рисунок 14 – Диаграмма деятельности для прецедента «удалить аккаунт»

Для прецедентов «получить в формате JSON», «получить в формате MathML» и «получить в формате LaTeX» действия различаются только в преобразованиях, так математическая формула в базе данных представлена в формате JSON, а остальные форматы формируются из информации JSON.

Диаграммы деятельности и диаграмма прецедентов графически описывают функционал разрабатываемого программного обеспечения.

## 3.5 Проектирование макетов пользовательского интерфейса

Проектирование макета пользовательского интерфейса - важнейшая часть реализации разработанного программного обеспечения, с помощью которого пользователь взаимодействует с программой. Создавая макет окон и страниц программы, можно заранее определить расположение кнопок управления и работу программы.

Макет страницы, которая будет отображаться гостю, представлен на рисунке 15.

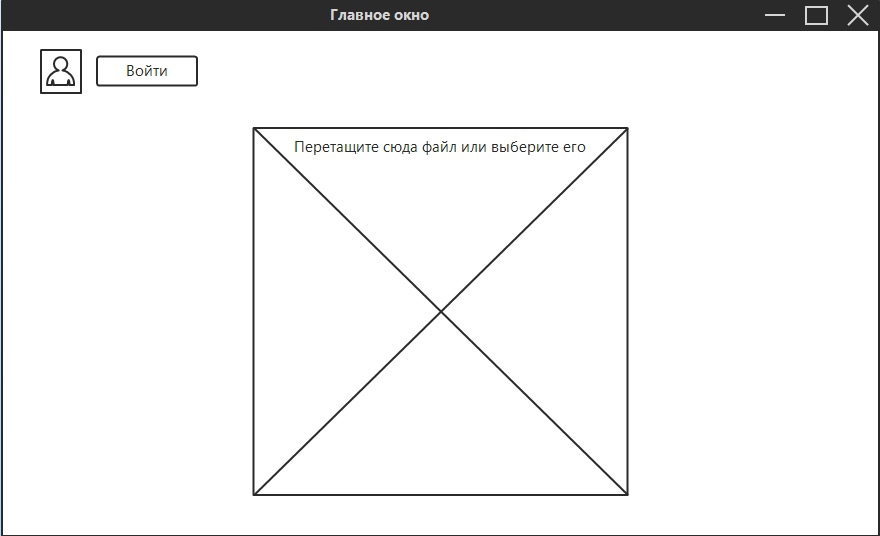


Рисунок 15 – Макет страницы гостя

На странице можно увидеть область, в которую можно перетащить изображение для распознавания, либо же нажать на область для выбора файла изображения. Также на странице можно увидеть кнопку входа в систему.

При успешном распознавании пользователю отображается модальное окно, которое представлено на рисунке 16.

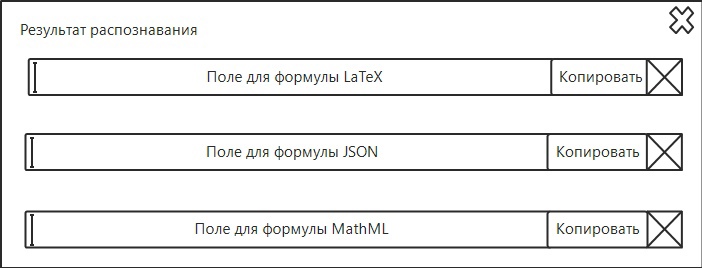


Рисунок 16 – Модальное окно результатов распознавания

В окне результатов распознавания можно увидеть текстовые поля, в которых представлена формула в формате LaTeX, JSON и MathML соответственно. У каждого текстового поля имеется кнопка копирования, при нажатии на которую происходит запись строки в поле в буфер обмена. В верхнем правом углу изображен значок закрытия модального окна.

Макет страницы для зарегистрированного пользователя изображен на рисунке 17.

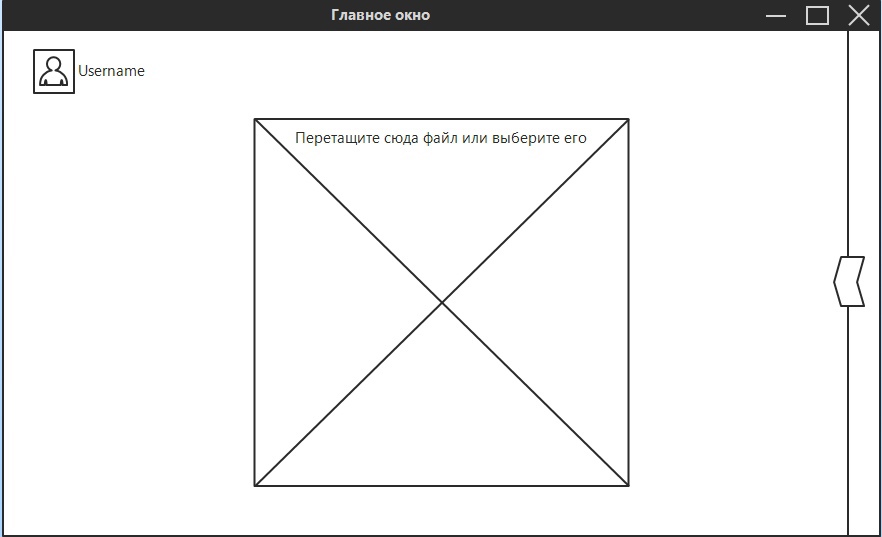


Рисунок 17 – Макет страницы зарегистрированного пользователя

На странице зарегистрированного пользователя можно наблюдать область для загрузки изображения, имя пользователя в верхнем левом углу и стрелку в правой части страницы. При нажатии на стрелку открывается история распознавания математических формул.

Макет страницы зарегистрированного пользователя с открытой историей распознавания математических формул можно увидеть на рисунке 18.

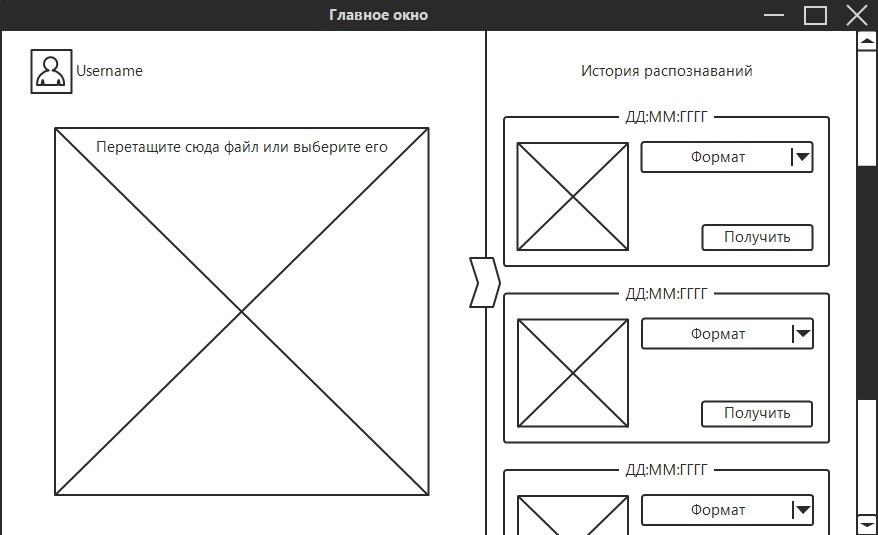


Рисунок 18 – Макет страницы зарегистрированного пользователя с открытой историей распознавания математических формул

В истории распознавания можно увидеть ячейки, которые имеют название в соответствии с датой проведения распознавания. Каждая ячейка включает в себя миниатюру изображения, с которого производилось распознавание, формат вывода результата (включает в себя LaTeX, MathML или JSON) и кнопку получить, которая отображает результат. Также можно наблюдать полоску для прокрутки истории распознавания.

Макет модального окна авторизации представлен на рисунке 19.

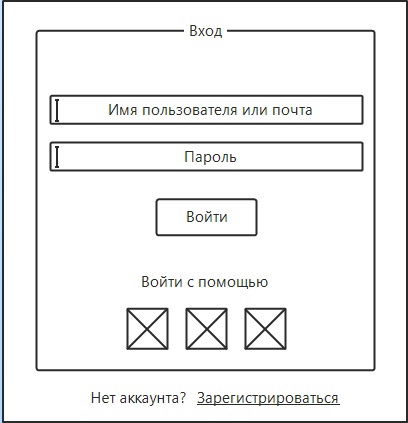


Рисунок 19 – Макет модального окна авторизации пользователя

На макете можно увидеть поле для ввода имени пользователя или почты, поле для ввода пароля, кнопку, которая при нажатии осуществляет процедуру авторизации пользователя, кнопку регистрации, которая открывает модальное окно регистрации, а также иконки для входа через сторонние сервисы.

Макет модального окна регистрации представлен на рисунке 20.

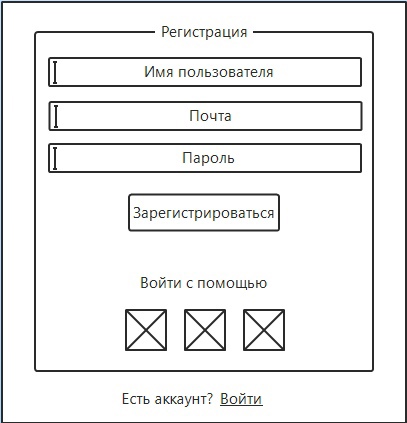


Рисунок 20 – Макет модального окна регистрации пользователя

На макете можно увидеть поля для ввода имени пользователя, почты и пароля. Также на изображении можно увидеть кнопку, осуществляющую процедуру регистрации пользователя. В нижней части макета имеется кнопка входа в систему, которая открывает модальное окно авторизации, а также иконки сторонних сервисов, с помощью которых можно осуществить авторизацию без регистрации в системе.

Таким образом, были разработаны макеты пользовательского интерфейса для веб-сервиса распознавания математических формул. Каждая страница и модальное окно используется для возложенных на нее задач, что обеспечивает положительный пользовательский опыт при работе с ним.

# 4. РАЗРАБОТКА ПРИЛОЖЕНИЯ

## 4.1 Описание инструментальных средств разработки программного обеспечения

В настоящее время существует огромное количество прогрессивных проектов и исследований в области нейронных сетей. Отличительной чертой специализированных коллекций и баз данных является их содержимое, которое состоит из наборов изображений, разработанных специально для использования сетями глубокого обучения. Кроме того, разработаны целые библиотеки, предназначенные для непосредственной работы и настройки этих нейронных сетей. Данные ресурсы включают примеры и концепции для разработки нейронных сетей, что делает их незаменимыми инструментами.

Для разработки и проектирования программного обеспечения были использованы следующие инструменты:

* Visual Studio 2022 - это комплексная интегрированная среда разработки (IDE), которую можно использовать для написания, редактирования, отладки и сборки кода, а затем для развертывания приложения. Помимо редактирования и отладки кода, Visual Studio включает компиляторы, средства завершения кода, систему управления версиями, расширения и многие другие функции для улучшения каждого этапа процесса разработки программного обеспечения [9]. В разрабатываемом программном обеспечении используется для реализации веб-сервиса.
* Язык программирования C# - это современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET [10]. Используется для программирования веб-сервиса.
* Язык программирования Python – это высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ [11]. Используется для реализации нейронных сетей и обработки изображений.
* Blazor - это экспериментальный веб UI – фреймворк на базе C#, Razor и HTML, который работает непосредственно в браузере посредством WebAssembly [12]. Цель эксперимента – в значительной мере упростить задачу построения простых и качественных одностраничных приложений, которые могут быть запущены в рамках любого браузера. Используется для реализации веб-сервиса в качестве одностраничного приложения.
* OpenCV - это библиотека с открытым исходным кодом для работы с компьютерным зрением. В OpenCV есть встроенные алгоритмы компьютерного зрения на основе машинного обучения в виде отдельных модулей с разной функциональностью [13]. Используется для обработки изображений до их использования в нейронной сети.
* Tensorflow - открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия [14]. В программном обеспечении используется для обучения нейронной сети классификации математических формул.
* Microsoft SQL Server - Система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов - Transact-SQL [15]. Используется для реализации базы данных.

## 4.2 Альтернативы и преимущества инструментальных средств разработки программного обеспечения

В качестве альтернативы для языка программирования C# может выступить язык программирования Java. Язык C# был выбран, так как имеет ряд преимуществ, таких как:

* Имеет лучшую кроссплатформенную поддержку.
* Код на языке программирования C# выполняется быстрее, так как C# использует современный компилятор, а Java – опережающий.

В качестве альтернативы для Microsoft SQL Server может быть PostgreSQL. Microsoft SQL Server был выбран, так как имеет следующие преимущества:

* Масштабирование системы. Взаимодействовать с ней можно как на простых ноутбуках, так и на ПК с мощным процессором, который способен обрабатывать большой объем запросов.
* Размер страниц – до 8 Кб. Данные извлекаются быстро, а сложную информацию удобнее хранить. Система обрабатывает транзакции в интерактивном режиме, есть динамическая блокировка.
* Автоматизация рутинных административных задач. Например, управление блокировками и памятью, редактура размеров файлов. В программе продуманы настройки, можно создавать профили пользователей.

В качестве альтернативы для фреймворка Blazor может быть Angular. Blazor был выбран, так как он обладает следующими преимуществами:

* Написание кода веб-приложений с помощью C#.
* Использование всех имеющихся на данный момент возможностей экосистемы .NET.
* Клиентская и серверная части приложения могут использовать общую логику.
* Использование Visual Studio (или Visual Studio Code) в качестве инструмента для разработки.

В качестве альтернативы для OpenCV выступает SimpleCV. OpenCV был выбран, так как имеет следующие преимущества:

* Написана на C/C++ и в сравнении с другими библиотеками работает быстрее.
* Не требует много памяти и хорошо работает при небольшом объеме RAM.
* Поддерживает большинство операционных систем, в том числе Windows, Linux и MacOS.

В качестве альтернативы для TensorFlow выступает PyTorch. TensorFlow был выбран, так как имеет ряд преимуществ, таких как:

* Высокий уровень абстракции. Библиотека написана таким образом, что не приходится задумываться о технической реализации абстрактных понятий. Это позволяет сконцентрироваться на описании логики программы и математической составляющей, а конкретную реализацию вычислений доверить TensorFlow.
* Интерактивность. Библиотека TensorFlow даёт возможность работать с компонентами модели по отдельности, что называется «на ходу». Это удобный подход, так как структуру можно менять и настраивать.
* Гибкость. Библиотека эффективна для разных направлений машинного обучения, например, для создания нейросетей, глубокого обучения и прочего.
* Кроссплатформенность. TensorFlow, как и Python, поддерживает популярные операционные системы.

## 4.3 Создание и обслуживание базы данных

Перед началом разработки веб-сервиса была реализована база данных по спроектированной ER-модели. Физическая модель разработанной базы данных представлена на рисунке 21.

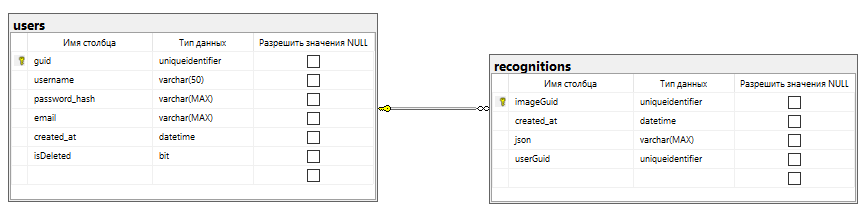


Рисунок 21 – Физическая модель базы данных

В базе данных между таблицами users (пользователи) и recognitions (распознавания) организована связь «один-ко-многим», которая обусловлена тем, что один конкретный пользователь веб-сервиса может иметь несколько успешно произведенных распознаваний математических формул за все время. Таким образом, связь «один-ко-многим» позволяет получить все распознавания для запрошенного пользователя.

SQL-скрипт создания базы данных и таблиц представлен в приложении А.

## 4.4 Решение административных функций в базе данных

В базе данных была создана роль server, которая обладает правами на удаление, добавление и редактирование информации в таблицах базы данных. Права роли server приведены на рисунке 22.

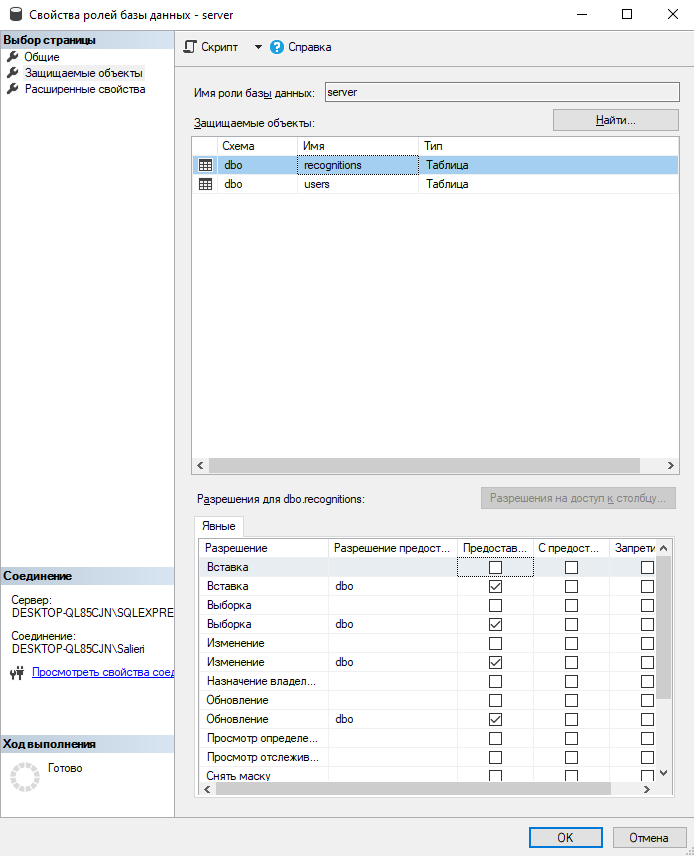


Рисунок 22 – Права роли server

Для роли server был создан пользователь recogntiton\_server. Результат создания пользователя с присвоением роли представлен на рисунке 23.

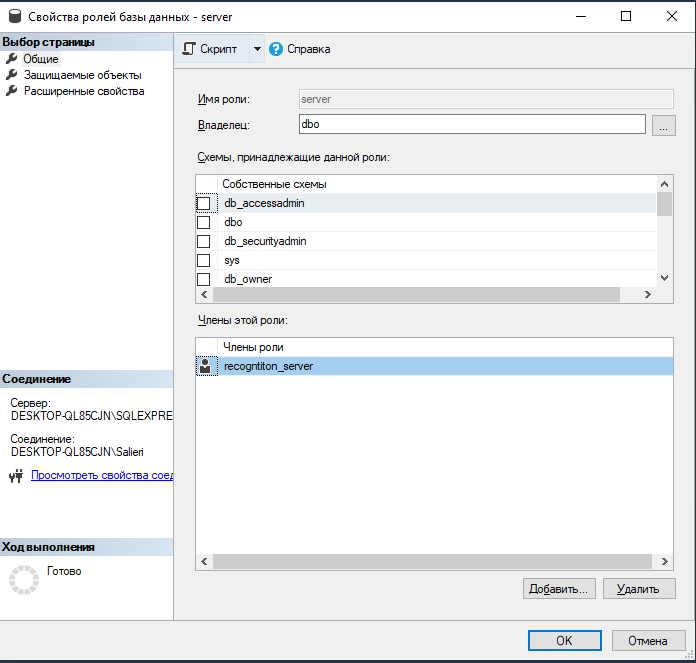


Рисунок 23 – Созданный пользователь с присвоенной ролью

SQL-скрипт создания ролей и пользователей представлен в приложении Б.

## 4.5 Разработка веб-сервиса

Перед началом разработки страниц веб-сервиса был создан базовый слой приложения (MainLayout), который включает в себя меню навигации, основной контент и нижнюю часть всех страниц.

Для базового слоя приложения были созданы отдельные компоненты для меню навигации и нижней части страницы.

Компонент меню навигации (NavMenu компонент) включает в себя наименование бренда, кнопки входа и регистрации в системе и меню выбора языка.

Компонент нижней части страницы включает в себя краткое описание веб-сервиса, которое можно наблюдать на всех страницах веб-сервиса.

После создания базового слоя были разработаны основные компоненты главной страницы приложения, такие как:

* Зона для загрузки изображения (DragAndDropImageZone компонент) – отвечает за загрузку исходного изображения на сервер для последующего распознавания на нем математической формулы.
* Модальное окно для авторизации в веб-сервисе (ModalLoginWindow компонент) – отвечает за отображение окна авторизации на веб-странице и заключает в себе логику для проведения процедуры авторизации в веб-сервисе.
* Модальное окно для регистрации в веб-сервисе (ModalRegisterWindow компонент) – отвечает за отображение окна регистрации на веб-странице и заключает в себе логику для проведения процедуры регистрации в веб-сервисе.
* Меню выбора языка на отображаемой странице (CultureSelector компонент) – отвечает за смена языка на веб-странице.

Разработанные компоненты были размещены на главной странице веб-сервиса, которая представлена на рисунке 24.

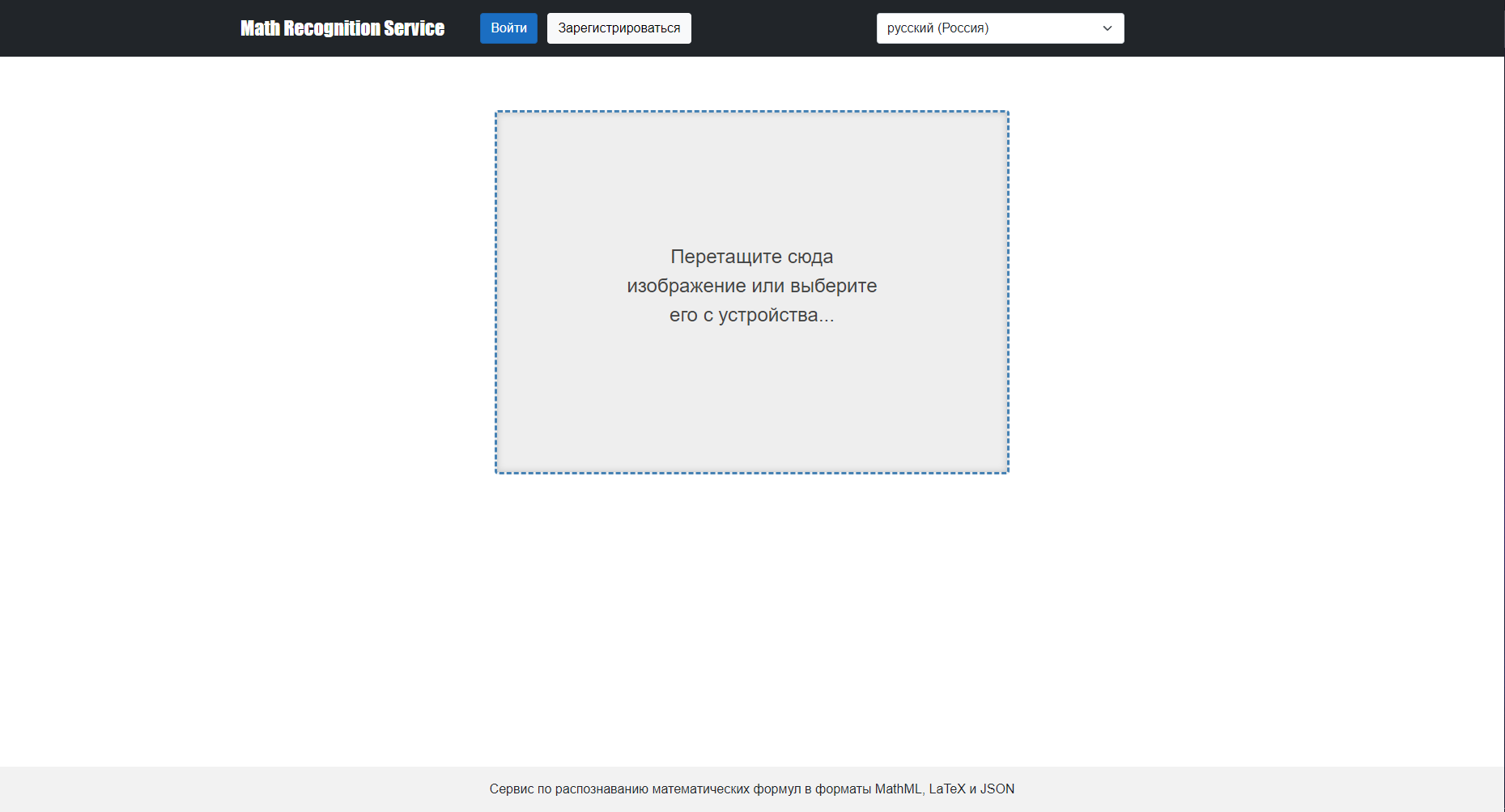


Рисунок 24 – Главная страница веб-сервиса

При нажатии кнопки «Войти» в меню навигации веб-сервиса отображается модальное окно авторизации в системе, которое представлено на рисунке 25.

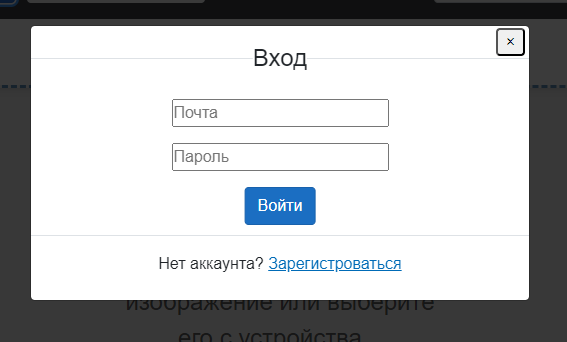


Рисунок 25 – Модальное окно авторизации в веб-сервисе

Для окна авторизации реализована логика валидации вводимых данных. Так, если пользователь не заполнил поля для авторизации, то отображается ошибка валидации, которая представлена на рисунке 26.

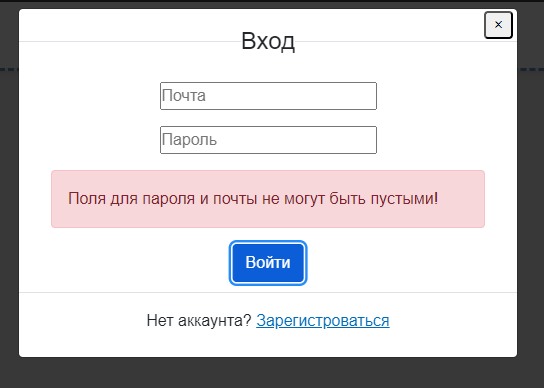


Рисунок 26 – Ошибка валидации при некорректном вводе данных для авторизации

Окно также поддерживает разный вид ошибок валидации, исходя из того, какие именно поля не заполнил пользователь.

Также окно имеет проверку на корректность введенной почты с помощью использования регулярных выражений, так при вводе некорректного формата почты отображается ошибка валидации о некорректном виде почты, которая представлена на рисунке 27.

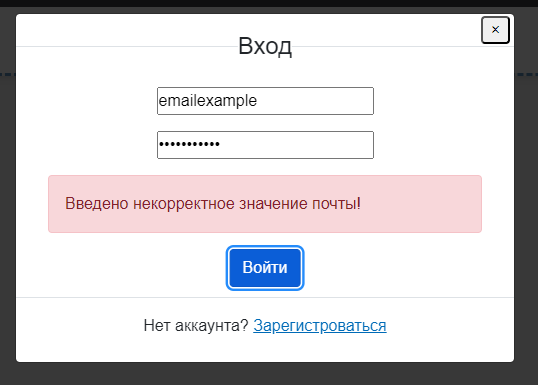


Рисунок 27 – Ошибка валидации при некорректном виде электронной почты

Регулярное выражение и часть логики для валидации правильности ввода почты приведено в листинге 1.

Листинг 1 – Регулярное выражение и часть логики для валидации почты

private bool ValidateData()

{

bool validateComplete = true;

Regex regex = new Regex(@"^([\w**\.\-**]+)@([\w**\-**]+)((**\.**(\w){2,3})+)$");

Match match = regex.Match(loginOrEmail);

ResetValidateNotifications();

if (string.IsNullOrWhiteSpace(loginOrEmail)){

isIncorrectLoginOrEmail = true;

validateComplete = false;

}

if (string.IsNullOrWhiteSpace(password)){

isIncorrectPassword = true;

validateComplete = false;

}

if (!match.Success)

{

isIncorrectEmailView = true;

validateComplete = false;

}

return validateComplete;

}

// ...

@if (isIncorrectEmailView)

{

<div class="alert alert-danger my-2" role="alert">

@\_localizer["incorrectEmailViewError"]

</div>

}

При условии, что все данные для авторизации введены верно, при нажатии кнопки «Войти» происходит попытка авторизации в веб-сервисе под указанными пользователем данными.

При нажатии кнопки «Регистрация» происходит вызов модального окна регистрации, которое представлено на рисунке 28.

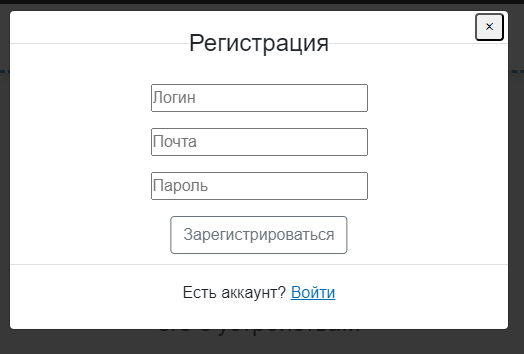


Рисунок 28 – Модальное окно регистрации в веб-сервисе

Для окна регистрации реализована логика валидации данных, аналогичная окну авторизации. Пример ошибки валидации представлен на рисунке 29.

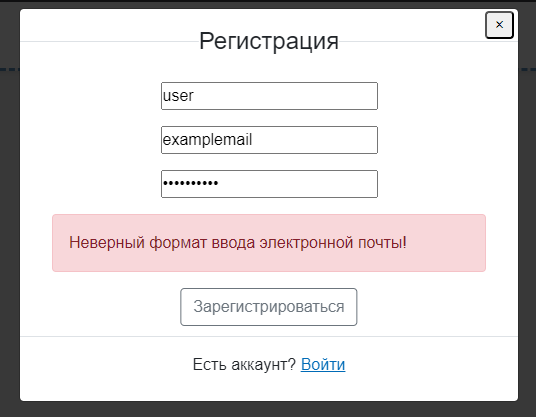


Рисунок 29 – Ошибка валидации данных в окне регистрации

При условии, что все поля окна заполнены корректными данными и при нажатии на кнопку «Зарегистрироваться» происходит попытка регистрации в веб-сервисе.

Основным контентом на главной странице веб-сервиса является поле для загрузки изображения для распознавания. Для загрузки изображения существует два способа:

* Перетаскивание изображения в зону для загрузки (Drag And Drop) – метод подразумевает, что пользователь с помощью мыши перетащит исходное изображение в зону загрузки.
* Выбор файла с устройства – метод подразумевает, что пользователь нажмет на зону загрузки изображения и выберет его с устройства.

Пример загруженного изображения в компонент зоны загрузки изображения представлен на рисунке 30.

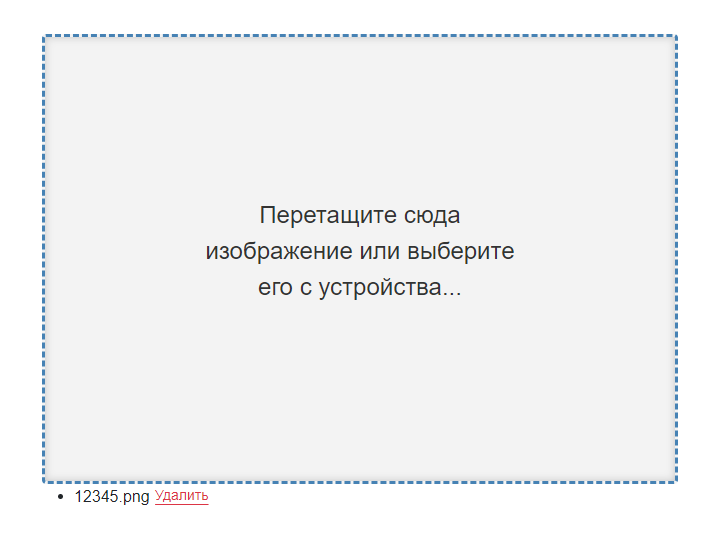


Рисунок 30 – Пример загруженного изображения в компонент

Зона загрузки также имеется логику для валидации загруженных изображений, так компонент имеет ограничения на количество загружаемых за раз изображений (за раз можно загрузить только одно изображение), на формат файла изображения и на размер файла.

Листинг исходного кода валидации зоны загрузки изображения представлен в листинге 2.

Листинг 2 – Валидация изображения при загрузке на сервер

private async Task SingleUpload(IFileListEntry[] files)

{

dropClass = "";

fileSizeError = false;

fileTypeError = false;

singleUploadError = false;

List<string> acceptedFileTypes = new List<string>() { "image/png", "image/jpeg", "image/gif" };

if (files != null)

{

if (files.Length != 1) // Нельзя загрузить больше одного файла за раз

{

singleUploadError = true;

return;

}

foreach (var file in files)

{

bool error = false;

if (file.Size > MaxFileSize)

{

error = true;

fileSizeError = true;

}

if (!acceptedFileTypes.Contains(file.Type))

{

error = true;

fileTypeError = true;

}

if (!error)

{

selectedFiles.Add(file);

}

}

}

}

// ...

@if (fileTypeError)

{

<div class="alert alert-danger my-2">

@\_localizer["fileTypeError"]

</div>

}

@if (singleUploadError)

{

<div class="alert alert-danger my-2">

@\_localizer["singleUploadError"]

</div>

}

@if (fileSizeError)

{

<div class="alert alert-danger my-2">

@\_localizer["fileSizeError"] @MaxFileSizeMB MB.

</div>

}

Пример ошибки валидации изображения представлен на рисунке 31.

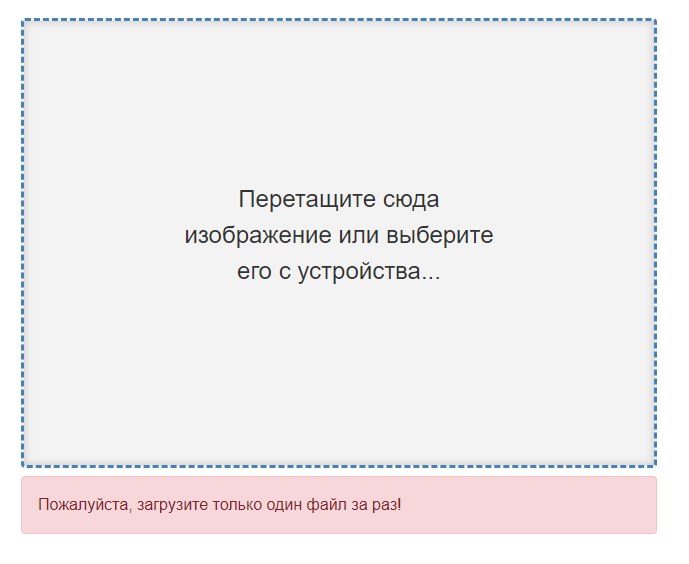


Рисунок 31 – Пример ошибки валидации при загрузке изображения

При загрузке изображения оно загружается на сервер после чего происходит процедура распознавания изображения. Также если пользователь был авторизован в системе, то изображение и информация о распознавании сохраняется в базе данных.

На странице также существует меню выбора языка, которое находится в правой части меню навигации и представлено на рисунке 32.

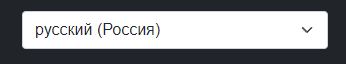


Рисунок 32 – Меню выбора языка

В веб-сервисе на данный момент поддерживаются русский и английский языки. Пример страницы, локализованной для английского языка, представлен на рисунке 33.

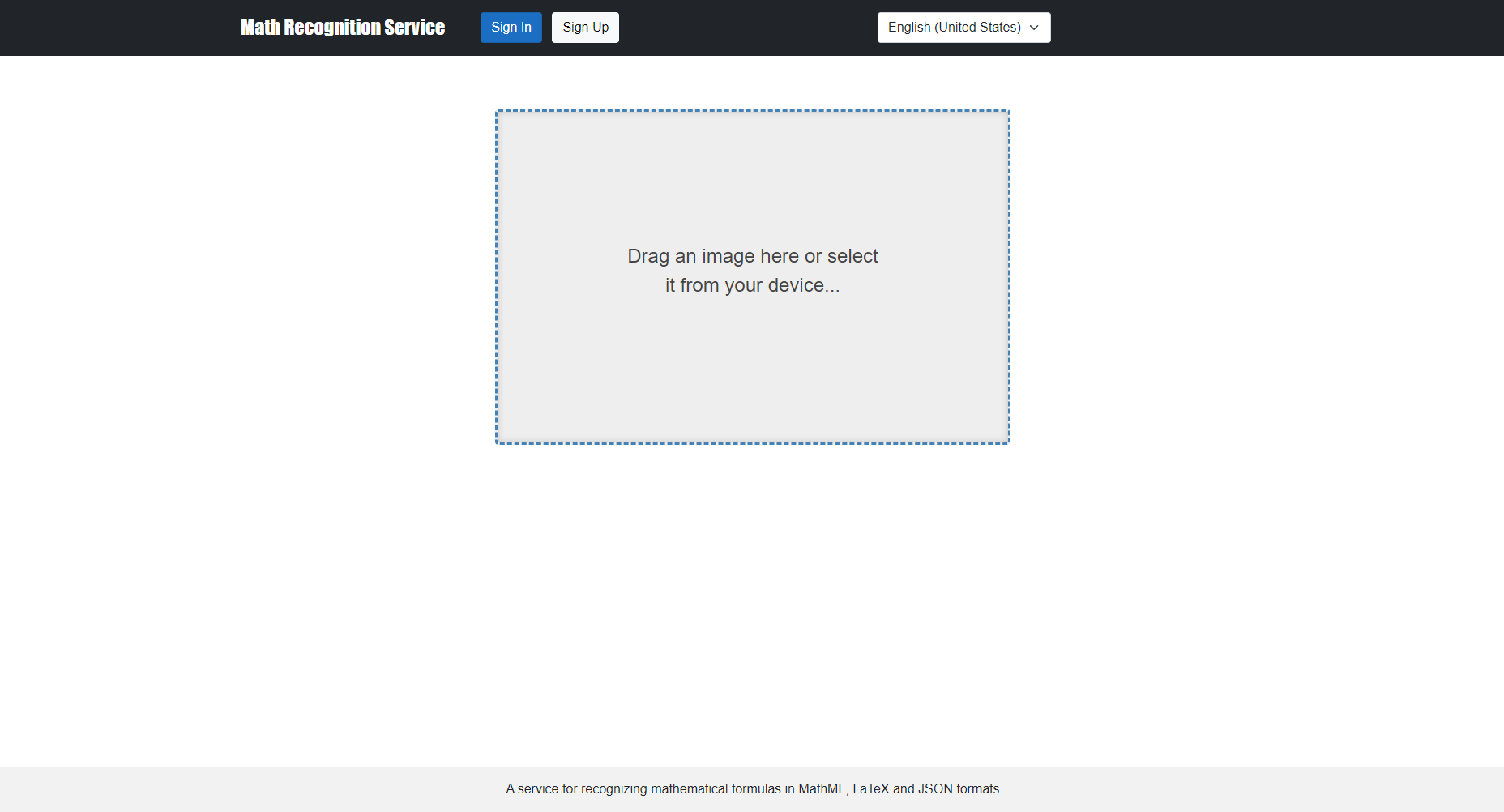


Рисунок 33 – Пример страницы на английском языке

Для локализации страниц под определенные языки были созданы файлы ресурсов, которые содержат пары «ключ-значение» для каждого выводимого текстового содержимого в компонентах страницы. Файлы имеют наименования, согласно правилам наименования файлов локализации. Расположение файлов локализации и их содержимое представлено на рисунках 34-36 соответственно.

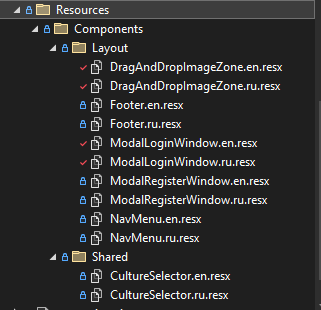


Рисунок 34 – Расположение файлов локализации компонентов

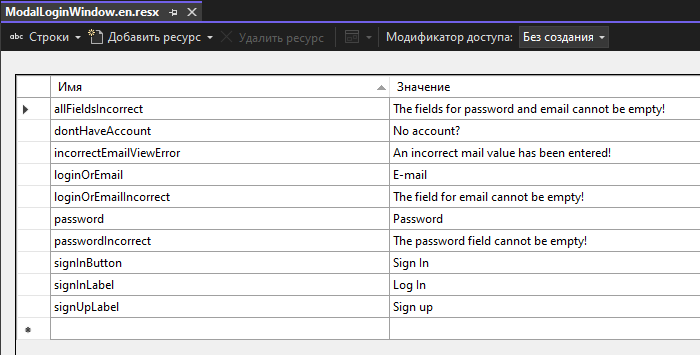


Рисунок 35 – Содержимое файла ресурсов для локализации компонента ModalLoginWindow на английский язык

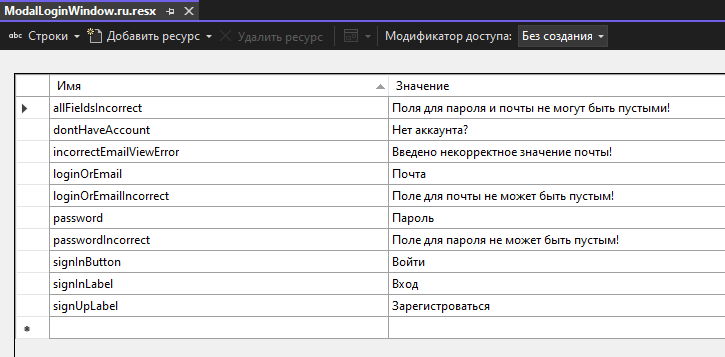


Рисунок 36 – Содержимое файла ресурсов для локализации компонента ModalLoginWindow на русский язык

Фрагмент исходного кода, который используется для локализации элемента окна авторизации представлен в листинге 3.

Листинг 3 – Фрагмент исходного кода локализации элемента окна авторизации

<div class="row-auto position-relative">

<div class="w-50 mx-auto py-2">

@\*

Пароль

\*@

<input type="password" style="width: 100%;" placeholder="@\_localizer["password"]" @bind="password" />

</div>

</div>

Так, при выборе языка на странице, данные для локализации извлекаются из связанного с этим языком файла ресурсов. В данном случаем сервер понимает, из какого файла и для какого компонента взять локализацию из соглашения о наименовании файлов ресурсов для локализации.

Исходный код страниц и компонентов представлен в приложении В.

# 5. ТЕСТИРОВАНИЕ

## 5.1 Разработка тест-требований

После основательного изучения предметной области и формирования общего понимания о проектируемой системе, становится возможным четко определить требования, которые предъявляются к разрабатываемому приложению.

Тест-требования могут представлять собой:

1. Проверка авторизации в системе:
   1. проверить, что при вводе некорректных данных об имени пользователя, почте или пароле происходит отображение соответствующей ошибки;
   2. проверить, что при вводе корректных данных об имени пользователя, почте и пароле осуществляется предоставление доступа к истории распознаваний;
   3. проверить, что при вводе корректных данных об имени пользователя и почте, при неоднократном вводе неверного пароля, появляется подсказка о восстановлении пароля.
2. Проверка восстановления пароля для входа в систему:
   1. проверить, что при вводе незарегистрированной почты отображается соответствующее сообщение;
   2. проверить, что при вводе корректной почты на нее отправляется письмо и отображается сообщение об успешной отправке.
3. Проверка регистрации в системе:
   1. проверить, что все данные о пользователе (имя пользователя, почта и пароль) успешно добавлены в базу данных системы, в противном случае происходит отображение соответствующего сообщения;
   2. проверить, что при успешной регистрации производится отправка подтверждающего сообщения на почту, а также открывается доступ к истории распознавания.
4. Проверка возможности распознавания математических формул:
   1. проверить, что при загрузке изображения и успешном распознавании отображается модальное окно, в котором приведен результат распознавания в форматах JSON, MathML и LaTeX;
   2. проверить, что при загрузке изображения и неудавшемся распознавании отображается сообщение об невозможности распознавания изображения.
5. Проверка истории распознавания математических формул:
   1. проверить, что при использовании веб-сервиса в качестве гостя пользователь не имеет возможности видеть и взаимодействовать с историей распознавания математических формул;
   2. проверить, что при использовании веб-сервиса в качестве зарегистрированного пользователя имеется возможность просмотра предыдущих распознаваний математических формул;
   3. проверить, что все ячейки из истории распознавания математических формул имеют миниатюру распознаваемого изображения, возможность выбора формата получения результата (в качестве JSON, MathML или LaTeX) в указанном формате.
   4. проверить, что при отсутствии распознаваний в истории отображается соответствующее сообщение.
6. Проверка удаления аккаунта:
   1. проверить, что при инициировании процедуры удаления аккаунта происходит его деактивация.

## 5.2 Разработка тест-плана

После написания требований к тестированию, необходимо создать план тестирования. Этот план является неотъемлемой частью процесса тестирования разработанного приложения, так как он обеспечивает ясность ожидаемых результатов и определяет, какие данные должны быть введены и куда они должны быть внесены.

**Тестовый пример 1.**

Номер тест-требования: 1.1.

Описание теста: тест проверяет, что при вводе некорректных данных об имени пользователя, почте или пароле происходит отображение соответствующей ошибки.

Входные данные:

* окно авторизации;
* «логин или почта»: user;
* «пароль»: user.

Ожидаемые выходные данные: отображение ошибки входа.

Сценарий:

* открыть веб-сервис по распознаванию математических формул;
* нажать кнопку «Вход»;
* ввести данные в поле «логин или почта» и в поле «пароль»;
* нажать кнопку «Войти».

**Тестовый пример 2.**

Номер тест-требования: 1.2.

Описание теста: тест проверяет, что при вводе корректных данных об имени пользователя, почте и пароле осуществляется предоставление доступа к истории распознаваний.

Входные данные:

* окно авторизации;
* «логин или почта»: user;
* «пароль»: password12345.

Ожидаемые выходные данные: открытие доступа к истории распознаваний.

Сценарий:

* открыть веб-сервис по распознаванию математических формул;
* нажать кнопку «Вход»;
* ввести данные в поле «логин или почта» и в поле «пароль»;
* нажать кнопку «Войти».

**Тестовый пример 3.**

Номер тест-требования: 1.3.

Описание теста: тест проверяет, что при вводе корректных данных об имени пользователя и почте, при неоднократном вводе неверного пароля, появляется подсказка о восстановлении пароля.

Входные данные:

* окно авторизации;
* «логин или почта»: user;
* «пароль»: user.

Ожидаемые выходные данные: отображение подсказки для восстановления пароля.

Сценарий:

* открыть веб-сервис по распознаванию математических формул;
* нажать кнопку «Вход»;
* ввести данные в поле «логин или почта» и в поле «пароль»;
* нажать кнопку «Войти» и повторять, пока не появится подсказка о восстановлении пароля.

**Тестовый пример 4.**

Номер тест-требования: 2.1.

Описание теста: тест проверяет, что при вводе незарегистрированной почты отображается соответствующее сообщение.

Входные данные:

* окно восстановления пароля;
* «почта»: [examplemail@bk.ru](mailto:examplemail@bk.ru)

Ожидаемые выходные данные: отображение ошибки о том, что почта не зарегистрирована.

Сценарий:

* нажать на подсказку для перехода к восстановлению пароля;
* ввести в поле «почта» почту для восстановления;
* нажать кнопку «Восстановить».

**Тестовый пример 5.**

Номер тест-требования: 2.2.

Описание теста: тест проверяет, что при вводе корректной почты на нее отправляется письмо и отображается сообщение об успешной отправке.

Входные данные:

* окно восстановления пароля;
* «почта»: maxim.bash@yandex.ru

Ожидаемые выходные данные: отображение сообщения об успешной отправке письма на указанную почту.

Сценарий:

* нажать на подсказку для перехода к восстановлению пароля;
* ввести в поле «почта» почту для восстановления;
* нажать кнопку «отправить».

**Тестовый пример 6.**

Номер тест-требования: 3.1.

Описание теста: тест проверяет, что все данные о пользователе (имя пользователя, почта и пароль) успешно добавлены в базу данных системы, в противном случае происходит отображение соответствующего сообщения;

Входные данные:

* окно регистрации;
* «логин»: user1;
* «почта»: [user1@bk.com](mailto:user1@bk.com);
* «пароль»: user1234.

Ожидаемые выходные данные: данные успешно добавлены в базу данных.

Сценарий:

* ввести данные в поле «логин»;
* ввести данные в поле «почта»;
* ввести данные в поле «пароль»;
* нажать кнопку «Зарегистрироваться».

**Тестовый пример 7.**

Номер тест-требования: 3.2.

Описание теста: тест проверяет, что при успешной регистрации производится отправка подтверждающего сообщения на почту, а также открывается доступ к истории распознавания.

Входные данные:

* окно регистрации;
* «логин»: user1;
* «почта»: [user1@bk.com](mailto:user1@bk.com);
* «пароль»: user1234.

Ожидаемые выходные данные: отправка подтверждающего сообщения на указанную почту, открытие доступа к истории распознаваний.

Сценарий:

* ввести данные в поле «логин»;
* ввести данные в поле «почта»;
* ввести данные в поле «пароль»;
* нажать кнопку «Зарегистрироваться»;
* проверить почту [user1@bk.com](mailto:user1@bk.com) и перейти по ссылке в сообщении от веб-сервиса;
* на главной странице веб-сервиса нажать кнопку «История».

**Тестовый пример 8.**

Номер тест-требования: 4.1.

Описание теста: тест проверяет, что при загрузке изображения и успешном распознавании отображается модальное окно, в котором приведен результат распознавания в форматах JSON, MathML и LaTeX;

Входные данные:

* зона загрузки изображения.

Ожидаемые выходные данные: отображение модального окна с результатом распознавания в форматах JSON, MathML и LaTeX.

Сценарий:

* с помощью перетаскивания или выбора изображения загрузить изображение математической формулы для распознавания.

**Тестовый пример 9.**

Номер тест-требования: 4.2.

Описание теста: тест проверяет, что при загрузке изображения и неудавшемся распознавании отображается сообщение о невозможности распознавания изображения.

Входные данные:

* изображение математической формулы для распознавания;
* главная страница веб-сервиса;
* зона загрузки изображения.

Ожидаемые выходные данные: отображение сообщения о невозможности распознавания изображения.

Сценарий:

* с помощью перетаскивания или выбора изображения загрузить изображение математической формулы для распознавания.

**Тестовый пример 10.**

Номер тест-требования: 5.1.

Описание теста: тест проверяет, что при использовании веб-сервиса в качестве гостя пользователь не имеет возможности видеть и взаимодействовать с историей распознавания математических формул;

Входные данные:

* главная страница веб-сервиса.

Ожидаемые выходные данные: отсутствие возможности открыть историю распознавания.

Сценарий:

* попытаться произвести взаимодействие с историей.

**Тестовый пример 11.**

Номер тест-требования: 5.2.

Описание теста: тест проверяет, что при использовании веб-сервиса в качестве зарегистрированного пользователя имеется возможность просмотра предыдущих распознаваний математических формул;

Входные данные:

* главная страница веб-сервиса;
* история распознавания математических формул;
* «логин или почта»: user;
* «пароль»: user1234.

Ожидаемые выходные данные: возможность открыть историю распознавания.

Сценарий:

* нажать кнопку «Войти»;
* ввести данные в поле «логин или почта» и в поле «пароль»;
* нажать кнопку «Войти»;
* попытаться произвести взаимодействие с историей.

**Тестовый пример 12.**

Номер тест-требования: 5.3.

Описание теста: тест проверяет, что все ячейки из истории распознавания математических формул имеют миниатюру распознаваемого изображения, возможность выбора формата получения результата (в качестве JSON, MathML или LaTeX) в указанном формате.

Входные данные:

* главная страница веб-сервиса;
* история распознавания математических формул;
* «логин или почта»: user;
* «пароль»: user1234.

Ожидаемые выходные данные: все ячейки из истории распознавания математических формул имеют миниатюру распознаваемого изображения, возможность выбора формата получения результата в указанном формате.

Сценарий:

* нажать кнопку «Войти»;
* ввести данные в поле «логин или почта» и в поле «пароль»;
* нажать кнопку «Войти»;
* открыть историю распознаваний;
* на последнем распознавании в истории распознаваний выбрать формат LaTeX и нажать кнопку «получить».

**Тестовый пример 13.**

Номер тест-требования: 5.4.

Описание теста: тест проверяет, что при отсутствии распознаваний в истории отображается соответствующее сообщение.

Входные данные:

* главная страница веб-сервиса;
* история распознавания математических формул;
* «логин или почта»: user;
* «пароль»: user1234.

Ожидаемые выходные данные: отображение сообщения о отсутствии распознаваний.

Сценарий:

* нажать кнопку «Войти»;
* ввести данные в поле «логин или почта» и в поле «пароль»;
* нажать кнопку «Войти»;
* открыть историю распознаваний.

**Тестовый пример 14.**

Номер тест-требования: 6.1.

Описание теста: тест проверяет, что при инициировании процедуры удаления аккаунта происходит его деактивация.

Входные данные:

* главная страница веб-сервиса;

Ожидаемые выходные данные: деактивация аккаунта.

Сценарий:

* нажать кнопку «удалить аккаунт».

Заключение

В ходе курсовой работы были рассмотрены основные аспекты проектирования и разработки веб-сервиса по распознаванию математических формул.

Цель работы была достигнута путём выполнения основных задач курсовой работы:

* изучены существующие системы и технологии распознавания объектов, рукописных и печатных символов;
* проанализированы требования системы к формату исходного изображения для распознавания;
* определена структура базы данных;
* реализован и протестирован веб-сервис;
* произведен анализ результатов тестирования.

Таким образом, разработанный веб-сервис упрощает перенос рукописных или печатных математических формул в форматы LaTeX, MathML и JSON.

В качестве перспективы развития веб-сервиса можно выделить реализацию программного интерфейса (API) для запросов пользователей и расширение функционала веб-сервиса для поддержки прочих форматов вывода результата распознавания.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Саймон Хайкин. Нейронные сети. Полный курс / Саймон Хайкин – Вильямс, 2019. – 1104 с.
2. Geoffrey E. Hinton. A Fast Learning Algorithm for Deep Belief Nets / Geoffrey E. Hinton, Simon Osindero, Yee-Whye The. – Toronto: Department of Computer Science, 2006. – 28 с.
3. Google Документы / [Электронный ресурс] // Google Docs : [сайт]. — URL: https://www.google.com/docs/about/ (дата обращения: 12.10.2023).
4. Philipp Kühl, Daniel Kirsch Detexify LaTeX handwritten symbol recognition / Philipp Kühl, Daniel Kirsch [Электронный ресурс] // Detexify : [сайт]. — URL: https://detexify.kirelabs.org/classify.html (дата обращения: 12.10.2023).
5. Free Online OCR / [Электронный ресурс] // Free Online OCR : [сайт]. — URL: https://www.newocr.com/ (дата обращения: 12.10.2023).
6. Adobe Acrobat Pro / [Электронный ресурс] // Adobe : [сайт]. — URL: https://www.adobe.com/acrobat/acrobat-pro.html (дата обращения: 12.10.2023).
7. Business-Oriented OCR Software / [Электронный ресурс] // Kofax : [сайт]. — URL: https://www.kofax.com/products/omnipage/ultimate (дата обращения: 12.10.2023).
8. ABBYY FineReader PDF / [Электронный ресурс] // ABBYY FineReader PDF : [сайт]. — URL: https://pdf.abbyy.com/ (дата обращения: 12.10.2023).
9. Что такое Visual Studio? / [Электронный ресурс] // Microsoft : [сайт]. — URL: https://learn.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2019 (дата обращения: 15.11.2023).
10. Краткий обзор языка C# / [Электронный ресурс] // Microsoft : [сайт]. — URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/ (дата обращения: 15.11.2023).
11. Python / [Электронный ресурс] // Python : [сайт]. — URL: https://www.python.org/?hl=ru. (дата обращения: 14.11.2023).
12. ASP.NET Core Blazor / [Электронный ресурс] // Microsoft : [сайт]. — URL: https://learn.microsoft.com/ru-ru/aspnet/core/blazor/?view=aspnetcore-7.0 (дата обращения: 15.11.2023).
13. OpenCV / [Электронный ресурс] // OpenCV : [сайт]. — URL: https://opencv.org/ (дата обращения: 15.11.2023).
14. TensorFlow / [Электронный ресурс] // TensorFlow : [сайт]. — URL: https://www.tensorflow.org/?hl=ru (дата обращения: 15.11.2023).
15. Знакомство с SQL Server 2022 / [Электронный ресурс] // Microsoft : [сайт]. — URL: https://www.microsoft.com/ru-ru/sql-server (дата обращения: 15.11.2023).

# ПРИЛОЖЕНИЕ А

Создание базы данных

CREATE DATABASE [MathRecognitionService]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'MathRecognitionService', FILENAME = N'F:\SQLServer\MSSQL16.SQLEXPRESS\MSSQL\DATA\MathRecognitionService.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'MathRecognitionService\_log', FILENAME = N'F:\SQLServer\MSSQL16.SQLEXPRESS\MSSQL\DATA\MathRecognitionService\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

WITH CATALOG\_COLLATION = DATABASE\_DEFAULT, LEDGER = OFF

GO

Создание таблицы users

CREATE TABLE [dbo].[users](

[guid] [uniqueidentifier] NOT NULL,

[username] [varchar](50) NOT NULL,

[password\_hash] [varchar](255) NOT NULL,

[email] [varchar](255) NOT NULL,

[created\_at] [datetime] NOT NULL,

[isDeleted] [bit] NOT NULL,

PRIMARY KEY CLUSTERED

(

[guid] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

Создание таблицы recognitions

CREATE TABLE [dbo].[recognitions](

[imageGuid] [uniqueidentifier] NOT NULL,

[created\_at] [datetime] NOT NULL,

[JSON] [nvarchar](max) NOT NULL,

[userGuid] [uniqueidentifier] NOT NULL,

PRIMARY KEY CLUSTERED

(

[imageGuid] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

# ПРИЛОЖЕНИЕ Б

Создание роли server и установка прав

CREATE ROLE [server]

GO

GRANT DELETE,UPDATE,SELECT,INSERT ON users TO server

GRANT DELETE,UPDATE,SELECT,INSERT ON recognitions TO server

Создание пользователя recognition\_server и установка роли

EXEC sp\_addlogin 'recognition\_server','1234567890', 'MathRecognitionService'

use MathRecognitionService

EXEC sp\_adduser 'recognition\_server', 'recognition\_server'

EXEC sp\_addrolemember 'server', 'recognition\_server'

# ПРИЛОЖЕНИЕ В

Слой MainLayout

@inherits LayoutComponentBase

<body>

<div class="wrapper">

<header>

<**NavMenu** />

</header>

<main class="content">

@Body

</main>

<div class="footer">

<**Footer** />

</div>

</div>

</body>

Компонент Footer

@inject IStringLocalizer<Footer> \_localizer

@rendermode InteractiveServer

<section class="">

<!-- Footer -->

<footer class="bg-body-tertiary footer">

<div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.05);">

@\_localizer["message"]

</div>

<!-- Copyright -->

</footer>

<!-- Footer -->

</section>

Компонент NavMenu

@inject IStringLocalizer<NavMenu> \_localizer

@rendermode InteractiveServer

<nav class="navbar navbar-expand-lg navbar-dark bg-dark py-3 position-relative">

<div class="container-xxl d-flex align-items-md-center">

<div class="">

<a class="navbar-brand ms-3 brand">@\_localizer["brand"]</a>

</div>

<div class="container">

<button type="button" class="btn btn-primary ms-3" @onclick="() => OpenLoginDialog()">@\_localizer["signIn"]</button>

<button type="button" class="btn btn-light ms-2" @onclick="() => OpenRegisterDialog()">@\_localizer["signUp"]</button>

</div>

<div class="container">

<**MathRecognitionService.Components.Shared.CultureSelector** />

</div>

</div>

</nav>

@if (LoginDialogOpen)

{

ResetModalWindows();

<**ModalLoginWindow** **OnClose**="@OnLoginDialogClose" **OnRegisterCall**="OpenRegisterDialog"/>

}

@if (RegisterDialogOpen)

{

ResetModalWindows();

<**ModalRegisterWindow** **OnClose**="@OnRegisterDialogClose" **OnLoginCall**="OpenLoginDialog"/>

}

@code {

public bool LoginDialogOpen { get; set; }

public bool RegisterDialogOpen { get; set; }

private void ResetModalWindows()

{

LoginDialogOpen = false;

RegisterDialogOpen = false;

}

private async Task OnLoginDialogClose(bool accepted)

{

LoginDialogOpen = false;

StateHasChanged();

}

private async Task OnRegisterDialogClose(bool accepted)

{

RegisterDialogOpen = false;

StateHasChanged();

}

private void OpenRegisterDialog()

{

RegisterDialogOpen = true;

StateHasChanged();

}

private void OpenLoginDialog()

{

LoginDialogOpen = true;

StateHasChanged();

}

}

Компонент ModalLoginWindow

@using System.Text.RegularExpressions

@inject IStringLocalizer<ModalLoginWindow> \_localizer

@rendermode InteractiveServer

<div class="modal fade show" id="myModal" style="display:block; background-color: rgba(10,10,10,.8);"

aria-modal="true" role="dialog">

<div class="modal-dialog py-5">

<div class="modal-content">

<div class="modal-header position-relative">

<h4 class="modal-title position-absolute start-50 top-50 translate-middle my-3" style="background-color: white;">@\_localizer["signInLabel"]</h4>

@\*

Крестик (закрыть окно)

\*@

<button type="button" class="close position-absolute top-50 end-0 translate-middle-y mx-1 rounded px-2 py-0" @onclick="@ModalCancel">&times;</button>

</div>

<div class="modal-body container">

<div class="col justify-content-center p-3">

<div class="row-auto position-relative">

<div class="w-50 mx-auto py-2">

@\*

почта

\*@

<input type="text" style="width: 100%;" placeholder="@\_localizer["loginOrEmail"]" @bind="loginOrEmail" />

</div>

</div>

<div class="row-auto position-relative">

<div class="w-50 mx-auto py-2">

@\*

Пароль

\*@

<input type="password" style="width: 100%;" placeholder="@\_localizer["password"]" @bind="password" />

</div>

@if (isIncorrectPassword && isIncorrectLoginOrEmail)

{

<div class="alert alert-danger my-2" role="alert">

@\_localizer["allFieldsIncorrect"]

</div>

}

else if (isIncorrectPassword)

{

<div class="alert alert-danger my-2" role="alert">

@\_localizer["passwordIncorrect"]

</div>

}

else if (isIncorrectEmailView)

{

<div class="alert alert-danger my-2" role="alert">

@\_localizer["incorrectEmailViewError"]

</div>

}

else if (isIncorrectLoginOrEmail)

{

<div class="alert alert-danger my-2" role="alert">

@\_localizer["loginOrEmailIncorrect"]

</div>

}

</div>

<div class="row-auto position-relative">

<div class="w-50 mx-auto py-2 position-relative">

@\*

Войти

\*@

<button type="button" class="btn btn-primary position-absolute start-50 top-50 translate-middle-x" @onclick=@ModalOk>@\_localizer["signInButton"]</button>

</div>

</div>

</div>

</div>

<div class="modal-footer my-2" style="flex-direction:column;">

<p>@\_localizer["dontHaveAccount"] <a href="" @onclick="ModalRegisterCall">@\_localizer["signUpLabel"]</a></p>

</div>

</div>

</div>

</div>

@code {

private string loginOrEmail = string.Empty;

private string password = string.Empty;

private bool isIncorrectLoginOrEmail = false;

private bool isIncorrectPassword = false;

private bool isIncorrectEmailView = false;

[Parameter]

public EventCallback<bool> OnClose { get; set; }

[Parameter]

public EventCallback<bool> OnRegisterCall { get; set; }

private EventCallback<bool> OnIncorrectInfo { get; set; }

private Task ModalCancel()

{

return OnClose.InvokeAsync(false);

}

private Task ModalOk()

{

if (!ValidateData()) return OnIncorrectInfo.InvokeAsync(true);

return OnClose.InvokeAsync(true);

}

private Task ModalRegisterCall()

{

return OnRegisterCall.InvokeAsync(false);

}

private bool ValidateData()

{

bool validateComplete = true;

Regex regex = new Regex(@"^([\w**\.\-**]+)@([\w**\-**]+)((**\.**(\w){2,3})+)$");

Match match = regex.Match(loginOrEmail);

ResetValidateNotifications();

if (string.IsNullOrWhiteSpace(loginOrEmail)){

isIncorrectLoginOrEmail = true;

validateComplete = false;

}

if (string.IsNullOrWhiteSpace(password)){

isIncorrectPassword = true;

validateComplete = false;

}

if (!match.Success)

{

isIncorrectEmailView = true;

validateComplete = false;

}

return validateComplete;

}

private void ResetValidateNotifications()

{

isIncorrectLoginOrEmail = false;

isIncorrectPassword = false;

}

}

Компонент ModalRegisterWindow

@using System.Text.RegularExpressions

@inject IStringLocalizer<ModalRegisterWindow> \_localizer

<div class="modal fade show" id="myModal" style="display:block; background-color: rgba(10,10,10,.8);"

aria-modal="true" role="dialog">

<div class="modal-dialog py-5">

<div class="modal-content">

<div class="modal-header position-relative">

<h4 class="modal-title position-absolute start-50 top-50 translate-middle my-3" style="background-color: white;">@\_localizer["signUpLabel"]</h4>

@\*

Крестик (закрыть окно)

\*@

<button type="button" class="close position-absolute top-50 end-0 translate-middle-y mx-1 rounded px-2 py-0" @onclick="@ModalCancel">&times;</button>

</div>

<div class="modal-body container">

<div class="col justify-content-center p-3">

<div class="row-auto position-relative">

<div class="w-50 mx-auto py-2">

@\*

Имя пользователя

\*@

<input type="text" style="width: 100%;" placeholder="@\_localizer["login"]" @bind="login"/>

</div>

</div>

<div class="row-auto position-relative">

<div class="w-50 mx-auto py-2">

@\*

Почта

\*@

<input type="text" style="width: 100%;" placeholder="@\_localizer["email"]" @bind="email"/>

</div>

</div>

<div class="row-auto position-relative">

<div class="w-50 mx-auto py-2">

@\*

Пароль

\*@

<input type="password" style="width: 100%;" placeholder="@\_localizer["password"]" @bind="password"/>

</div>

@if (isIncorrectLogin)

{

<div class="alert alert-danger my-2" role="alert">

@\_localizer["incorrectLoginError"]

</div>

} else if (isIncorrectEmail)

{

<div class="alert alert-danger my-2" role="alert">

@\_localizer["incorrectEmailError"]

</div>

} else if (isIncorrectEmailView)

{

<div class="alert alert-danger my-2" role="alert">

@\_localizer["incorrectEmailViewError"]

</div>

}

else if (isIncorrectPassword)

{

<div class="alert alert-danger my-2" role="alert">

@\_localizer["incorrectPasswordError"]

</div>

}

</div>

<div class="row-auto position-relative">

<div class="w-50 mx-auto py-2 position-relative">

@\*

Войти

\*@

<button type="button" class="btn btn-outline-secondary position-absolute start-50 top-50 translate-middle-x" @onclick=@ModalOk>@\_localizer["signUpButton"]</button>

</div>

</div>

</div>

</div>

<div class="modal-footer my-2" style="flex-direction:column;">

<p>@\_localizer["hasAccount"] <a href="" @onclick="ModalLoginCall">@\_localizer["signIn"]</a></p>

</div>

</div>

</div>

</div>

@code {

private string login = string.Empty;

private string email = string.Empty;

private string password = string.Empty;

private bool isIncorrectLogin = false;

private bool isIncorrectEmail = false;

private bool isIncorrectEmailView = false;

private bool isIncorrectPassword = false;

[Parameter]

public EventCallback<bool> OnClose { get; set; }

[Parameter]

public EventCallback<bool> OnLoginCall { get; set; }

private EventCallback<bool> OnIncorrectInfo { get; set; }

private Task ModalCancel()

{

return OnClose.InvokeAsync(false);

}

private Task ModalOk()

{

if (!ValidateData()) return OnIncorrectInfo.InvokeAsync(true);

return OnClose.InvokeAsync(true);

}

private Task ModalLoginCall()

{

return OnLoginCall.InvokeAsync(false);

}

private bool ValidateData()

{

bool validateComplete = true;

Regex regex = new Regex(@"^([\w**\.\-**]+)@([\w**\-**]+)((**\.**(\w){2,3})+)$");

Match match = regex.Match(email);

ResetValidateNotifications();

if (string.IsNullOrWhiteSpace(login))

{

isIncorrectLogin = true;

validateComplete = false;

}

if (string.IsNullOrWhiteSpace(password))

{

isIncorrectPassword = true;

validateComplete = false;

}

if (string.IsNullOrWhiteSpace(email))

{

isIncorrectEmail = true;

validateComplete = false;

}

if (!match.Success)

{

isIncorrectEmailView = true;

validateComplete = false;

}

return validateComplete;

}

private void ResetValidateNotifications()

{

isIncorrectLogin = false;

isIncorrectEmail = false;

isIncorrectPassword = false;

isIncorrectEmailView = false;

}

}

Компонент CultureSelector

@using System.Globalization

@inject NavigationManager Navigation

@inject IStringLocalizer<CultureSelector> \_localizer

@rendermode InteractiveServer

<p class="m-0">

<label>

<select class="form-select" @bind="Culture">

@foreach (var culture in supportedCultures)

{

<option value="@culture">@culture.DisplayName</option>

}

</select>

</label>

</p>

@code

{

private CultureInfo[] supportedCultures = new[]

{

new CultureInfo("ru\_RU"),

new CultureInfo("en\_US"),

};

protected override void OnInitialized()

{

Culture = CultureInfo.CurrentCulture;

}

private CultureInfo Culture

{

get => CultureInfo.CurrentCulture;

set

{

if (CultureInfo.CurrentCulture != value)

{

var uri = new Uri(Navigation.Uri)

.GetComponents(UriComponents.PathAndQuery, UriFormat.Unescaped);

var cultureEscaped = Uri.EscapeDataString(value.Name);

var uriEscaped = Uri.EscapeDataString(uri);

Navigation.NavigateTo(

$"Culture/Set?culture={cultureEscaped}&redirectUri={uriEscaped}",

forceLoad: true);

}

}

}

}

Веб-страница Home

@page "/"

@using MathRecognitionService.Components.Layout

<**PageTitle**>Home</**PageTitle**>

<div>

<div class="container h-100 py-5">

<div class="row h-100 justify-content-center align-items-center">

<div class="col-6">

<**DragAndDropImageZone** />

</div>

</div>

</div>

</div>