

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

Базовая кафедра автоматизации бизнес-процессов  
(на платформе "1С:Предприятие")

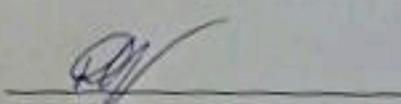
ОТЧЁТ

по учебной практике

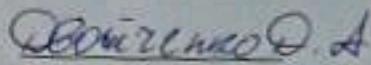
(практика по получению первичных профессиональных умений и навыков, в  
том числе первичных умений и навыков научно-исследовательской  
деятельности)

Место прохождения практики  
ООО «Строительный двор»

Выполнил  
обучающийся 3 курса,  
ПИ 10-03.1 группы



(подпись)



(ФИО)

Руководитель практики  
от института / школы

(подпись)

(ФИО)

Руководитель практики  
от профильной организации

(подпись)

(ФИО)

Тюмень, 2022

## Оглавление

<b>Введение</b> .....	<b>3</b>
<b>1.</b> Описание проделанной работы.....	<b>4</b>
<b>1.1.</b> Установка необходимых библиотек в окружение.....	<b>5</b>
<b>1.2.</b> Создание части скрипта по определению номера.....	<b>6</b>
<b>1.2.1.</b> Реализация скрипта нахождения номера .....	<b>6</b>
<b>1.2.2.</b> Оптимизационная предобработка.....	<b>7</b>
<b>1.2.3.</b> Подбор результативной нейронной сети .....	<b>8</b>
<b>1.2.4.</b> Объединение частей скриптов в групповой проект.....	<b>9</b>
<b>1.3.</b> Создание клиентской и серверной частей приложения.....	<b>10</b>
<b>1.3.1.</b> Создание модели «Отсканированные документы».....	<b>11</b>
<b>1.3.2.</b> Создание модели «Сотрудники».....	<b>12</b>
<b>1.3.3.</b> Создание модели «Отделы» .....	<b>13</b>
<b>1.3.3.</b> Реализация пользовательской части приложения.....	<b>14</b>
<b>1.4.</b> Демонстрация работы системы .....	<b>15</b>
<b>1.4.1.</b> Авторизация пользователя .....	<b>15</b>
<b>1.4.2.</b> Загрузка изображения на пользовательскую часть.....	<b>16</b>
<b>1.4.3.</b> Просмотр сохранённых документов и результатов алгоритма .....	<b>17</b>
<b>Заключение</b> .....	<b>19</b>
<b>Источники</b> .....	<b>22</b>

## **Введение**

Учебная практика проходила на предприятии ООО «Строительный двор».

Работа в предприятии организована так, что отделению разработчиков приходит заказ от одного из отделений Строительного двора. Разработка ведётся по методологиям agile/scrum. На проект выделяется команда разработчиков, обладающих необходимыми компетенциями, и ведут разработку заказанного продукта. Ежедневно, в команде проходят daily meeting-и, на которых обсуждается то, что было сделано, что предстоит сделать и с чем требуется помощь. По понедельникам команда созванивается с заказчиком и отчитывается о проделанной работе. Проходившие практику так же принимали участие во встречах, где рассказывали о проделанной работе.

Цели прохождения учебной практики – получение первоначального профессионального опыта, развитие общих и профессиональных компетенций, проверка готовности к самостоятельной трудовой деятельности, получение опыта решения практических задач.

Для этого было необходимо:

- Изучить документацию необходимых библиотек
- Изучить принципы «Компьютерного зрения»
- Решить реальную задачу;

Руководителем предприятия мне была поставлена задача: разработать систему хранения и определения основных данных документа типа «Товарная накладная» с изображения документа, в рамках проекта сервиса по определению типа документа с изображения.

# 1 Описание проделанной работы

Задание: разработать систему хранения и определения основных данных документа типа «Товарная накладная» с изображения документа.

Для разработки системы было необходимо:

- Изучить документацию необходимых библиотек
- Изучить принципы «Компьютерного зрения»
- Решить реальную задачу

В процессе выполнения поставленной задачи, мною были разработаны алгоритм определения номера документа типа «Товарная накладная» с изображения, посредством отбора замкнутых объектов на изображении по ключевым характеристикам и вычислений нейронной сети. Произведены оптимизация алгоритма и способы загрузки изображений в систему и последующего их хранения, с вычисленной информацией. В рамках прохождения практики, меня приглашали на встречи команды, по принципу методологии ягит, где команда, выделенная для разработки сервиса обсуждала проделанную работу и ставила новые цели на последующий рабочий день.

## 1.1 Установка необходимых библиотек в окружение

Для начала разработки потребовалась установка на свой персональный компьютер необходимых для работы библиотек языка python. В частности, были установлены numpy, matplotlib, pandas, opencv2, easyocr, pytesseract, pytorch, Image. Так же, использовалась библиотека Django, которая была установлена на момент начала работы. Так же, для удобства тренировки моделей нейронной сети было установлено программное обеспечение anaconda, содержащее Jupiter notebook. Для работы с пользовательской частью приложения потребовались node js и расширения браузера, для упрощения работы с React компонентами.

## 1.2 Реализация скрипта нахождения номера

### 1.2.1 Создание части скрипта по определению номера

Для нахождения месторасположения номера на документе, в следствии прочтения документации, было принято решение не читать весь документ нейросетью, так как это слишком трудоёмкий процесс, а обозначить на изображении замкнутые контуры и произвести отбор замкнутых фигур по ключевым параметрам. Этими параметрами стали количество контуров, равное четырём, для нахождения прямоугольников и размер, в зависимости от ориентации, отбираемый посредством соотношения размеров фигур, к размерам документа. В результате нахождения среднего диапазона значений соотношения рамки содержащей номер документа и размера документа, алгоритм стал отбирать среди всех контуров на изображении один конкретный. Но, появились проблемы с производительностью и распознаванием границ. Границы документов не распознавались на изображениях плохого качества, а при переборе всех найденных контуров выяснилось, что на изображении их, как правило, свыше сотни и перебор всех контуров является трудоёмким процессом. Было принято решить эти проблемы путём предобработки изображений.

### 1.2.2 Оптимизационная предобработка

Для повышения эффективности алгоритма и оптимизации производительности было принято решение производить предобработку изображений документов.

Первой задачей стало решение количества найденных контуров на изображении, что становилось достаточно трудоёмкой задачей, когда их количество достигало несколько сотен. Посредством библиотеки open cv2, я обрезал входящие изображения по краям, до таких областей, где встречается часть с номером документа, отчего количество найденных контуров сократилось, в среднем, в несколько раз, а время поиска контуров сократилось в среднем с 0,6 секунд, до 0,2 секунд.

Второй задачей начальной предобработки стало повышение качества изображений, на которых алгоритм не мог найти необходимые контуры, из за того, что помарки и точки, оставшиеся от некачественной распечатки документа, перекрывали границы контуров искомой рамки. Изображения прошли размытие по Гауссу и бинаризацию, приводящую более тёмные пиксели к чёрным, а более светлые к белым. Это способствовало нахождению искомой рамки в 99,5% случаев, при 87% результативности без обработки.

Так же, для предотвращения нагрузки сети, посредством считывания лишнего текста, я обрезал найденную рамку, а далее и её часть с номером в отдельное изображение и передал его в функцию, содержащую нейронную сеть.

### 1.2.3 Подбор результативной нейронной сети

Для поиска самой подходящей под мой случай нейронной сети, я сравнил три различных случая. Этими нейронными сетями стали easyocr, pytesseract и Sequential тренированная на базе данных mnist, содержащей изображения рукописных чисел и их значений.

Так же, для каждого случая понадобились предобработки выходных значений. Например, easyocr часто ставила вместо «1» - «+» или «]» вместо «7». Я произвёл предобработку выходных значений, а так же, предобработку картинок, поэкспериментировав с различными степенями размытия по Гауссу, бинаризации, и градациями серого и выбрал самые результативные предобработки для каждой нейронной сети. В результате, тренировка Sequential не увенчалась успехом и нейронная сеть показывала результативность в районе 40%. Pytesseract справился на 89%, так как часто путал значения «8» и «6». Easyocr показал самую большую результативность, набрав 94,5%, не справившись лишь с дефектными изображениями, распознать номер на которых не способен и человеческий глаз. В итоге, оптимальной сетью для распознавания текста на изображении, стал easyocr.

#### **1.2.4 Объединение частей скриптов в групповой проект**

При достижении максимальной результативности наших частей алгоритма, мне и проходящем практику со мной Большакову Григорию дали задание объединить части проекта в одну и реализовать получение всех результатов алгоритма посредством вызова одной функции. Эта часть не вызвала у меня затруднений, а сама работа заключалась лишь в приведении структуры поступления данных на вход к единому виду.

### 1.3 Создание клиентской и серверной частей приложения

После завершения работы с алгоритмом для поиска основной информации на изображении документа, передо мной стояла задача разработать информационную систему, для загрузки изображений в базу данных, прохождение изображения через алгоритм и хранение полученных результатов, вместе с информацией о сотруднике, загрузившем изображение и дочернего отдела компании, из которого это было сделано.

Для реализации серверной части web приложения и реализации работы с базами данных, мною была выбрана библиотека Python Django, содержащая базу sql и средства работы с ней по умолчанию, а так же rest api, для реализации запросов вида клиент-сервер и наоборот.

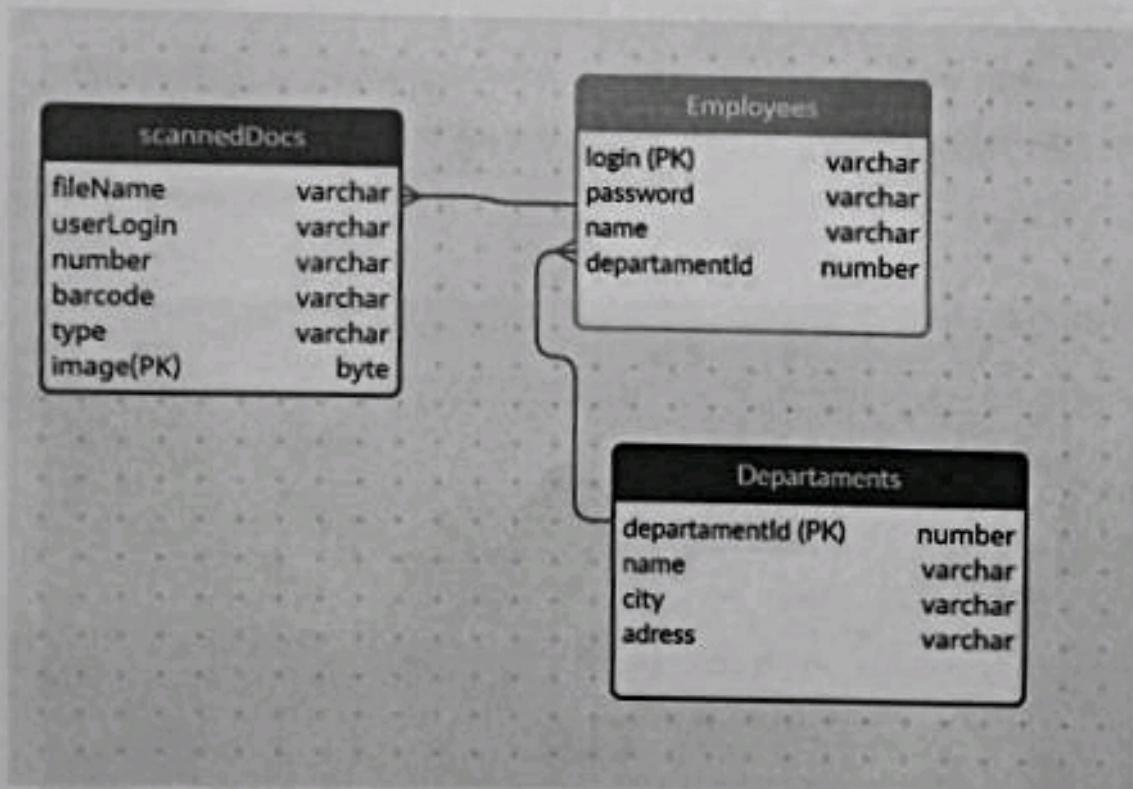


Рис 1 – схема базы данных проекта

### 1.3.1 Создание модели «Отсканированные документы»

Была создана модель базы данных «ScannedDocs»(Отсканированные документы).

Поля содержащиеся в этой таблице:

- 1 fileName – Имя файла, при загрузке пользователем.
- 2 userLogin – логин сотрудника, загрузившего изображение в систему.
- 3 number – номер документа, полученный из алгоритма.
- 4 barcode – декодированный номер штрихкода, полученный из алгоритма.
- 5 type – тип штрих-кодировки, полученный из алгоритма.
- 6 image – Изображение и путь до него, сохранённого в файловой системе корневой папки проекта.

Для связи с таблицей сотрудников используется поле логин. Полем содержащим уникальные значения является «Изображение». При загрузке изображения с наименованием, содержащимся в базе данных – Django присваивает файлу новое наименование по паттерну «Имя файла + случайный набор символов + .jpg».

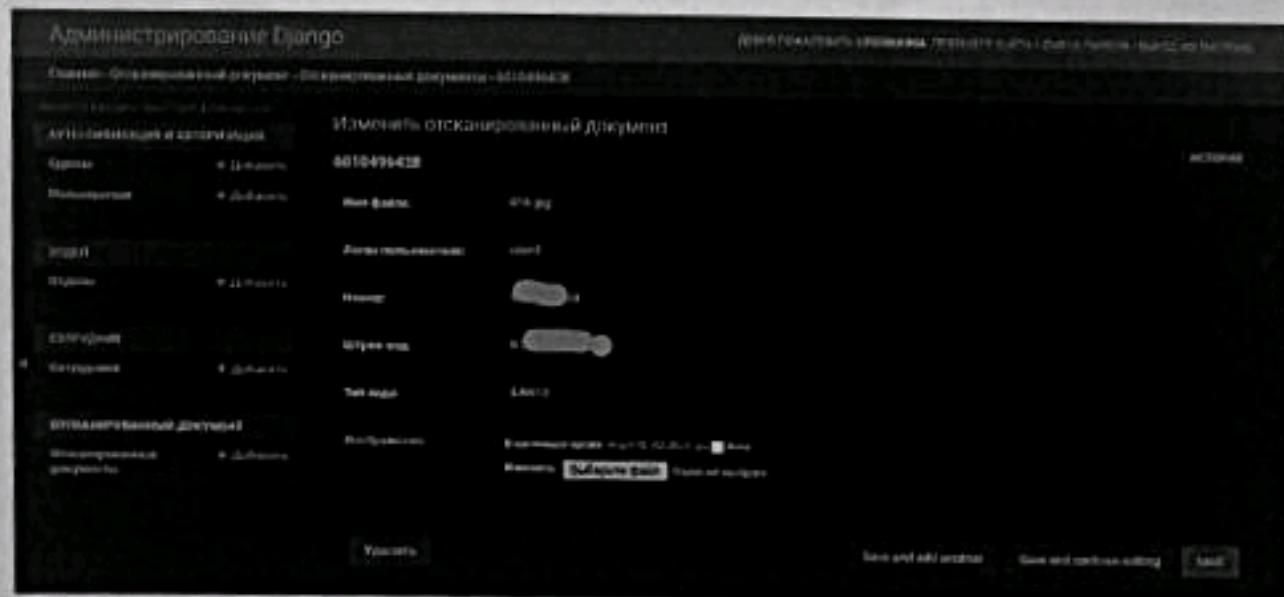


Рис 2 – модель базы данных «Отсканированные документы»

### 1.3.2 Создание модели «Сотрудники»

Была создана модель базы данных «employees»(Сотрудники).

Поля содержащиеся в этой таблице:

- 1 login – Логин пользователя, для входа в систему.
- 2 password – Пароль пользователя.
- 3 name – содержащее ФИО сотрудника.
- 4 departmentId – идентификатор отдела, к которому принадлежит сотрудник.

Полем содержащим уникальные значения в этой таблице является поле login. Регистрация пользователей с одинаковым логином блокируется. Пароль не хешируется, так как для корпоративного аккаунта, необходимого лишь для классификации документов по принадлежности к отделам это лишняя нагрузка. Пароли администраторов серверной системы admin-панели Django хешируются и содержатся во встроенной модели «Пользователи».

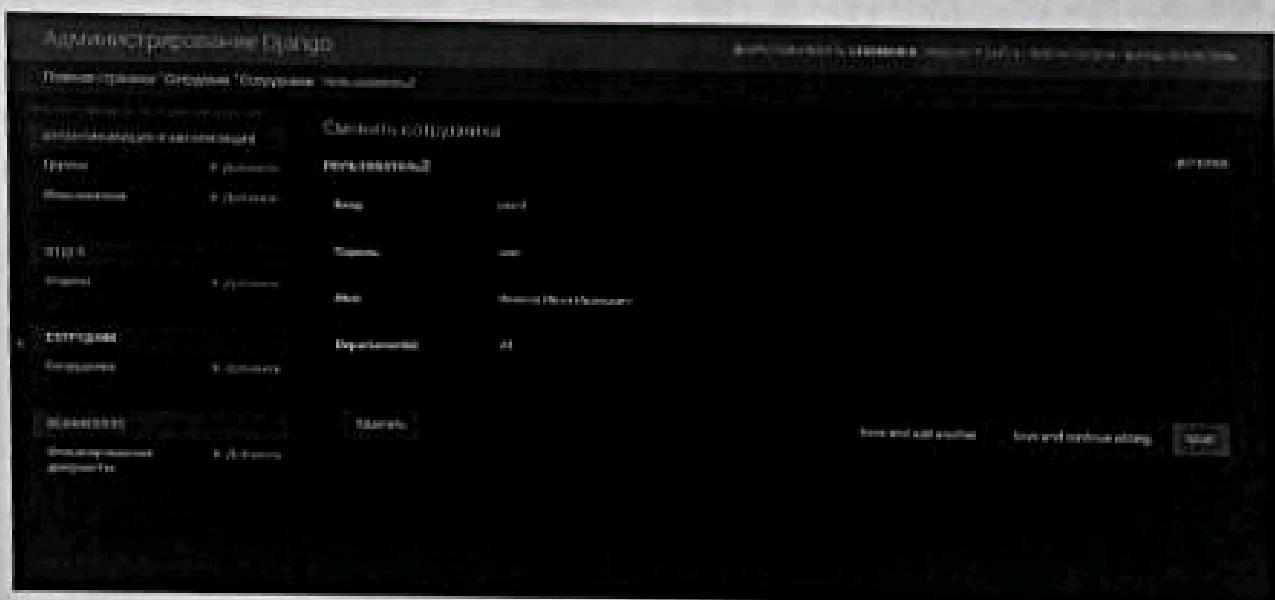


Рис 3 – модель базы данных «Сотрудники»

### 1.3.3 Создание модели «Отделы»

Была создана модель базы данных «departments»(отделения).

Поля содержащиеся в этой таблице:

- 1 departmentId – Уникальный идентификатор дочернего отделения.
- 2 name – наименование дочернего предприятия.
- 3 city – город в котором располагается отделение.
- 4 adress – Адрес, по которому располагается отделение.

Для связи таблицы с таблицей сотрудников используется поле «departmentId».

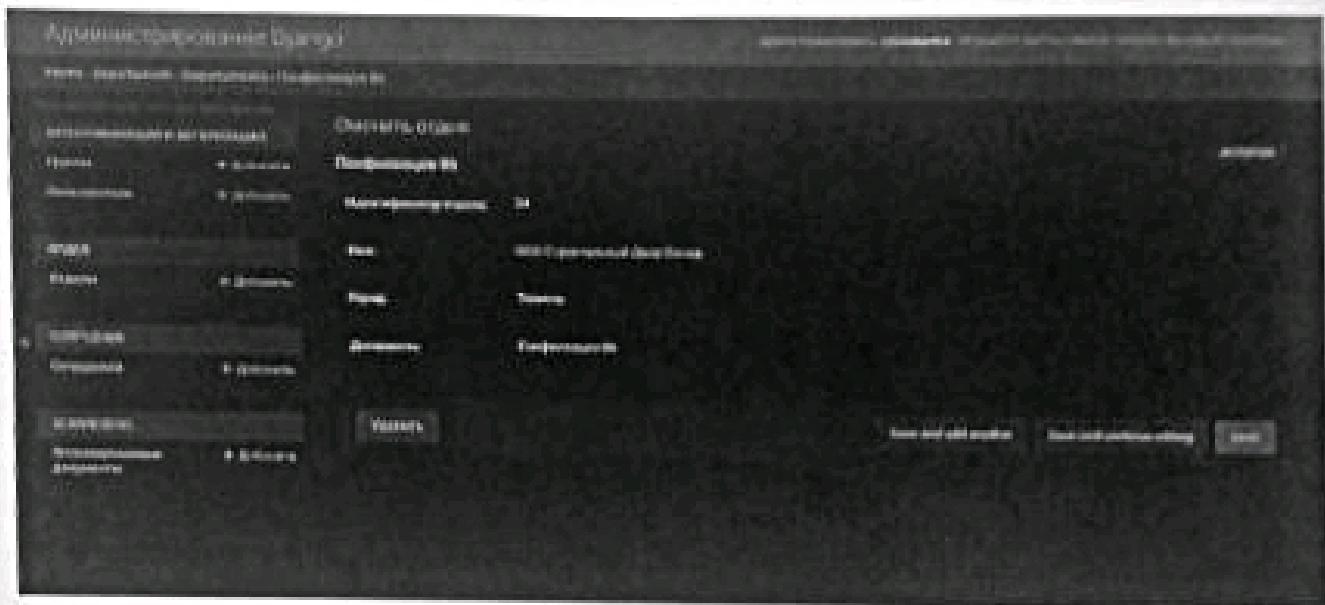


Рис 4 – модель базы данных «Отделы»

#### **1.3.4 Реализация пользовательской части приложения**

В конечном итоге, передо мной стояла задача создать клиентскую часть web приложения, и связать её с серверной частью, посредством передачи данных через rest api.

Изначально, я создал React компонент регистрации-авторизации пользователя, производящего запрос с поиском существующего логина, сверяющего введённые данные, а при отсутствии пользователя с введённым логином, записывало бы в систему нового пользователя.

Так же, мною были написаны компонент для добавления изображения в базу, всплывающее окно, показывающее содержащиеся в базе данных изображения и всплывающее окно с изображением.

Для передачи и изменения данных были созданы арі-представления в приложениях Django и точки арі для соединения клиентской и серверной частей приложения.

## 1.4 Демонстрация работы системы

### 1.4.1 Авторизация пользователя

React компонент регистрации-авторизации пользователя, принципиальный запрос с поиском существующего логина, сверяющий введённые данные, а при отсутствии пользователя с введённым логином, записывающей в систему нового пользователя.

В списке отделений содержатся все результаты запроса к получению всех объектов модели отделений.

При клике на клавишу «войти» пользователь попадает на страницу загруженных изображений.

При неверно введённых данных показывается соответствующее сообщение:

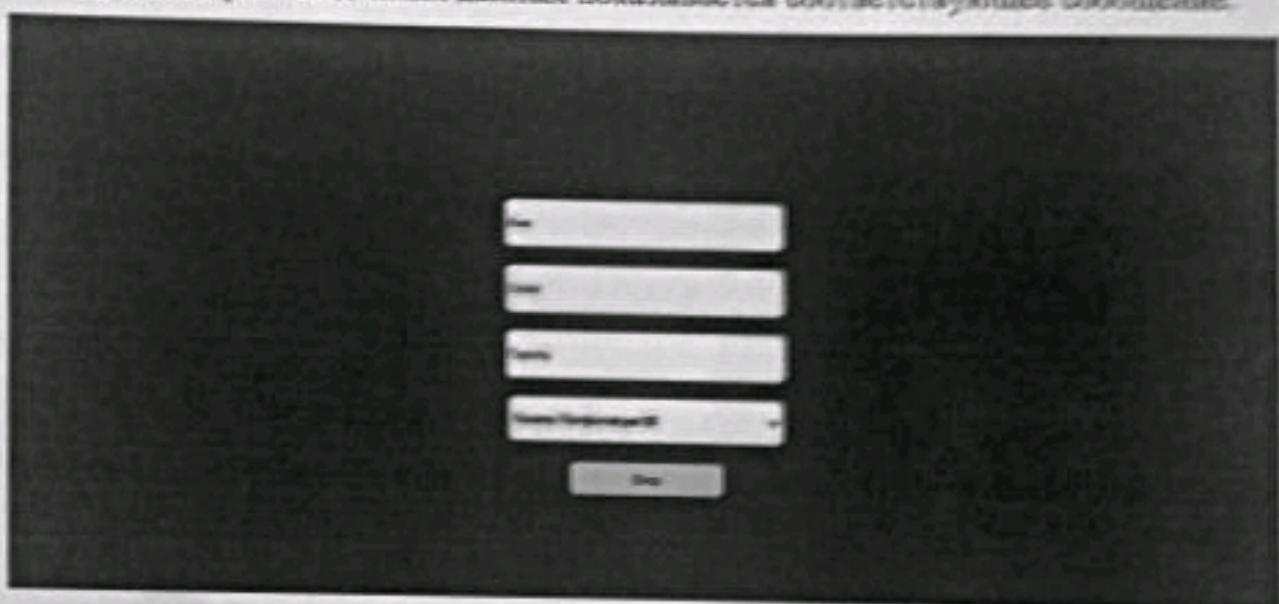


Рис 5 – компонент авторизации

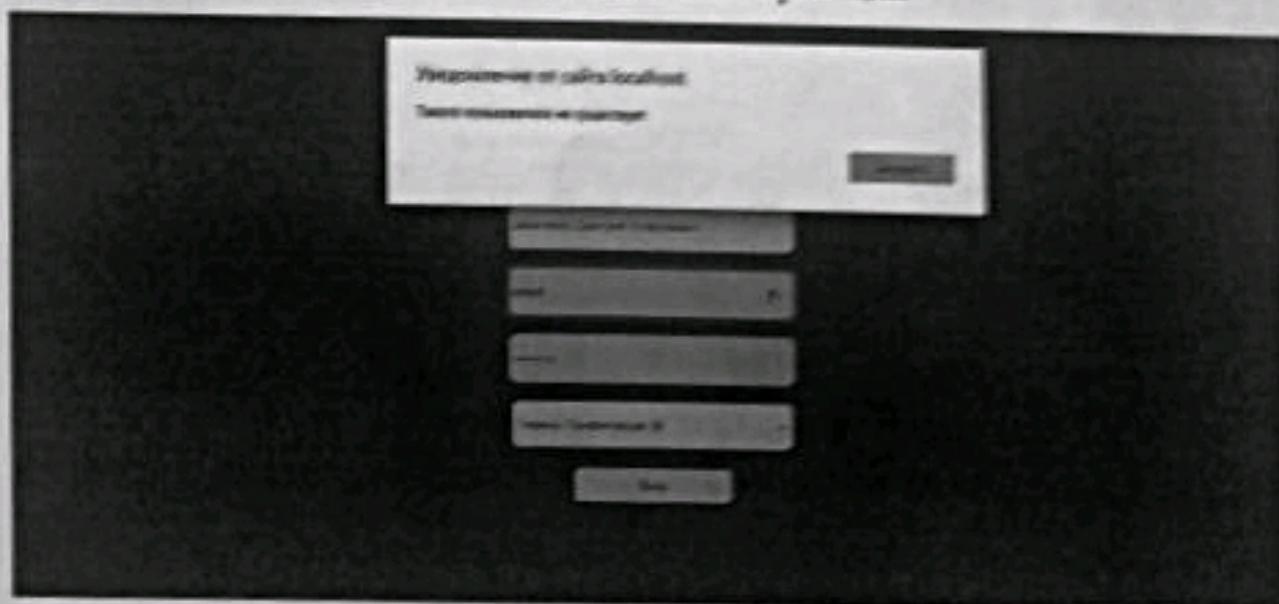


Рис 6 – уведомление о неверно введенных данных

#### 1.4.2 Загрузка изображения на пользовательскую часть

При выборе файла, пользователю показывается предварительный просмотр загружаемого изображения, а при нажатии на кнопку «Отправить» интерфейс показывает пользователю информацию, полученную из алгоритма.

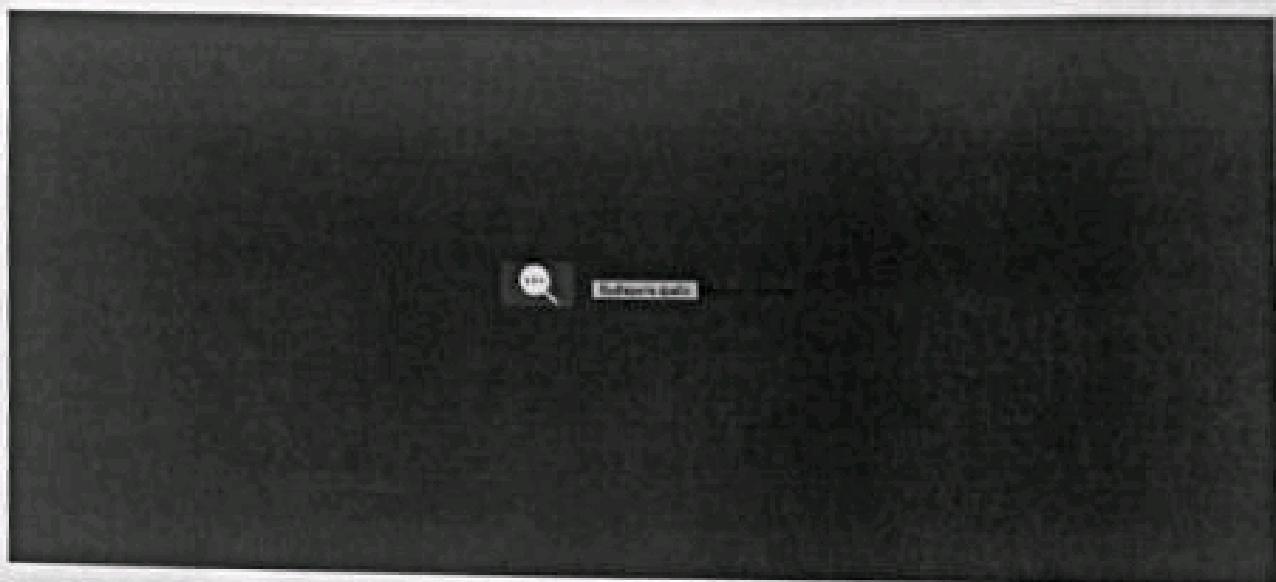


Рис 7 – интерфейс загрузки документа

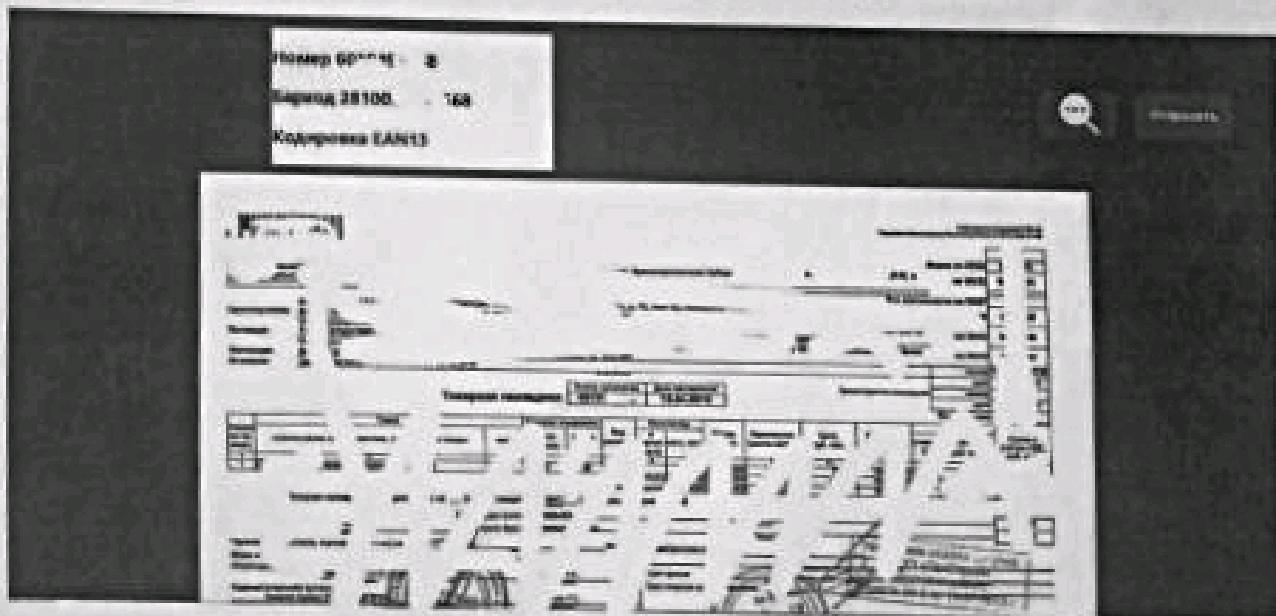


Рис 8 – интерфейс предварительного осмотра документа

#### 1.4.3 Просмотр сохранённых документов и результатов алгоритма

При клике на кнопку с изображением лупы, появляется всплывающее окно, содержащее список полей с именем пользователя, добавившего документ и его основной информацией.

При клике на одно из полей, откроется всплывающее окно, с соответствующим ему сохранённым изображением.

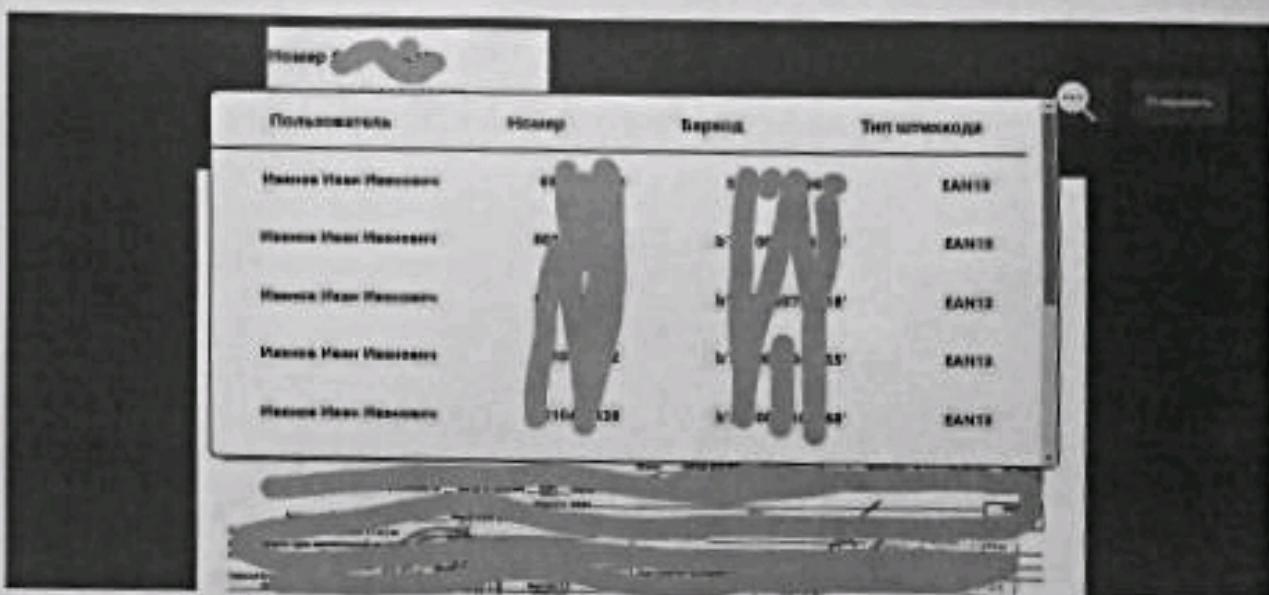


Рис 8 – панель вывода сохранённых документов

## **Заключение**

В результате прохождения практики мной был получен первоначальный профессиональный опыт, произошло развитие общих и профессиональных компетенций, так же я получил опыт решения практических задач.

Что касается решения поставленной мне задачи – данная мне задача выполнена, а требуемый функционал реализован. Наша работа будет использована в дальнейшей разработке проекта, как сборник решений некоторых трудоёмких задач. В будущем планируется добавить распознаватель ортогонального положения документа и его типа.

## Источники

<https://www.comp.nus.edu.sg/~cs4243/doc/opencv.pdf> - Документация библиотеки open cv

<https://www.digitalocean.com/community/tutorials/how-to-build-a-modern-web-application-to-manage-customer-information-with-django-and-react-on-ubuntu-18-04-ru> - пример настройки и использования rest api.

<https://pytesseract.readthedocs.io/en/latest/> - документация pytesseract

<https://ru-keras.com/home/> - документация keras

Защищен

20

Результаты экзамена / зачета

(подпись руководителя практики от института)