

Оглавление

Введение	3
Постановка задачи	5
1. Обзор использованных в интеграции инструментов	6
1.1. Особенности Cloud Platform ServiceNow.....	6
1.2. Анализ и классификация чеков	9
1.3. Анализ существующих OCR сервисов.....	11
2. Использование информации с чеков в корпоративной системе	18
2.1. Внедрение Google Vision AI.....	18
2.2. Внедрение Tesseract JS в корпоративную систему	24
3. Анализ результатов	26
Заключение.....	29
Литература	30

Введение

Процессы – это фундаментальная часть любого, будь то малого, среднего или крупного бизнеса. Они могут быть рассмотрены как алгоритмы, определяющие пути решения возникающих задач. При их построении, используется концепция BPM – Business Process Management – практика проектирования, выполнения, мониторинга и оптимизации бизнес-процессов. Если рассматривать критерий автоматизации процессов более подробно, то необходимо обратиться к специальному программному обеспечению, которое поддерживает разработку технологических решений для выполнения бизнес-процессов. Одним из таких инструментов является Cloud Platform ServiceNow (SN) [1].

Основное предложение ServiceNow – это готовая платформа, которая позволяет упростить и автоматизировать рутинные рабочие задачи и обеспечить плавное выполнение проектов с использованием единой модели данных. Сервис построен на инструментальной панели управления событиями (Event Management Dashboard), наглядно отображая состояние каждого элемента в различных сферах компании и предлагая быстрый доступ к оптимальным решениям. Это максимально абстрактное понятие включает возможность организовать любую необходимую работу, будь то обмен данными между сотрудниками, клиентская поддержка, ведение базы данных и документации. Гибкая система настроек помогает организовать работу для любого типа бизнеса, в любой среде, где есть доступ к интернету. Ключевым фактором является то, что SN может стать отличным инструментом для конфигурации бизнес процессов, связанных с электронным документооборотом, так как обладает необходимыми для этого свойствами:

- Централизованная организация документов в стандартизированных файловых структурах и форматах;
- Защита документов в соответствии со списком управления доступом;
- Простота в распространении нужных документов среди работников;
- Хранение и доступ к информации для более эффективного исполь-

зования.

Но несмотря на полный спектр средств, направленных на автоматизацию и оптимизацию процессов связанных с документооборотом, актуальной остаётся проблема конвертации печатных документов в электронный формат. Именно поэтому в данной работе разработана система распознавания информации на изображениях кассовых чеков на основе возможностей платформы ServiceNow по внедрению преобразований бумажных документов в цифровые активы. Это облегчает и автоматизирует создание финансовой отчетности, так как пропадает необходимость вручную вносить в базу данных информацию о расходах сотрудников компании. Сотрудник сам загружает изображение чека в систему, на основе чего генерируются отчеты. В качестве инструментов для распознавания данных были использованы Google Vision AI API [2] и JavaScript [3] библиотека Tesseract.JS [4]. Также для проверки правописания слов была подключена библиотека SpellChecker.JS [5].

Постановка задачи

Целью выпускной квалификационной работы является расширение базового функционала корпоративной ServiceNow платформы путём внедрения OCR инструментов для распознавания текста на изображениях кассовых чеков. В качестве таких инструментов были выбраны одни из самых популярных и пользующихся спросом сервисов: Google Vision AI и Tesseract.JS.

Можно выделить следующие этапы выполнения работы:

- Изучение особенностей и возможностей Cloud Platform ServiceNow;
- Сбор и анализ тестовых данных – изображений чеков разных типов;
- Анализ существующих OCR сервисов и инструментов для распознавания текста на изображениях чеков;
- Определение оптимального сервиса и изучение его методов;
- Внедрение выбранных OCR сервисов в ServiceNow;
- Организация системы по распознаванию информации на чеках ;
- Тестирование разработанного приложения и анализ полученных результатов.

1. Обзор использованных в интеграции инструментов

1.1. Особенности Cloud Platform ServiceNow

ServiceNow - это полностью облачное решение, которое оптимизирует и согласовывает все внутренние процессы компании без необходимости приобретать или обслуживать дорогую инфраструктуру или дополнительное программное обеспечение. Многоэкземплярная архитектура лежит в основе работы платформы: для каждого сотрудника создается уникальный экземпляр (учётная запись в платформе), который поддерживает отдельный стек ресурсов. Это дает возможность реагировать на конкретные потребности каждого пользователя, что позволяет наладить работу на индивидуальной основе.

Также стоит отметить, что в самой системе существует огромное количество модулей, отвечающих за тот или иной бизнес-процесс. Например, существует модуль, который позволяет пользователю заказать новое оборудование или создать инцидент. Эти приложения представляют собой таблицы со своей уникальной структурой и логикой. Пользователи с правами системного администратора могут ограничивать доступ, вносить изменения в логику и структуру приложений, конфигурировать таблицы и записи.

1.1.1. Некоторые понятия, использованные в работе

Триггер - хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено действием по модификации данных.

REST Message используют протоколы HTTP в качестве средства связи между клиентом и сервером. Клиент отправляет сообщение в форме HTTP-запроса, а сервер отвечает в виде HTTP-ответа.

REST API - это общие средства взаимодействия приложения/сайта с

сервером посредством протокола HTTP.

KPI (Key Performance Indicator) — это показатель достижения успеха в определенной деятельности или в достижении определенных целей.

OCR (optical character recognition) - это технология, позволяющая преобразовывать различные типы документов, в редактируемые форматы. Например изображение с текстом в текстовый формат.

SDK - набор средств разработки, который позволяет специалистам по программному обеспечению создавать приложения.

AI -приложение с использованием искусственного интеллекта.

1.1.2. Приложения SN, задействованные в разработке

- Business Rules (BR) [6] представлены в виде серверного кода, который запускается, когда запись из таблицы отображается/вставляется/обновляется/удаляется, а также когда запрашивается непосредственно сама таблица. В основном BR принято использовать для выполнения таких задач, как автоматические изменения значений в полях формы при соблюдении определенных условий. Возможно настроить отправку всевозможных уведомлений. В интеграции используется для отправки REST запроса, получения ответа от сервера в виде JSON объекта и дальнейшей работы с ним. Триггером служит добавление изображения на форму.
- Script Includes [7] используются для хранения JavaScript кода, который действует на сервере платформы. Каждая запись данного модуля хранит функции и классы для глобального использования. Алгоритмы кодирования изображения в формат BASE64 и формирование тела JSON запроса было решено вынести в Script Include отдельным классом. Этот шаг предоставил возможность облегчить читаемость созданного BR кода.
- REST (Representational State Transfer) [8] представляет собой архитектуру, обеспечивающую связь между компьютерными системами в Интернете, облегчая для них взаимодействие друг с другом.

Платформа Now предоставляет различные REST API, которые по умолчанию активны. Эти API предоставляют возможность взаимодействия с различными функциями ServiceNow в вашем приложении. Такая функциональность включает в себя возможность выполнения операций создания/чтения/обновления/удаления в существующих таблицах, вставки данных, извлечения информации и запуска преобразований для базы данных.

- Client Scripts [9] позволяют системе запускать JavaScript код на клиентской стороне (веб-браузере), когда форма загружается, поле меняет значение, или когда изменения на форме сохраняются. С помощью данного модуля реализована интеграция с библиотекой Tesseract.JS.
- Service Portal [10] — один из многочисленных модулей ServiceNow, позволяющий разработчикам и администраторам создавать действительно удобный пользовательский интерфейс, с интуитивно понятным управлением. Он взаимодействует с существующими компонентами ServiceNow, поэтому пользователи могут получать доступ к определенным функциям платформы с помощью него, например воспользоваться виджетами для распознавания текста.
- UI Scripts [11] предоставляют возможность не только хранить клиентский JavaScript, но и запускать его «из браузера», встраивать в HTML код страницы. Принцип работы схож с «Script Includes», только применяется не для серверной, а для клиентской части. В рамках поставленной задачи, используется для хранения библиотек Tesseract.JS и Spell-checker.JS.
- UI Pages [12] могут использоваться для создания и отображения форм, диалоговых окон, списков и других компонентов пользовательского интерфейса.
- Performance Analytics [13] - это платформенное решение для оптимизации процессов, позволяющее создавать информационные панели управления, составлять отчеты по KPI и метрикам, а также отвечать на ключевые бизнес-вопросы, помогая повысить качество

и сократить расходы на предоставление услуг,

- System Properties [14] модуль задействован для того, чтобы скрыть в коде уникальные ключи записей, пароли и другую конфиденциальную информацию,
- UI Action [15] включают кнопки, ссылки и элементы контекстного меню в формах и списках, чтобы сделать пользовательский интерфейс более интерактивным.

1.1.3. Преимущества использования данного облачного сервиса

- Платформа позволяет разрабатывать с использованием новейших веб-технологий и поддерживает интеграцию со сторонними приложениями. Служба поддержки клиентов ServiceNow эффективно решает любые вопросы/заявки.
- Для того, чтобы иметь доступ к личному экземпляру платформы, необходимо лишь создать учётную запись на официальном портале.
- ServiceNow отлично справляется с большинством административных обязанностей, включающих в себя документирование и организацию проектных задач, обеспечивая своевременную связь и соблюдение всех нюансов и тонкостей. Данное преимущество позволяет автоматизировать рутинные бизнес-процессы (создание запросов, отчётов, ведение учёта времени) и тем самым сохранить время для по-настоящему важных дел, например для создания итогового продукта,
- Подробная техническая документация доступна на официальном сайте.

1.2. Анализ и классификация чеков

Федеральным законом РФ установлено два вида платёжных документов: обычный (на бумажной кассовой ленте) и электронный. Первый

выдаётся при принятии оплаты в торговом пункте, второй направляется через SMS-сообщение или на e-mail при любой онлайн покупке. В данной работе рассмотрены оба типа кассовых чеков, так как принципы распознавания каждого из них одинаковы. Каждый чек должен включать в себя следующие реквизиты:

- Наименование организации либо ИП;
- Сумма, на которую пробит чек;
- Дата и время транзакции;
- Номер документа (чека);
- Номер кассового аппарата;
- Номер ЭКЛЗ;
- Номер КПК по порядку;
- значение КПК - контрольный проверочный код.

На изображении (рис. 1) красным отмечена ключевая информация, которая в последующем будет храниться в виде записей в самой платформе. Зелёный указывает на необязательные, но также полезные данные.

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "ЯНДЕКС.ТАКСИ"
ИНН: 7704340310
taxi.yandex.ru

Кассовый чек. Приход


N 653 N АВТ 101922631
Смена N 269 25.11.19 02:05

N	Наим. пр.	Цена за ед. пр.	Колич. пр.	НДС	Сум. пр.
1.	Перевозка пассажиров и багажа	1069,00	1,000	—	1069,00

ИНН Поставщика: 616804585963

СУФМА —	0,00
АГЕНТ	
Итого	1069,00
ЭЛЕКТРОННЫМИ	1069,00

N ККТ: 0004078447026396 N ФД: 158117
N ФН: 9284000100229960 ФП: 2598436901
СНО: ОСН ЗН ККТ: 00000003820040051501



Эл. адр. получателя: +79281038433 Эл. адр. отправителя: support@taxi.yandex.ru
Сайт ФНС: nalog.ru

Рис. 1. Важные реквизиты кассового чека

Проанализировав данные о покупках товаров и услуг, совершаемых внутри компаний, можно выделить основные типы, пользующиеся спросом

(рис. 2). Для каждого из них характерен определённый словарный состав. Проверив документ на наличие слов, подтверждающих принадлежность к тому или иному типу, можно сделать вывод и о самом классе денежной транзакции. Например, если в кассовом чеке встречаются названия продуктов питания, то логично будет заключить, что данные расходы можно определить в категорию "Питание".

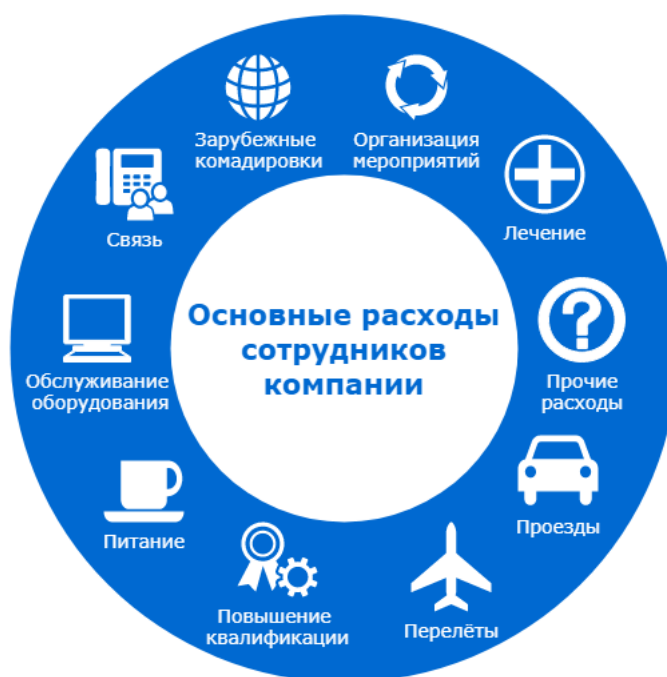


Рис. 2. Классификация расходов

1.3. Анализ существующих OCR сервисов

OCR (optical character recognition) расшифровывается как «Оптическое распознавание символов». Это процесс преобразования изображения в машинный код. Например, мы можем получить отсканированное изображение книги и использовать технологию распознавания текста, чтобы прочитать изображение и вывести текст в формате, который мы можем использовать на любом вычислительном устройстве.

Критерии, по которым оценивались предложенные на рассмотрение сервисы:

- Качество распознанного текста;

- Экономическая обоснованность;
- Возможность интегрирования в платформу;
- Распознавание текста, включающего в себя более одного языка.

Для получения более наглядных результатов, сервисам было предложено справиться с данным тестовым изображением (рис. 3).

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "ЯНДЕКС.ТАКСИ"
ИНН: 7704340310
taxi.yandex.ru

Кассовый чек. Приход

N 653 N АВТ 101922631
Смена N 269 25.11.19 02:05

N	Наим. пр.	Цена за ед. пр.	Колич. пр.	НДС	Сум. пр.
1.	Перевозка пассажиров и багажа ИНН Поставщика: 616804585963	1069,00	1,000	—	1069,00
СУММА —					0,00
АГЕНТ					
Итого					1069,00
ЭЛЕКТРОННЫМИ					1069,00

N ККТ: 0004078447026396 N ФД: 158117
N ФН: 9284000100229960 ФП: 2598436901
СНО: ОСН ЗН ККТ: 0000003820040051501



Эл. адр. получателя: +79281038433 Эл. адр. отправителя: support@taxi.yandex.ru
Сайт ФНС: nalog.ru

Рис. 3. Изображение кассового чека

1.3.1. Tesseract JS

Tesseract.JS - JavaScript библиотека для распознавания текста на изображениях, поддерживающая огромное количество языков, включая русский и английский языки.

Текст полученный, в результате использования Tesseract.JS (рис. 4).

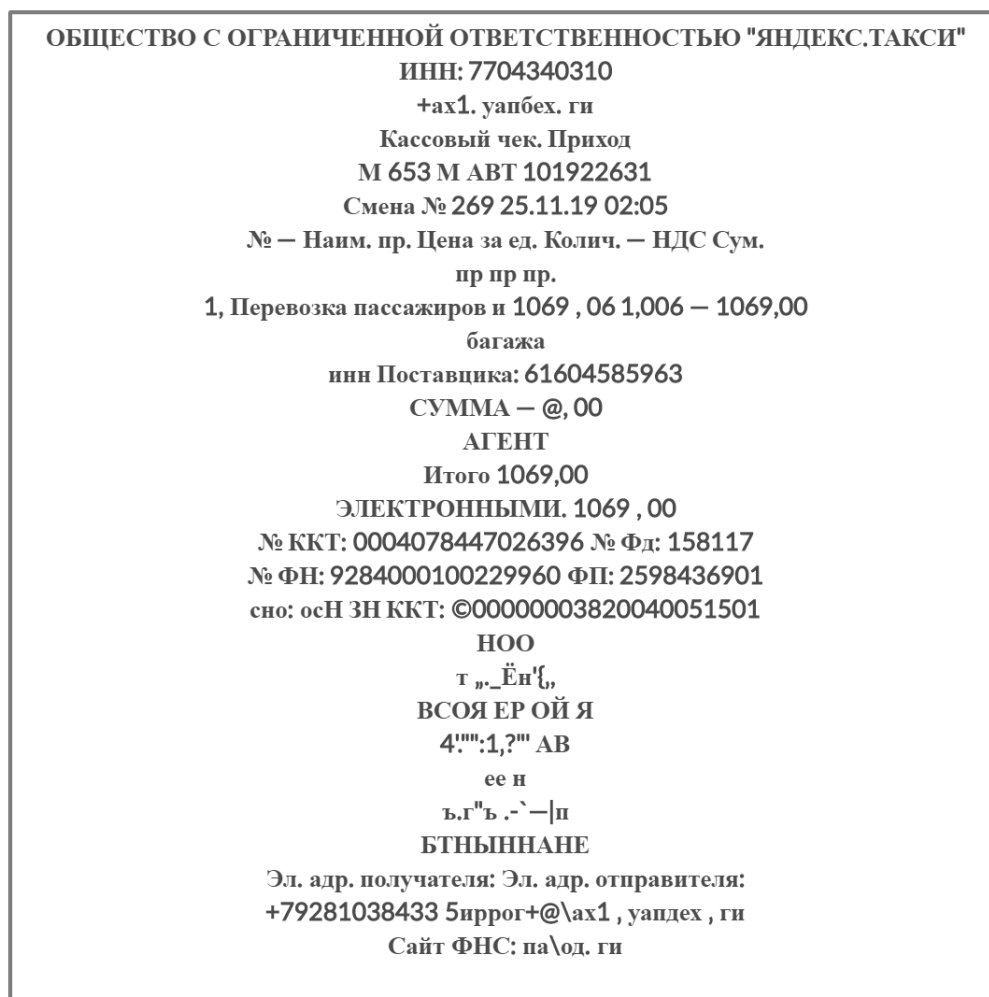


Рис. 4. Распознавание чека с помощью Tesseract.JS

Достоинства:

- Автоматическая ориентация текста;
- Открытый код;
- Бесплатное использование;
- Доступный интерфейс для определения пространственных границ слов; абзацев,
- Может работать как в браузере, так и на сервере с NodeJS.

Недостатки:

- Невозможность распознавания текста, включающего в себя более одного языка;
- Зашумлённый фон и нестандартные шрифты существенно снижа-

ют точность обработки;

- Ограниченный спектр возможностей.

Анализируя полученные результаты, можно сделать выводы, данная библиотека - лучшее решение для распознавания чеков, так как возможность запускать скрипты на клиентской стороне является выигрышной при интеграции с платформой SN. Также стоит отметить, что с экономической точки зрения выгодно использовать этот инструмент, потому что он не требует дополнительных вложений (например приобретения подписки). Но несмотря на превалирующие достоинства, неспособность распознавать более одного языка одновременно препятствует получению хорошего результата.

1.3.2. ABBYY Cloud OCR FineReader Engine

ABBYY FineReader Engine – многофункциональный инструмент разработчика, который позволяет встраивать в приложения интеллектуальные технологии распознавания данных. С помощью OCR на основе технологий искусственного интеллекта вы можете создавать приложения с функциями качественного распознавания информации из документов, изображений, фотографий, скриншотов, мониторов и дисплеев, определения типа документа, конвертации сканированных документов в файлы форматов Word, Excel и PDF с возможностью поиска.

Текст полученный, в результате использования FineReader Engine (рис. 5).

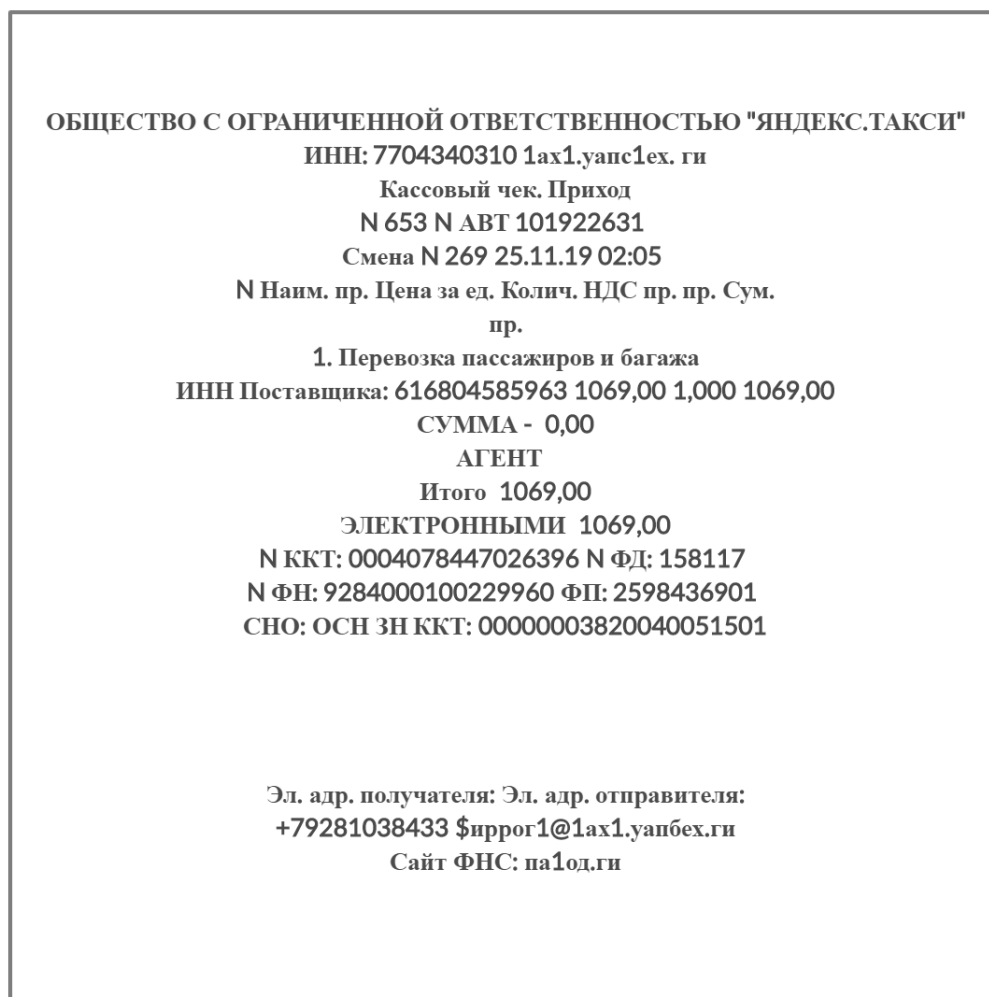


Рис. 5. Распознавание чека с помощью FineReader Engine

Достоинства:

- Высокая точность распознавания;
- Полный спектр технологий распознавания;
- Инструменты для обработки PDF;
- Поддержка множества языков;
- Распознавание текста, включающего в себя более одного языка.

Недостатки:

- Высокая стоимость подписки.

Данный сервис показывает высокое качество распознанного текста. Также стоит отметить удобство при работе с интеграцией: клиентская программа, передавая изображения с помощью одного или нескольких POST-

запросов, формирует задание на сервере. После того, как задание сформировано, необходимо отправить его на обработку, указав настройки обработки. Кроме того, автоматическое определение языка позволяет сервису распознавать разноязычный текст, что делает его выигрышным по сравнению с Tesseract.JS. Единственным недостатком становится ощутимая плата за использование данного инструмента, подтверждающая его экономическую нерациональность.

1.3.3. Google Vision AI

API Google Cloud Vision позволяет разработчикам легко интегрировать функции обнаружения объектов в приложение. Мощные, предварительно обученные модели машинного обучения достойно справляются с такими задачами, как распознавание лиц и ориентиров, оптическое распознавание символов (OCR), классификация изображений по множественным категориям.

Текст, полученный в результате использования Google Vision AI представлен на Рисунке 6.



Рис. 6. Распознавание чека с помощью Google Vision AI

Достоинства:

- Высокая точность распознавания;
- Простота использования;
- Распознавание текста, включающего в себя более одного языка;
- Бесплатное использование на протяжении года;
- Поддержка множества языков.

Недостатки:

- Ограниченное количество бесплатных запросов в месяц.

Google Vision AI обладает всеми преимуществами рассмотренного ранее ABBYY Cloud OCR SDK сервиса. Тестовые прогоны картинок показали сопоставимые результаты. Также схожи принципы передачи изображе-

ния для последующего рассмотрения с помощью POST-запросов. Главным достоинством данного инструмента является возможность на протяжении года использовать сервис без каких-либо денежных вложений, но с ограниченным количеством запросов в месяц. Поэтому было принято решение выбрать данный сервис для поставленных задач.

2. Использование информации с чеков в корпоративной системе

2.1. Внедрение Google Vision AI

2.1.1. Подготовительный этап

Чтобы использовать API, необходимо воспроизвести следующие шаги в Google Cloud Developer Console:

- Создать проект в Google Cloud Console или использовать уже существующий;
- Включить в проекте Billing (оплату услуг). Если это первое использование Google Cloud Console, необходимо начать бесплатный пробный период использования. Будут запрошены данные карты, но денежные средства будут списывать только в случае превышения лимита запросов;
- Активировать Google Cloud Vision API;
- В меню консоли необходимо перейти в Диспетчер API, затем в раздел "Учетные данные" и создать ключ API, который будет использоваться для формирования endpoint url.

Данная блок-схема (рис. 7) наглядно демонстрирует процесс получения текста, содержащегося в изображении.

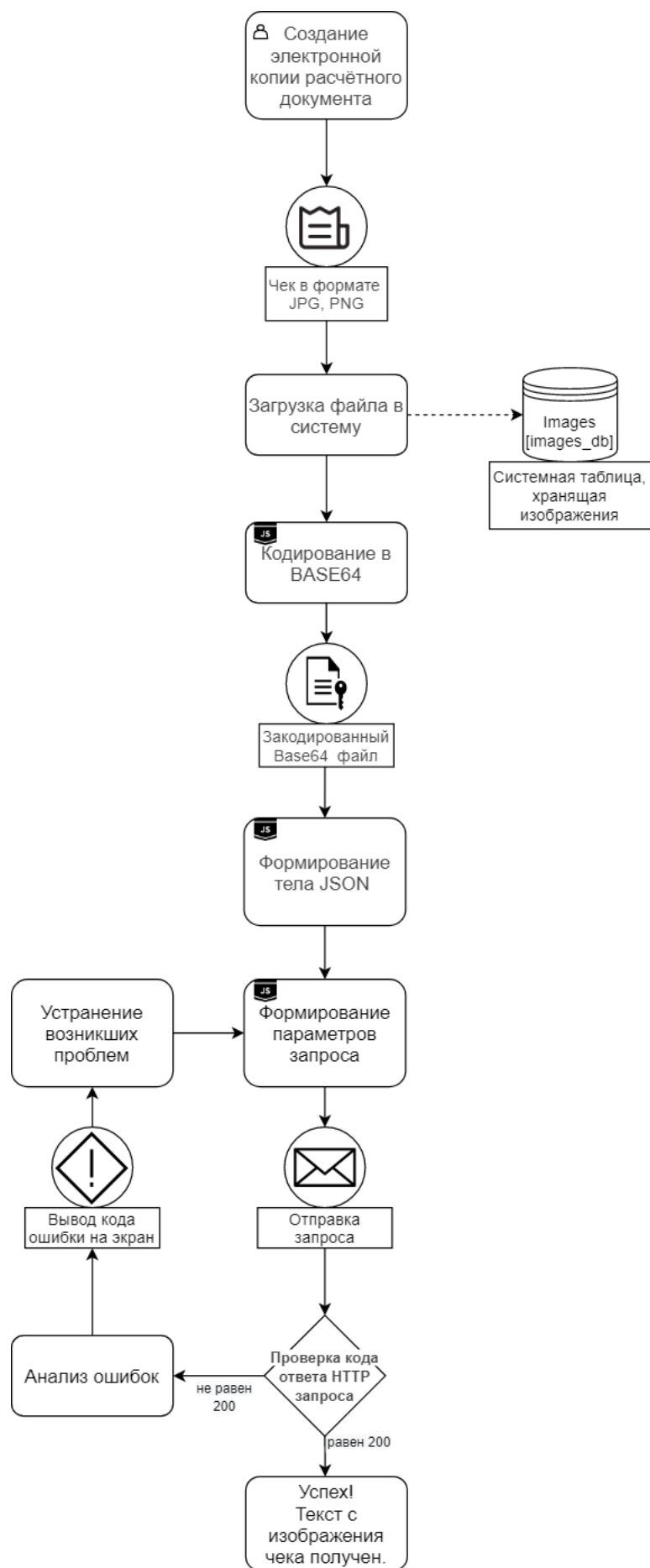


Рис. 7. Распознавание чеков с помощью Google Vision AI

Рассмотрим некоторые этапы более подробно.

2.1.2. Кодирование изображение в Base64 формат

Кодирование Base64 вводит избыточность, увеличивая размер. Основная цель - сделать так, чтобы данные могли быть представлены исключительно в виде текста. Есть ряд ситуаций, для которых шифрование необходимо. Например, в интеграции, для того чтобы защитить файлы от кражи (к чему могло бы привести открытие публичного доступа), было принято решение передавать закодированное изображение. Наиболее примечательным является то, что форматы сообщений для электронной почты раньше не принимали двоичный код. Кодировка Base64 преобразует файл так, что он содержит только буквенно-цифровые символы без знаков препинания (за исключением, возможно, одного или двух знаков равенства в конце). Всё вышеупомянутое позволяет поместить изображение как текст электронного письма и отправить так, чтобы его можно было восстановить на другой стороне. Это не единственный способ передать файл в закодированном виде. Очевидно, можно просто кодировать как текст, запоминая шестнадцатеричное значение каждого байта. Это означало бы, что каждый байт в исходных данных становится двумя символами в кодировке, то есть удваивается размер данных. Base64 кодирует каждые три байта ввода, используя четыре байта на выходе. Вместо того, чтобы удваивать размер, он только увеличивает его на треть. Он добавляет некоторую избыточность, но не такую существенную, и позволяет восстанавливать данные при передаче по каналу, который разрешает только текст.

С целью закодировать документ была создана функция (Листинг 1.), основанная на методе `base64Encode()` встроенной библиотеки `GlideStringUtil`. Функция получает разбитое на биты методом `getBytes()` изображение. Для того чтобы получить доступ к нему, необходимо прописать запрос к таблице `[sys_attachment]`, хранящей все существующие в системе документы (изображения в том числе). Первичный ключ, присвоенный изображению, передаётся в качестве параметра запроса.

```

1  base64Encode: function(attachID) {
2      var AttachGR = new GlideRecord('sys_attachment');
3      AttachGR.addQuery('table_sys_id', attachID);
4      AttachGR.query();
5      var base64 = '';
6
7      if (AttachGR.next()) {
8          var sysAttach = new GlideSysAttachment();
9          var binData = sysAttach.getBytes(AttachGR);
10         base64 += GlideStringUtil.base64Encode(binData);
11         return (base64);
12     }
13 },

```

Листинг 1. Base64 кодирование

2.1.3. Формирование JSON Request

Согласно официальной документации, сервис должен получить JSON объект с полями:

- requests (представляет собой массив, состоящий из объектов, описанных ниже);
- image (объект с полем content, где содержится Base64 закодированное изображение);
- features (массив с указанием нужного типа распознавания type, в данном случае TEXT_DETECTION).

Для формирования запроса (Листинг 2.) необходимо передать функции закодированное ранее изображение, затем создать строковую переменную, поместив в неё структуру будущего JSON объекта. Далее необходимо преобразовать переменную obj в JSON формат с помощью встроенного в систему метода encode. Функция возвращает готовое к отправке на сервер тело HTTP запроса.

```

1  requestJSON: function(b64Image) {
2      var obj = {
3          "requests": [{
4              "image": {
5                  "content": b64Image
6              },
7              "features": [{
8                  "type": "TEXT_DETECTION"
9              }]
10         }]
11     };
12
13     var parser = new JSON();
14     var request = parser.encode(obj);
15     return (request);

```

Листинг 2. Функция создания JSON запроса

2.1.4. Отправка Rest Message

REST API ServiceNow позволяет извлекать/создавать/обновлять/удалять данные на сервере веб-служб, которые поддерживают REST архитектуру. Для того чтобы отправить запрос (Листинг 3.) на Cloud Google сервер, необходимо обратиться к встроенным в платформу методам RESTMessageV2(). Сначала необходимо указать, что это POST HTTP запрос, добавить url конечной точки связи (должен включать в себя ранее созданный в Google Cloud Console ключ). Передаём данные для базовой HTTP авторизации. Остаётся только описать тело запроса и запрос готов к отправке.

```

1      var restMessage = new sn_ws.RESTMessageV2();
2      restMessage.setHttpMethod("POST");
3      restMessage.setEndpoint(gs.getProperty('google.auth.
4  endpoint'));
5      restMessage.setBasicAuth(username, password);
6      restMessage.setRequestBody(reqBody);
7      var response = restMessage.execute();

```

Листинг 3. Rest Message

2.1.5. Обработка JSON Response

Получив ответ от сервера, необходимо удостовериться, что код статуса ответа находится в пределах от 200 до 299. Именно этот числовой промежуток характеризует был ли успешно выполнен определённый HTTP запрос. В случае, когда все условия соблюдены, необходимо получить тело ответа в виде JSON объекта, воспользоваться методом `JSON.parse()`. Далее следует полю `[u_vision]` на форме присвоить текстовое значение, хранящееся по данному адресу указанному в 4 строке (Листинг 4). Если же случилась проблема в процессе отправки запроса или получения ответа, необходимо вывести сообщение об ошибке (строки 6-10).

```
1         if (response.getStatusCode() === 200) {
2             var responseBody = response.getBody();
3             var responseData = JSON.parse(responseBody);
4             current.u_vision = responseData.responses[0].
fullTextAnnotation.text;
5             current.update();
6         } else {
7             // process error response
8             var statusCode = response.getStatusCode();
9             var erorMessage = response.getErrorMessage();
10            var contentType = response.getHeader("Content-Type"
);
11            var body = response.getBody();
12        }
```

Листинг 4. JSON Request

2.1.6. Дополнительная обработка распознанного текста

Для дальнейшей работы с распознанным текстом необходимо сконструировать дополнительное модальное окно (рис. 8) в интерфейсе Service-Now. Основная задача по ключевым словам создать и заполнить поля будущей записи в системной таблице расходов компании.

OCR Text:

Json Text
1

Sum
2

Date
3

Provider
4

5 Cancel 6 Save 7 Parse

Рис. 8. Форма для обработки распознанного текста

Текстовое поле номер 1 предназначено для того, чтобы получить содержимое чека и исключить все орфографические ошибки, допущенные при распознавании. Для проверки орфографии используется встраиваемая JS библиотека Spell-checker.JS. Так же поле намеренно было оставлено редактируемым, для того чтобы у пользователя была возможность вручную подправить огрехи.

Реакция на нажатие кнопки Parse (номер 7 на изображении) заключается в том, чтобы поля 2-4 заполнились в соответствии с их названиями. С помощью встроенного метода `indexOf()` происходит поиск необходимой информации по ключевым словам, которая затем будет передана каждому из полей. Если пользователя устраивает итоговая заполненная форма, он может с помощью клавиши Save (6) сохранить результат в виде записи в таблице расходов, а по нажатию клавиши Cancel, все предыдущие действия будут отменены.

2.2. Внедрение Tesseract JS в корпоративную систему

В ситуациях, когда невозможно получить ответ от облачного сервера Google, необходимо предусмотреть существование запасного варианта, не

зависящего от стабильности работы сторонних API. Таким вариантом как раз является библиотека Tesseract.JS, хранящаяся локально в UI Script записях SN. Несмотря на то, что качество распознанного текста не сопоставимо с качеством, которое показывает Vision AI, после обработки и некоторых правок сделанных вручную в модальном окне, можно достичь желаемого результата. Во время работы с данной библиотекой были воспроизведены следующие шаги:

- Добавление библиотеки Tesseract.JS
Для того чтобы использовать возможности JS библиотеки внутри ServiceNow платформы, необходимо:
 - Выгрузить файл с исходным кодом библиотеки, перейдя по указанной ссылке [4];
 - Создать запись "OCR.JSdbx" в модуле UI Script;
 - В запись добавить исходный код библиотеки из скачанного ранее файла.
- Конфигурация и настройка системной таблицы Images [db_images] (добавление недостающих полей на форму);
- Создание приложения по распознаванию в форме Client Script (Листинг 5.);
- Сохранение итогового результата в форме.

Так как клиентский скрипт срабатывает по изменению значения в поле OCR language, задаём OnChange функцию, которая отслеживает эти изменения. С помощью метода getScripts подключаем библиотеку. В метод recognize передаём:

- Ссылку изображения, которая состоит из неизменяемой части "https://dev103823.service-now.com/" (адрес личного экземпляра платформы) и непосредственно названия изображения;
- Язык распознавания (передаётся в переменную newValue, когда пользователь выбирает значение в поле "OCR language");
- Объект настроек с логгером.


```

1 function onChange(control, oldValue, newValue, isLoading,
    isTemplate) {
2     if (isLoading || newValue === '') {
3         return;
4     }
5
6     ScriptLoader.getScripts('OCR.JSdbx', function() {
7         Tesseract.recognize('https://dev10382.service-now.com/'
+ g_form.getValue('name'), newValue, {
8             logger: function logger(m) {
9                 return g_form.setValue('u_ocr_text', 'Progress:'
+ ((parseFloat(m['progress']) * 100).toFixed(2)) + '%');
10            }
11        }).then(function(_ref) {
12            var text = _ref.data.text;
13            g_form.setValue('u_ocr_text', text);
14        });
15    });
16 }

```

Листинг 5. Распознавание чеков с помощью Tesseract.JS

3. Анализ результатов

Для оценки эффективности алгоритмов было принято решение провести тестирование. Было проанализировано порядка 40 чеков. Рассмотрим подробнее упомянутое ранее изображения (рис. 3).

На форме анализа текста (рис. 9) отчётливо видно, что текст, полученный в результате распознавания с помощью Google Vision был успешно обработан. Все поля на форме заполнены корректными значениями. Но нажатие клавиши Save была создана запись в платформе ServiceNow, содержащая в себе информацию о дате, итоговой сумме и классе расхода.

OCR Text:

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "ЯНДЕКС. ТАКСИ"
ИНН: 7704340310 taxi.yandex.ru
Кассовый чек. Приход
N 653 N АВТ 101922631 Смена N 269
25.11.19 02:05 N
Наим. пр. Цена за ед. Колич. НДС

Sum
1069.00

Date
25.11.19

Provider
taxi.yandex.ru

Cancel Save Parse

Рис. 9. Результат работы Google Vision AI

Обратившись к той же форме (рис. 10), можно заметить, что результат, полученный с использованием библиотеки Tesseract, оказался неполным. Даже после орфографической проверки, остались неточно распознанные элементы, содержащие в себе буквы английского алфавита и другие символы. Именно поэтому поле Provider осталось пустым и требует действий со стороны пользователя.

OCR Text:

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "ЯНДЕКС.ТАКСИ"
ИНН: 7704340310
+axl. yапбex. ги
Кассовый чек. Приход
M 653 M АВТ 101922631
Смена № 269 25.11.19 02:05

Sum
1069.00

Date
25.11.19

Provider
~

Cancel Save Parse

Рис. 10. Результат работы Tesseract.JS

Из результатов тестирования можно сделать следующий вывод:

- Сервис Google Vision AI в большинстве случаев показал лучший результат в сравнении с Tesseract.JS, в связи с наличием слов написанных на разных языках. Чеки содержащие только один язык были одинаково хорошо распознаны;
- В случае неполадок на сервере Google или проблем со связью по HTTP протоколу, всегда можно будет воспользоваться встраиваемой в платформу библиотекой Tesseract.JS, но необходимо будет вносить свои коррективы, чтобы качество текста оставалось на высоком уровне;
- С помощью внедрения OCR инструментов удалось успешно автоматизировать часть процессов документооборота, отвечающих за получение данных о совершённых сотрудниками компании покупках товаров и услуг.

Заключение

В рамках данной выпускной квалификационной работе были успешно выполнены шаги по изучению базового функционала и особенностей корпоративной системы ServiceNow и методов интеграции сторонних веб-сервисов в неё. Также дополнительно был проведён сбор и анализ тестовых данных, изучены принципы работы REST архитектуры.

Результатом данного исследования являются внедрённые в платформу инструменты для распознавания текста на изображениях кассовых чеков. Использование нескольких ресурсов (Google Vision AI, Tesseract.JS.) стало гарантом стабильной работы приложения: в случае неполадок на сервере Google Vision AI, всегда есть возможность переключиться на Tesseract.JS.

В ходе данной работы получены следующие результаты:

- Определён набор инструментов, успешно справляющихся с поставленной задачей;
- Создана тестовая база, содержащая в себе кассовые чеки разных типов;
- Настроена интеграция с библиотекой Tesseract.JS и с Google Vision AI;
- Создан интерфейс, в котором происходит пост-обработка текста;
- Протестирован функционал OCR инструментов;
- Предусмотрена бесперебойная работа по распознаванию.

Литература

1. ServiceNow официальный сайт – URL: <https://www.servicenow.com/> (дата обращения 17.05.2020)
2. Google Vision AI API – URL: <https://cloud.google.com/vision> (дата обращения 22.05.2020)
3. Дуглас Крокфорд "Как устроен JavaScript.-- Изд-во Питер, 2019 г.
4. Tesseract.JS библиотека – URL: <https://unpkg.com/tesseract.JS@v2.0.0-alpha.13/dist/tesseract.min.JS>. (дата обращения 23.05.2020)
5. Spell-checker.JS библиотека – URL: <https://www.npmJS.com/package/spell-checker-JS> (дата обращения 27.05.2020)
6. Business Rules документация SN – URL: <https://www.servicenowelite.com/blog/2016/6/11/learn-business-rules> (дата обращения 30.05.2020)
7. Script Includes официальная документация SN – URL: https://docs.servicenow.com/bundle/orlando-application-development/page/script/server-scripting/concept/c_ScriptIncludes.html (дата обращения 03.06.2020)
8. REST API документация SN – URL: <https://helpcentre.buttonwood.net/hc/en-us/articles/360000338036-ServiceNowExample-How-to-Create-REST-API-Requests> (дата обращения 19.06.2020)
9. Client Scripts официальная документация SN – URL: https://developer.servicenow.com/dev.do#!/guide/orlando/nowplatform/tpb-guide/client_scripting_technical_best_practices (дата обращения 03.06.2020)
10. Service Portal официальная документация SN – URL: https://docs.servicenow.com/bundle/orlando-servicenow-platform/page/build/service-portal/concept/c_ServicePortal.html (дата обращения 17.06.2020)
11. UI Scripts community портал SN – URL: https://community.servicenow.com/community?id=community_question&sys_id=21bb0877db80a744fff8a345ca96192b (дата обращения 17.06.2020)

08.06.2020)

12. UI Pages официальная документация SN – URL:

https://docs.servicenow.com/bundle/orlando-application-development/page/script/server-scripting/reference/r_UIPages.html (дата обращения 12.06.2020)

13. Performance Analytics официальная документация SN – URL:

<https://www.servicenow.com/content/dam/servicenow-assets/public/en-us/doc-type/resource-center/data-sheet/ds-performance-analytics.pdf> (дата обращения 20.06.2020)

14. System Properties официальная документация SN – URL:

https://docs.servicenow.com/bundle/orlando-platform-administration/page/administer/reference-pages/task/t_AddAPropertyUsingSysPropsList.html (дата обращения 05.06.2020)

15. UI Action официальная документация SN – URL:

https://hi.service-now.com/kb_view.do?syparm_article=KB0547282 (дата обращения 30.06.2020)