ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ

им. А.Н. ТИХОНОВА

Муленко Евгения Андреевна, группа БИВ195

Граур Полина Алексеевна, группа БИВ195

**РУКОВОДСТВО РАЗРАБОТЧИКА**

**по выпускной квалификационной работе**

Разработка программы автоматизированного извлечения данных из разноформатных документов по составу электронных устройств

Москва, 2024

## Аннотация

Была разработана программы для автоматизированного извлечения данных из документов различных форматов о структуре электронных устройств. Была разработана база данных, содержащая сведения о надежности элементов по их группам и типам, основанная на информации из справочников. Разработан код приложения, который ищет соответствие между переданной на вход серией элемента и информацией из базы данных, а при ее отсутствии производит запросы в интернет для получения описания элемента.

## Архитектура проекта

Ключевыми элементами архитектуры в данной выпускной работе будут являться веб-интерфейс для пользователя, через который будет происходить взаимодействие с программой, модули бизнес-логики приложения и база данных. К модулям бизнес-логики относятся разделы программного решения, отвечающие за парсинг файлов, загруженных пользователем, поиск информации о группах элементов и их надежности, классификацию групп ЭРИ с помощью нейронной сети и взаимодействие с внешними ресурсами для получения информации об электрорадиоэлементах.

Взаимодействие между модулями в предложенном архитектурном решении будет происходить через программные интерфейсы, что предоставит возможности для масштабирования программы.

На рисунке 1 продемонстрирована схема архитектуры приложения. Линиями синего цвета отображены потенциальные пути масштабирования программы по верификации электрорадиоэлементов согласно структуре технологических групп ЭКБ.

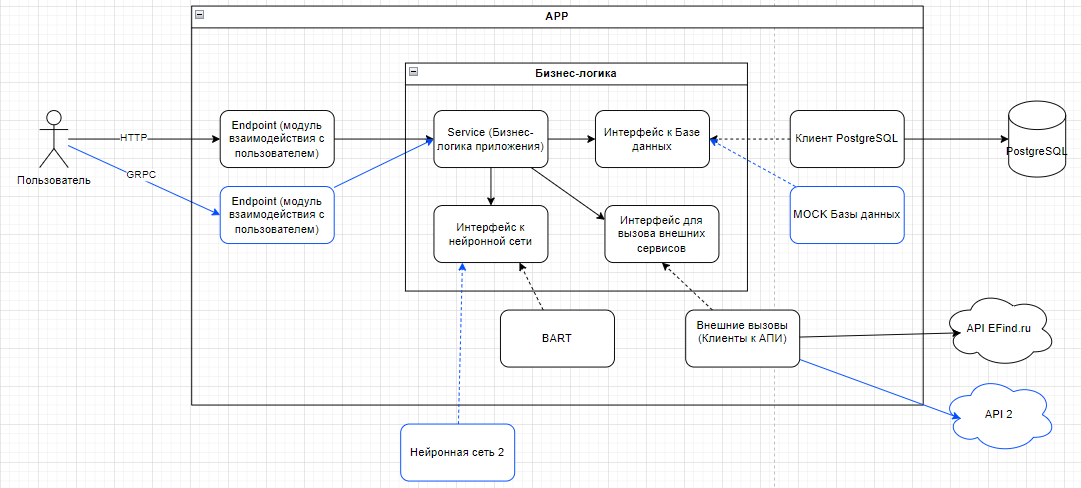


Рисунок 1 – Схема архитектуры программы

Так, например, для определения группы элемента по его описанию система использует нейронную сеть, которая была выбрана по критериям актуальности на момент реализации выпускной квалификационной работы. Развитие нейросетей идет с каждым годом с нарастающим темпом, в реализованной программной архитектуре есть возможность добавления дополнительной нейросети для реализации функций приложения, как созданной другими разработчиками, так и обученной отдельно для решения специфических для приложения задач.

Также в данном архитектурном решении предусмотрено расширение используемых источников информации, предоставляющих данные об элементах КМОП-перечня, и схем взаимодействия пользователя с системой.

На рисунке 2 представлена блок-схема, отображающая текущую логику работы системы.

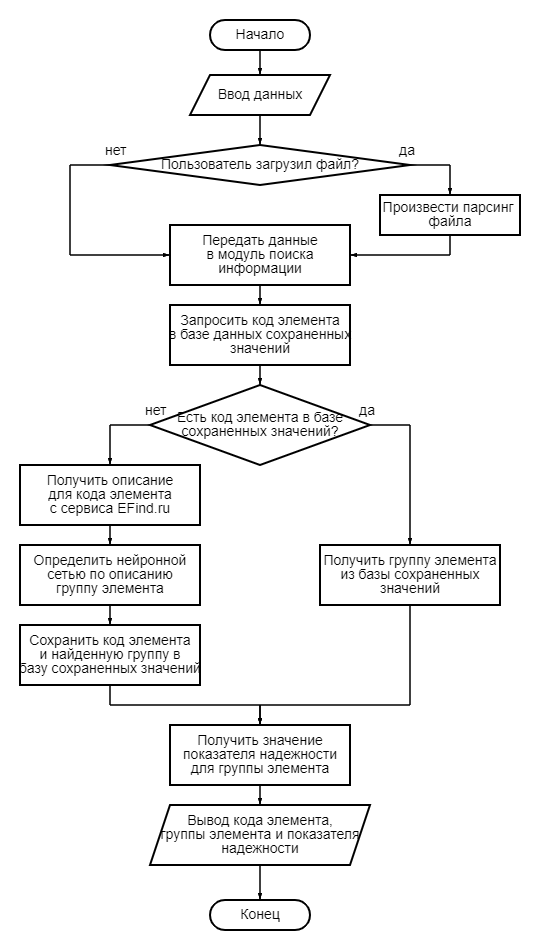


Рисунок 2 – Блок-схема логики программы приложения

## Инструменты и технологии

В данном научном проекте использованы современные инструменты и технологии в области машинного обучения и веб-разработки:

1. Python 3.10 - основной язык программирования для реализации логики сервера и API, а также обработки файлов и данных. Python выбран за его обширную экосистему библиотек для работы с данными и ML.
2. Flask - микрофреймворк для создания веб-приложений на Python. Используется для реализации REST API и роутинга.
3. SQLAlchemy - популярная ORM библиотека для работы с реляционными базами данных из Python кода.
4. transformers - библиотека для работы с SOTA моделями машинного обучения, в частности с BART моделью для zero-shot классификации.
5. pandas - библиотека для работы с табличными данными и ETL. Используется для чтения и обработки входных excel/pdf файлов.
6. NumPy - фундаментальная библиотека для научных вычислений на Python. Применяется для работы с массивами.
7. tabula-py - библиотека для извлечения таблиц из PDF файлов.
8. openpyxl - библиотека для чтения/записи Excel файлов.

Такой технологический стек позволил эффективно решить поставленные в исследовании задачи – классификация электронных компонентов по группам на основе их описания и определение базового показателя надежности, используя методы NLP, а также реализовать удобный веб-интерфейс для работы.

## Ключевые части программы

Ядром программы является веб-приложение Flask (app.py), реализующее API с двумя ключевыми методами. Получение значения надёжности по названию компонента

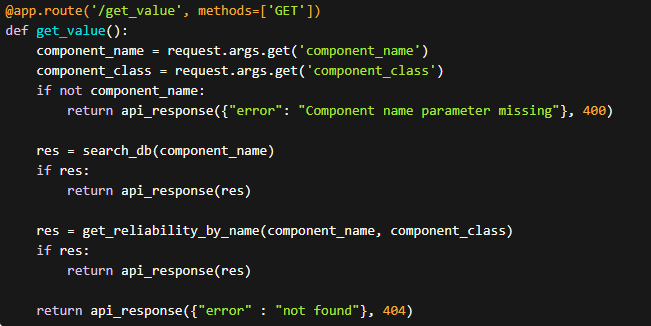


Рисунок 3 – Листинг взаимодействия с API

Метод пытается сначала найти компонент в базе данных по точному совпадению. Если не найден, то происходит поиск через модель ML, которой на вход подаётся текстовое описание компонента, полученное через API.

Пакетное получение значений надёжности из файла

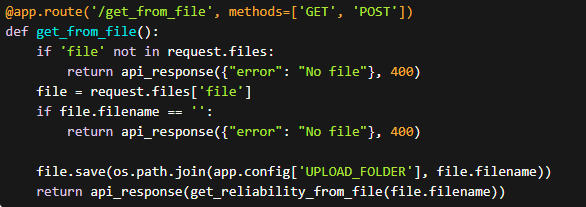


Рисунок 4 – Листинг получения надежности

Метод принимает файл excel или pdf, извлекает из него данные с помощью pandas и табличных парсеров (excel\_parser.py, pdf\_parser.py). Затем для каждой строки файла происходит поиск значения надёжности аналогично get\_value - сначала точное совпадение в БД, потом через ML модель.

Ещё одной важной частью является функция get\_reliability\_by\_name (app.py:171), которая реализует поиск через ML модель:

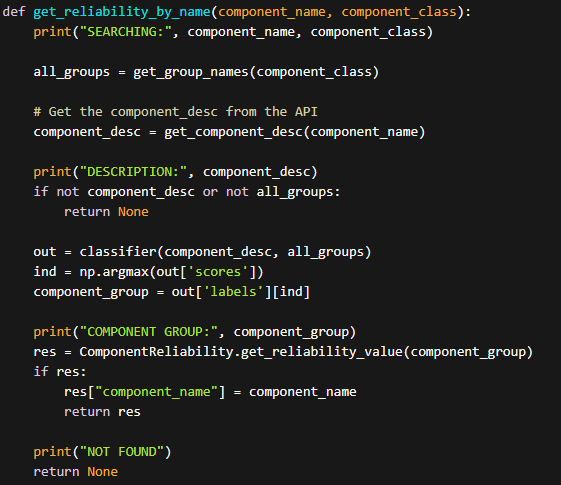


Рисунок 5 – Листинг определения группы по описанию

Сначала получаем текстовое описание компонента через API (функция get\_component\_desc). Затем передаём описание в модель zero-shot классификации BART. Модель возвращает оценки сходства описания с каждым из возможных классов надёжности. Выбираем класс с максимальной оценкой и по нему находим значение надёжности в БД.

Для хранения и быстрого доступа к данным используется реляционная БД SQLite. Описание моделей данных находится в app.py:38-107. Основные сущности:

1. DimGroup - справочник групп надёжности
2. ComponentReliability - значения надёжности для каждой группы
3. DimComponent - справочник компонентов для точного поиска
4. ClassComponent - справочник классов компонентов

В качестве небольшой оптимизации реализовано внутреннее кэширование описаний компонентов, полученных через API (cache/cache.py). Это позволяет разгрузить внешние сервисы.

Для удобства пользователей подготовлен простой веб-интерфейс (static/index.html).

Таким образом, созданная программа представляет собой законченный программный продукт для решения научной задачи классификации и оценки надежности электронных компонентов, готовый к практическому применению.

Указанные инструменты, подходы и ключевые части программы рекомендуется учитывать при дальнейшей разработке и сопровождении данного проекта.