Лицей Академии Яндекса

**Пояснительная записка к проекту «Mhelper»**

**Выполнил**:

Рекалов Артём

**Руководитель**:

Логинова Юлия Вячеславовна

Г.Ижевск

9.11.2021

Оглавление

[Ведение 2](#_Toc87967875)

[Описание идеи 2](#_Toc87967876)

[Актуальность 2](#_Toc87967877)

[Область применения 2](#_Toc87967878)

[Функции проекта 2](#_Toc87967879)

[Реализованные функции 2](#_Toc87967880)

[Путь пользователя (User path) 3](#_Toc87967881)

[Внутренняя реализация проекта 3](#_Toc87967882)

[Классы графического интерфейса, окон и основной логики 3](#_Toc87967883)

[MainWindow 3](#_Toc87967884)

[AddMailDialog 3](#_Toc87967885)

[SeeMailWindow 3](#_Toc87967886)

[SeeEmailText 4](#_Toc87967887)

[Специальные классы и функции 4](#_Toc87967888)

[BdConnect 4](#_Toc87967889)

[NerModel 4](#_Toc87967890)

[text\_extractor 4](#_Toc87967891)

[Файлы проекта 4](#_Toc87967892)

[Вывод 5](#_Toc87967893)

[Приложение 6](#_Toc87967894)

[1 6](#_Toc87967895)

[2 7](#_Toc87967896)

[3 7](#_Toc87967897)

[4 8](#_Toc87967898)

[5 8](#_Toc87967899)

# Ведение

## Описание идеи

Приложение, созданное для сортировки, извлечение именованных сущностей из фотографии или текста электронного письма с последующим занесением в локальную базу данных. По базе данных возможен поиск письма по критериям.

## Актуальность

В настоящее время весь оборот официальных сообщений происходит через электронную почту, именно поэтому проект максимально актуален.

## Область применения

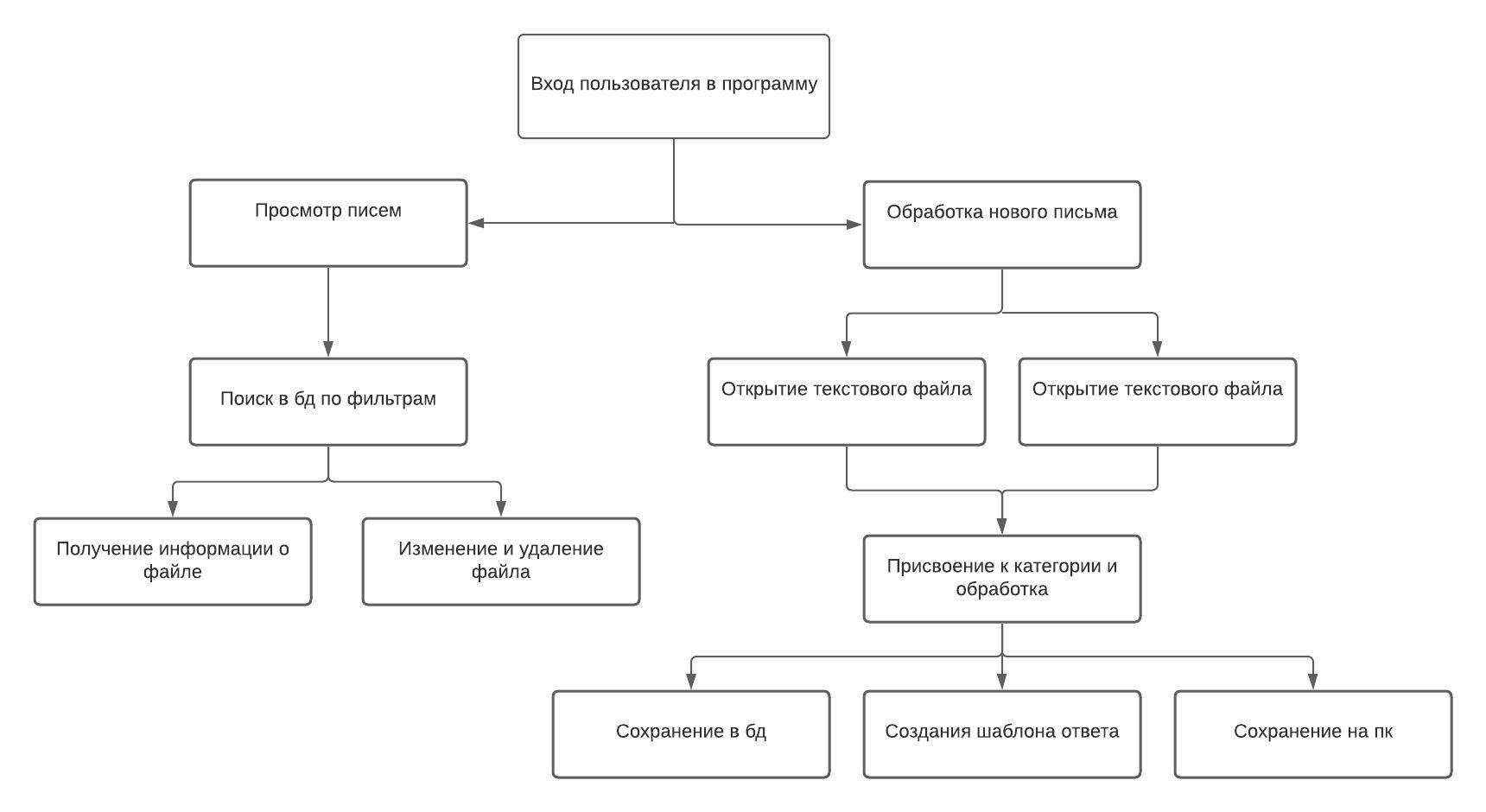
Изначально проект представляет отдельную программу, однако потом может быть переведен на плагин, внедряемый в систему коммуницирования предприятий.

# Функции проекта

## Реализованные функции

1. Преобразование фотографии или скриншота письма в текстовый формат.
2. Обработка текста письма: классификация письма к определенной категории, извлечение сущностей (отправитель, получатель, тема)
3. Сохранение писем в локальную базу данных.
4. Поиск писем в базе данных по фильтрам.
5. Просмотр писем.

## Путь пользователя (User path)

****

# Внутренняя реализация проекта

## Классы графического интерфейса, окон и основной логики

## MainWindow

Класс унаследован от QmainWindow и представляет из себя главное пользовательское окно с возможностями создания и обработки нового или уже имеющегося письма.

В приложении 1 можно найти дизайн окна, создаваемого классом

## AddMailDialog

Класс унаследован от Qdialog и является диалоговым окном для выбора обработки письма.

В приложении 2можно найти дизайн диалогового окна, создаваемого классом

## SeeMailWindow

Также унаследован от QMainWindow. Класс открывает окно, представляющие таблицу просмотра существующих писем, их сортировки, редактирование и удаления.

В приложении 3 можно найти дизайн окна, создаваемого классом

## SeeEmailText

Также унаследован от QMainWindow. Класс открывает окно, поле для просмотра текста письма.

В приложении 4 можно найти дизайн окна, создаваемого классом

## Специальные классы и функции

## BdConnect

Класс для работы с базой данных Sqlite3, добавление в нее новых записей и чтения имеющихся.

## NerModel

Функция, работающий с NER моделью, извлекающий именованные сущности из текста.

## text\_extractor

Функции, для считывания текста с картинки или txt файла в string питона.

## Файлы проекта

1. main.py – Основной файл состоящий из всех графических классов и управляющий общей логикой программы. Используемые библиотеки: PyQt5, pyqt5-tools, sys
2. NerModel.py – Файл содержащий одноименный класс и выполняющий ner функции. Используемые библиотеки: Sk-learn, Spacy.
3. Classificator.py - Файл содержащий одноименный класс и выполняющий функции классификации текста. Используемые библиотеки: Sk-Learn, NLTK.
4. BdConnect.py - Файл содержащий одноименный класс и выполняющий функции работы с базой данных. Используемые библиотеки: sqlite3.
5. Bd.bd – база данных, состоящая из 2 таблиц. Подробнее можно увидеть в приложении 5.
6. Exceptions.py – Файл, хранящий в себе классы исключений.
7. Директория designs с файлами dialog.ui, main.ui, see\_db.ui, see\_email.ui – файлы макетов окон.
8. Директория designs\_py с файлами dialog\_ui.py, main\_ui.py, see\_db\_ui.py, see\_email\_ui.py – файлы макетов окон в формате .py.
9. Requirements.py – Файл с использованными библиотеками для их удобной установки.
10. TZ.docx – Файл тз проекта.
11. README.md – Файл с технической документацией проекта для GitHub.
12. explanatory note.docx – Файл технической записки.
13. Classifier\_generator.py – генератор модели classifier.pql
14. Classifier.pql – модель машинного обучения, классифицирующая текст
15. Test\_extractor.py – файл с функциями для считывания текста с картинки и из текстового файла.
16. Vectorizer.pql – модель обученного векторайзера.

# Вывод

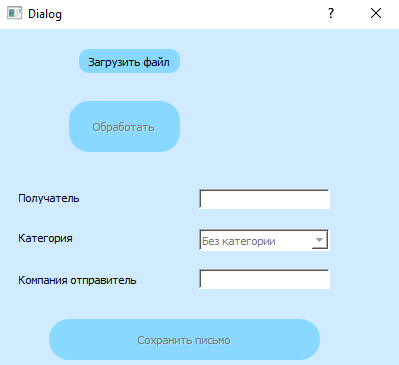
Проект актуален и монетизируем. На данной стадии mvp продукта может решать круг поставленных на него задач и выполняет все заявленные функции. Также у проекта множество перспектив и путей развития, от банального улучшения дизайна , до перехода на сверточную нейросеть и внедрения в крупные предприятия.

# Приложение

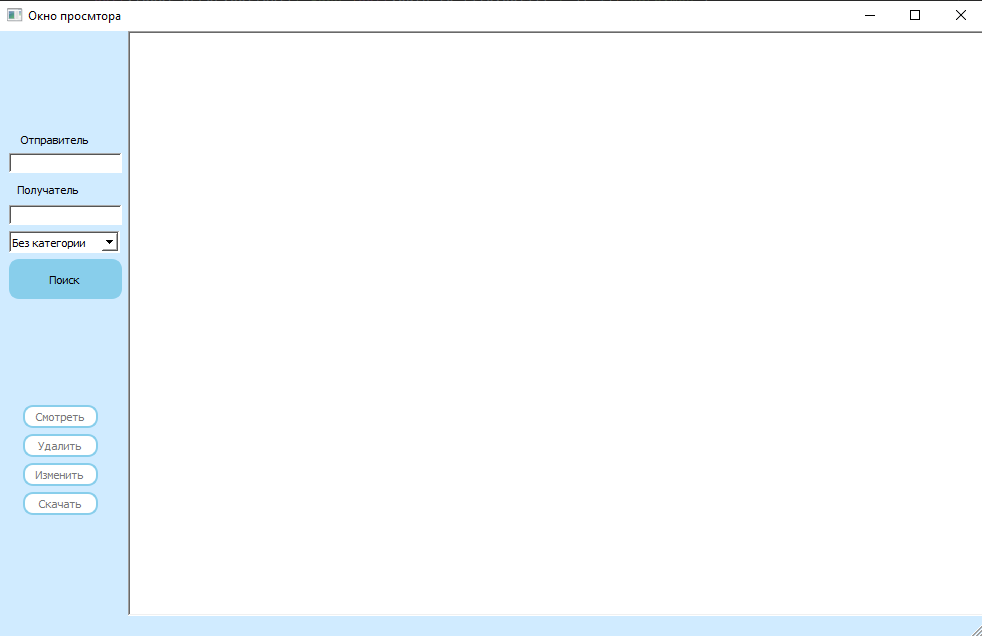
## 1



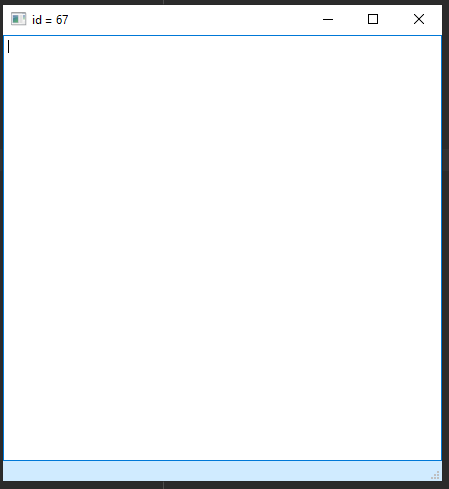
## 2



## 3



## 4



## 5

