

бюджетное профессиональное образовательное учреждение Вологодской области
«Череповецкий лесомеханический техникум им. В.П. Чкалова»

Специальность **09.02.07** «Информационные системы и программирование»

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ
ПП по ПМ.03 РЕВЬЮИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

Выполнил студент 3 курса группы ИС-32

подпись _____

Место практики _____

наименование юридического лица, ФИО ИП

Период прохождения:

с «__» _____ 2024 г.

по «__» _____ 2024 г

Руководитель практики от предприятия,

должность _____

подпись _____

МП Руководитель практики от

техникума: Материкова А.А.

Оценка: _____

«__» _____ 2024 года

г. Череповец 2024

Содержание

Введение	3
Актуальность работы.....	3
Цель производственной практики (по профилю специальности)	3
Задачи.....	3
Основная часть.....	4
1.Общая характеристика предприятия (организации) ООО “Малленом Системс” ..	4
1.1 Деятельность	4
1.2 Направления.....	4
1.3 Проекты	5
1.4 Структурная организация компании.....	5
2. Ревьюирование программных продуктов.....	6
2.1 Ревьюирование программного кода в соответствии с технической документацией.	6
2.2 Измерение характеристик компонентов программного продукта.	7
2.3 Исследование созданного программного кода с использованием специализированных программных средств.	8
2.4 Сравнительный анализ программных продуктов и средств разработки.	9
3. Выполняемые задания	11
Задания первой недели:	11
Задания второй недели:	14
Заключение	17
Источники информации.....	18
Приложения	19

Введение

Место прохождения практики: ООО “Малленом Системс”

Сроки прохождения производственной практики: с 24.11.2024 по 07.12.2024

Актуальность работы

Актуальность данной работы обусловлена необходимостью повышения эффективности разработки программного обеспечения и минимизации рисков, связанных с ошибками в коде. В условиях быстрого технологического прогресса и жесткой конкуренции на рынке IT, компании стремятся оптимизировать процессы разработки, что делает ревьюирование неотъемлемой частью жизненного цикла программного обеспечения.

Цель производственной практики (по профилю специальности)

Производственная практика (по профилю специальности) направлена на формирование у обучающегося общих и профессиональных компетенций, приобретение практического опыта и реализуется в рамках модуля ПМ.03 специальности 09.02.07 Информационные системы и программирование

Задачи

1. Иметь практический опыт в изменении характеристик программного проекта, использовании основных методологий процессов разработки программного обеспечения, оптимизации программного кода с использованием специализированных программных средств
2. Уметь работать с проектной документацией, разработанной с использованием графических языков спецификаций
3. Знать задачи планирования и контроля развития проекта принципы построения системы деятельности программного проекта

Основная часть

1. Общая характеристика предприятия (организации) ООО “Малленом Системс”

1.1 Деятельность

Основной вид деятельности организации: Разработка компьютерного программного обеспечения (код по ОКВЭД 62.01).

Дополнительно организация заявила следующие виды деятельности:

Производство компьютеров и периферийного оборудования.

Монтаж промышленных машин и оборудования.

Торговля оптовая компьютерами, периферийными устройствами к компьютерам и программным обеспечением.

Торговля розничная аудио- и видеотехникой в специализированных магазинах.

Деятельность консультативная и работы в области компьютерных технологий.

1.2 Направления

Визуальный контроль продукции на производстве.

Контроль качества, идентификация и прослеживание, обязательная маркировка «Честный Знак».

Видеоконтроль и учёт автомобильного и железнодорожного транспорта.

Распознавание номеров автомобилей, анализ трафика, автоматизация взвешивания.

Ситуационная видеоаналитика.

Контроль персонала, промышленного оборудования и опасных событий.

Видеоаналитика с применением БПЛА (дронов).

1.3 Проекты

ВИСКОНТ – интеллектуальные системы машинного зрения для контроля качества и прослеживания продукции

EYESCONT – система ситуационной видеоаналитики для контроля людей и опасных событий на видео

АВТОМАРШАЛ – система контроля доступа и учета автотранспорта на основе технологии распознавания автономеров

АВТОМАРШАЛ.ВЕСОВАЯ - аппаратно-программный комплекс, решающий полный перечень задач автоматизации автомобильных весовых

AVEDEX – программное обеспечение для подсчета и анализа автомобильного трафика по видеоизображению

АРДИС – семейство решений по идентификации вагонов, управлению отгрузкой продукции ж/д транспортом, коммерческому осмотру вагонов, контролю передвижения и местонахождения вагонов, построенное на платформе системы распознавания номеров вагонов.

1.4 Структурная организация компании

Отдел разработки: занимается проектированием и реализацией программных решений.

Отдел продаж и маркетинга: отвечает за привлечение клиентов и продвижение услуг компании.

Техническая поддержка: обеспечивает поддержку клиентов на всех этапах сотрудничества.

Финансовый отдел: управляет финансовыми потоками и отчетностью компании.

2. Ревьюирование программных продуктов

Ревьюирование программных продуктов — это критически важный процесс в разработке программного обеспечения, который позволяет выявить ошибки, улучшить качество кода и обеспечить соответствие техническим требованиям. Ревью может включать в себя различные аспекты, такие как анализ кода, измерение характеристик компонентов, использование специализированных инструментов и сравнительный анализ программных продуктов.

2.1 Ревьюирование программного кода в соответствии с технической документацией.

Ревьюирование кода — это процесс проверки исходного кода программы с целью обеспечения его качества и соответствия установленным стандартам и требованиям. Важным аспектом этого процесса является соответствие кода технической документации, которая описывает функциональные и нефункциональные требования к программному продукту.

Цели ревьюирования кода:

Обеспечение качества

Соблюдение стандартов

Улучшение командной работы

Процесс ревьюирования:

1. Подготовка к ревью
2. Анализ кода
3. Обратная связь
4. Документирование результатов

2.2 Измерение характеристик компонентов программного продукта.

Измерение характеристик компонентов программного продукта является важной частью процесса ревьюирования. Это позволяет оценить качество программного обеспечения с точки зрения его производительности, надежности, безопасности и других ключевых параметров.

Ключевые характеристики для измерения:

Производительность: Время отклика системы, скорость обработки данных, использование ресурсов (память, процессор).

Надежность

Масштабируемость: Способность системы справляться с увеличением нагрузки без ухудшения производительности.

Поддерживаемость: Легкость внесения изменений в код, наличие документации и тестов.

Безопасность

Методы измерения:

1. Статический анализ кода: Использование инструментов анализа для оценки качества кода, выявления потенциальных проблем и соблюдения стандартов кодирования.
2. Динамическое тестирование: Проведение нагрузочного тестирования для оценки производительности приложения под различными условиями.
3. Метрики кода: Измерение метрик, таких как количество строк кода, сложность алгоритмов, количество классов/функций и т.д., для оценки поддерживаемости и качества кода.

2.3 Исследование созданного программного кода с использованием специализированных программных средств.

Исследование созданного программного кода с использованием специализированных программных средств позволяет более глубоко проанализировать качество и безопасность программного обеспечения. Это может включать использование различных инструментов для статического анализа, профилирования, тестирования и анализа безопасности.

Инструменты для исследования кода:

Статические анализаторы кода: Программы, которые анализируют исходный код без его выполнения.

Инструменты для профилирования: Позволяют выявить узкие места в производительности программы.

Системы контроля версий: позволяют отслеживать изменения в коде, проводить ревью через pull-запросы и управлять версиями проекта.

Инструменты для тестирования

Инструменты для анализа безопасности: помогают выявить уязвимости в приложении до его развертывания в продакшн.

Процесс исследования:

1. Выбор инструментов: Определение необходимых инструментов в зависимости от целей анализа (например, статический анализ для выявления ошибок или профилирование для оценки производительности).
2. Проведение анализа: Использование выбранных инструментов для получения данных о качестве кода и его характеристиках.
3. Интерпретация результатов: Анализ полученных данных, выявление проблемных областей и формулирование рекомендаций по их устранению.
4. Документирование: Составление отчета о проведенном исследовании с выводами и предложениями по улучшению качества кода.

2.4 Сравнительный анализ программных продуктов и средств разработки.

Сравнительный анализ программных продуктов и средств разработки позволяет определить сильные и слабые стороны различных решений на рынке. Это может включать оценку функциональности, производительности, удобства использования, стоимости и других факторов.

Цели сравнительного анализа:

Выявление лучших практик и подходов в разработке программного обеспечения.

Определение наиболее подходящих инструментов и технологий для конкретных задач.

Оценка конкурентоспособности собственного продукта на фоне аналогичных решений.

Критерии для сравнения:

1. Функциональность: Наличие необходимых функций и возможностей для пользователей.
2. Производительность: Скорость работы приложения при различных нагрузках.
3. Удобство использования: Интуитивно понятный интерфейс и легкость в обучении пользователей.
4. Поддержка и документация: Наличие качественной документации, активного сообщества или службы поддержки.
5. Стоимость: Общая стоимость владения продуктом, включая лицензии, обучение и поддержку.

Методы проведения сравнительного анализа:

1. Тестирование продуктов: Проведение практических тестов различных решений для оценки их функциональности и производительности.
2. Сбор отзывов пользователей: Анализ мнений пользователей о различных продуктах через форумы, обзоры и социальные сети.
3. Изучение документации: Сравнение качества документации и доступности ресурсов для обучения пользователей.
4. Анализ рынка: Оценка позиций различных решений на рынке с точки зрения доли рынка, популярности и репутации.

3. Выполняемые задания

Задания первой недели:

Задание 1

1. В рамках первого задания я ознакомилась с организацией ООО "Малленом Системс". Также изучила сферу деятельности компании, её направления работы, текущие проекты и структуру. В результате я составила отчет, в котором изложила всю собранную информацию.

Задание 2

1. Для второго задания я создала репозиторий на GIT. В репозитории я разместила выполненные задания, отчеты и другие документы, относящиеся к текущей практике. (Рис.1)

Структура репозитория включает следующие разделы:

Отчет (Report)

Задания (Src)

Документы (Docs)

Также я добавила в репозиторий отчет и документы по практике, включая описание первого раздела.

Задание 3

В рамках выполнения задания я разработала три модуля на языке программирования Python, используя графический интерфейс PyQt.

1. Модуль обработки и работы с изображениями (Рис.2)

В этом модуле я реализовала следующие функциональные возможности:

Изменение размера изображения: Я создала функцию, которая позволяет пользователю вводить новый размер (ширину и высоту) для загружаемого изображения. После изменения размера изображение сохраняется в указанном формате.

Поворот изображения: Также я добавила возможность поворота изображения на заданный угол, который пользователь может указать (например, 90, 180 или 270 градусов). Изображение после поворота также сохраняется.

2. Модуль взаимодействия с пользователем и формирование и хранения данных (Рис.3)

В этом модуле я разработала графический интерфейс для пользователя, который включает в себя:

Отображение графического интерфейса: Я создала окно приложения с полями ввода для указания пути к изображению и новых размеров (ширины и высоты).

Передача данных модулю обработки изображений: Пользователь может задать путь к изображению и новые размеры. После нажатия кнопки выполняется операция изменения размера или поворота, и результат отображается в интерфейсе.

3. Модуль пользовательского интерфейса, который разработан с использованием библиотеки PyQt. Этот модуль отвечает за взаимодействие с пользователем и предоставляет графический интерфейс для выполнения операций над изображениями. (Рис.4)

В этом модуле я разработала пользовательский интерфейс для пользователя, который включает в себя:

1.Окно приложения: Главное окно приложения создается с помощью класса QWidget. В нем размещаются все элементы интерфейса.

2. Поля ввода:

Поле для ввода пути к изображению: Позволяет пользователю указать путь к файлу изображения, который он хочет обработать.

Поля для ввода новых размеров: Два поля, одно для ширины и другое для высоты, позволяют пользователю задать новые размеры изображения.

3. Кнопки:

Кнопка "Загрузить изображение": При нажатии открывается диалоговое окно для выбора изображения. Загруженное изображение сохраняется в переменной `self.image`, и отображается сообщение о его загрузке.

Кнопка "Изменить размер изображения": При нажатии на эту кнопку вызывается функция изменения размера изображения.

Кнопка "Повернуть изображение": Поворачивает загруженное изображение на выбранное количество градусов. Если изображение не загружено, выводится соответствующее сообщение.

Кнопка "Сохранить изображение": Открывает диалоговое окно для выбора места и имени файла для сохранения. Сохраняет текущее состояние изображения в выбранный файл.

4. Метка для отображения результата:

Метка (QLabel) используется для отображения сообщений о результате выполнения операции (например, успешного изменения размера).

В итоге получилось приложение, представленное на Рис.5

Задания второй недели:

Задание 1

Я провела обратное проектирование, используя графический язык UML через платформу diagrams.net. В процессе работы я создала и описала следующие диаграммы:

Диаграмма компонентов: показывает основные компоненты системы и их взаимосвязи. Компоненты могут представлять модули, библиотеки или сервисы, которые взаимодействуют друг с другом. (Рис.6)

Диаграмма сценариев использования: визуализирует пользователей и их взаимодействие с системой через сценарии использования. Каждый сценарий описывает конкретную функцию или задачу, которую пользователь может выполнить. (Рис.7)

Диаграмма последовательностей: иллюстрирует порядок взаимодействия объектов в рамках конкретного сценария использования. Показывает, какие сообщения отправляются между объектами и в каком порядке. (Рис.8)

Диаграмма деятельности: отображает поток работ и действий в системе, последовательность шагов и условия перехода между ними. (Рис.9)

Задание 2

Я провела анализ скоростных показателей программы и других характеристик. Я зафиксировала время выполнения каждой операции и составил отчет о результатах:

Время выполнения `resize_image`: 0.0000010000 секунд

Время выполнения `rotate_image`: 0.0000008000 секунд

Время выполнения `save_image`: 0.0000008000 секунд

Оригинальное изображение: 1920x1080, Каналы: 3, Занимаемая память: 6232328 байт

Повернутое изображение: 1080x1920, Каналы: 3, Занимаемая память: 6232328 байт

Задание 3

В данном задании я провела анализ средств разработки программ на Python.

Среда разработки: PyQt (Выбрана мной, далее расписано)

Преимущества:

1. Удобный графический интерфейс для создания приложений: PyQt предоставляет мощные инструменты для создания графических пользовательских интерфейсов (GUI), что позволяет разработчикам быстро разрабатывать приложения с современным интерфейсом.
2. Возможность быстрого прототипирования: Благодаря простоте использования и множеству готовых компонентов, разработчики могут быстро создавать прототипы своих идей.
3. Кроссплатформенность: PyQt работает на различных операционных системах (Windows, macOS, Linux), что позволяет разрабатывать приложения, которые будут работать на разных платформах без значительных изменений в коде.

Недостатки:

1. Может быть сложной для новичков из-за большого количества возможностей и настроек: Из-за богатства функционала PyQt может быть сложно освоить начинающим разработчикам. Потребуется время для изучения документации и понимания архитектуры.

Visual Studio Code (VS Code)

Преимущества:

Легкость и скорость: VS Code — это легкий редактор, который быстро запускается и работает.

Расширяемость: Поддерживает множество расширений, включая те, которые улучшают работу с Python (например, Python extension от Microsoft).

Кроссплатформенность: Работает на Windows, macOS и Linux.

Недостатки:

Настройка: Потребуется время на настройку редактора под свои нужды, особенно для работы с Python.

Отсутствие некоторых функций IDE: Хотя VS Code предоставляет много возможностей, он может не иметь всех функций полноценной IDE.

3. Jupyter Notebook

Преимущества:

Интерактивность: Позволяет запускать код по ячейкам, что упрощает экспериментирование и визуализацию данных.

Отлично подходит для анализа данных: Широко используется в научных исследованиях и анализе данных благодаря поддержке графиков и визуализации.

Markdown поддержка: Можно добавлять текстовые описания и формулы, что делает документы более информативными.

Недостатки:

Не предназначен для крупных проектов: Лучше всего подходит для небольших скриптов и анализа данных, а не для создания крупных приложений.

Ограниченные возможности отладки: Отладка кода не так удобна, как в традиционных IDE.

Заключение

Производственная практика по профилю специальности 09.02.07 «Информационные системы и программирование» была направлена на формирование необходимых компетенций, позволяющих эффективно работать в области разработки программного обеспечения. В ходе практики были выполнены задания, которые способствовали приобретению практического опыта и углублению знаний в ключевых аспектах разработки.

Одной из основных задач практики было ревьюирование программных модулей, что позволило не только улучшить качество кода, но и освоить методологии процессов разработки ПО. Участие в процессе ревью дало возможность оценить существующие решения, выявить недостатки и предложить оптимизации, что в свою очередь способствовало повышению эффективности работы команды.

Таким образом, производственная практика не только подтвердила теоретические знания, полученные в ходе обучения, но и предоставила возможность применить их на практике, развивая ключевые компетенции для будущей профессиональной деятельности в области информационных технологий.

Источники информации

1. <https://nationalteam.worldskills.ru/skills/proektirovanie-diagrammy-deyatelnosti-uml-activity-diagram/?ysclid=m4766kcc16455527304>
2. <https://habr.com/ru/articles/508710/>
3. https://www.mallenom.ru/Privacy_Policy_MallenomRU.pdf?ysclid=m49oc5hbnx168750212
4. https://www.audit-it.ru/contragent/1113528001685_ooo-mallenom-sistemas?ysclid=m49ociygbe808169762
5. <https://python-scripts.com/pyqt5?ysclid=m49odnx2ju951173369>
6. <https://multiurok.ru/files/umk-pm-02-reviuirovanie-programmnykh-modulei-09-02.html?ysclid=m49oe87az2847940667>
7. <https://practicum.yandex.ru/blog/uml-diagrammy/>

Приложения

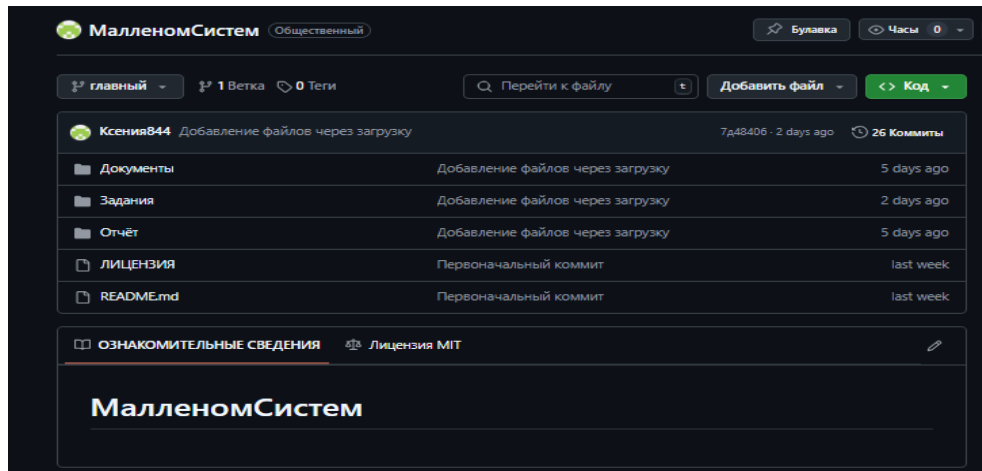


Рис.1 GIT репозиторий с заданиями, отчетом, документами.

```
from PIL import Image

class ImageHandler:
    def __init__(self):
        self.image = None
        self.image_path = None

    def load_image(self, path):
        self.image_path = path
        self.image = Image.open(path)
        return self.image

    def resize_image(self, width, height):
        if self.image:
            self.image = self.image.resize((width, height))
            return self.image
        raise ValueError("Изображение не загружено.")

    def rotate_image(self, angle):
        if self.image:
            self.image = self.image.rotate(angle)
            return self.image
        raise ValueError("Изображение не загружено.")

    def save_image(self, path):
        if self.image:
            self.image.save(path)
```

Рис.2 Модуль обработки и работы с изображениями

```
File Edit Format Run Options Window Help

import sys
from PyQt5.QtWidgets import QApplication
from user_interface import UserInterface

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = UserInterface()
    window.show()
    sys.exit(app.exec_())
```

Рис.3 Модуль взаимодействия с пользователем и формирование и хранения данных

```

from PyQt5.QtWidgets import (QMainWindow, QPushButton, QLabel,
                              QVBoxLayout, QHBoxLayout,
                              QWidget, QFileDialog, QMessageBox, QInputDialog)
from PyQt5.QtGui import QPixmap, QTransform
from PyQt5.QtCore import Qt
from image_handler import ImageHandler

class UserInterface(QMainWindow):
    def __init__(self):
        super().__init__()
        self.image_handler = ImageHandler()
        self.init_ui()

    def init_ui(self):
        self.setWindowTitle("Утилита работы с изображениями, Аксёнова Ксения Максимовна")
        self.setGeometry(100, 100, 800, 600) # Установка начального размера окна

        # Основной виджет и макет
        central_widget = QWidget()
        self.setCentralWidget(central_widget)
        layout = QVBoxLayout(central_widget)

        # Верхний горизонтальный макет для кнопок
        button_layout = QHBoxLayout()

        # Кнопки
        load_button = QPushButton("Загрузить изображение")
        load_button.clicked.connect(self.load_image)
        button_layout.addWidget(load_button)

        resize_button = QPushButton("Изменить размер изображения")
        resize_button.clicked.connect(self.resize_image)

```

Рис.4 Модуль пользовательского интерфейса (не весь, лишь его часть)

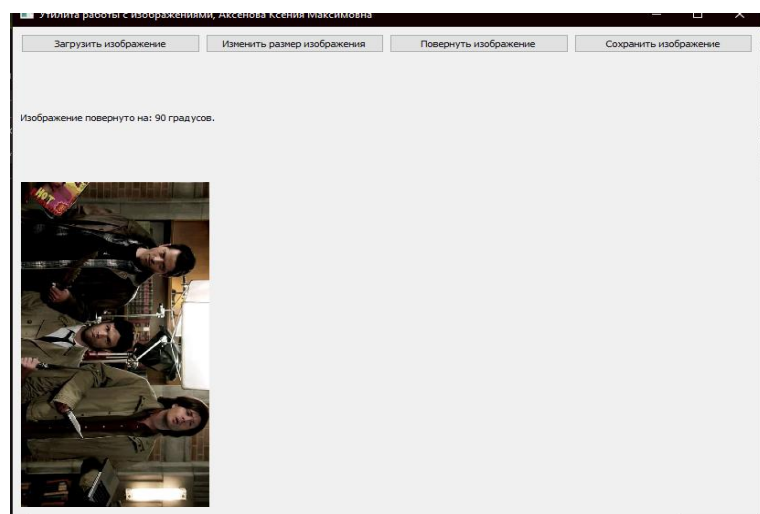
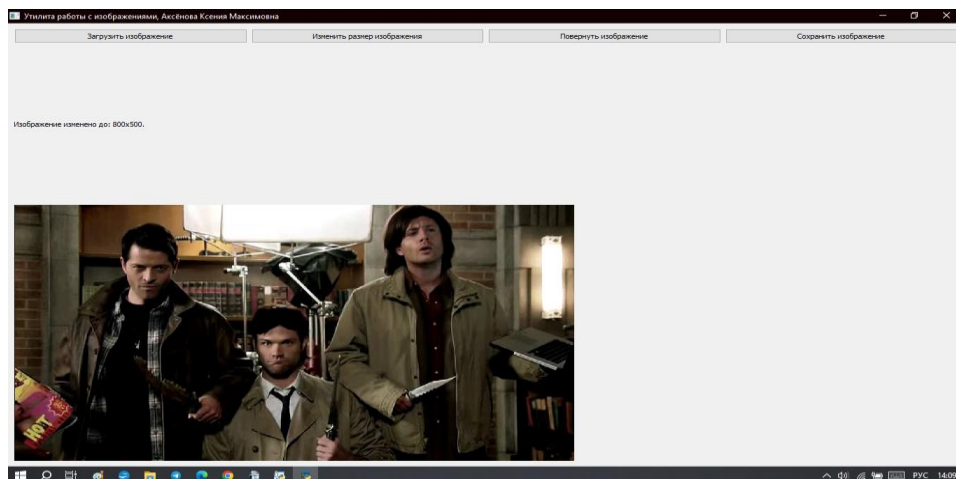


Рис.5 Вид итогового приложения (Изменение размера изображения, поворот изображения)

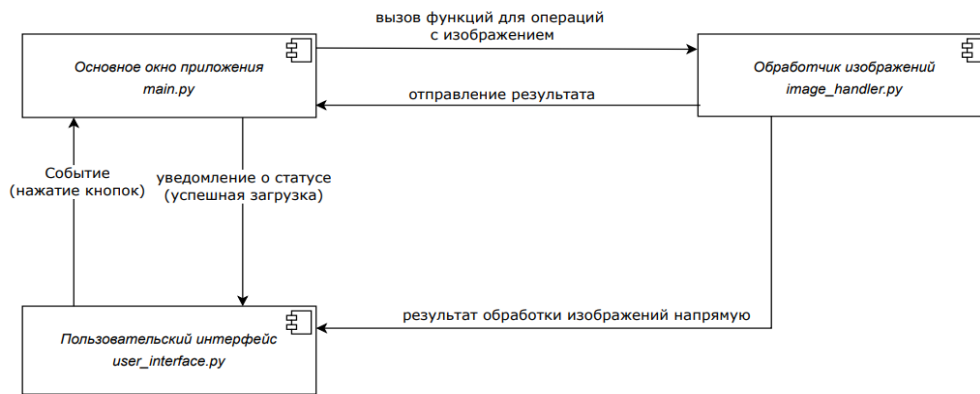


Рис.6 Диаграмма компонентов

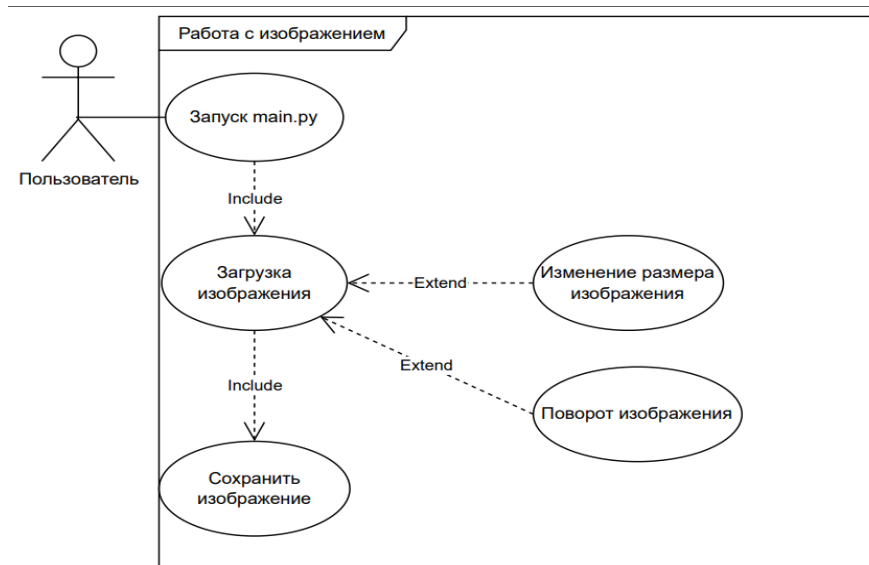


Рис.7 Диаграмма сценариев использования

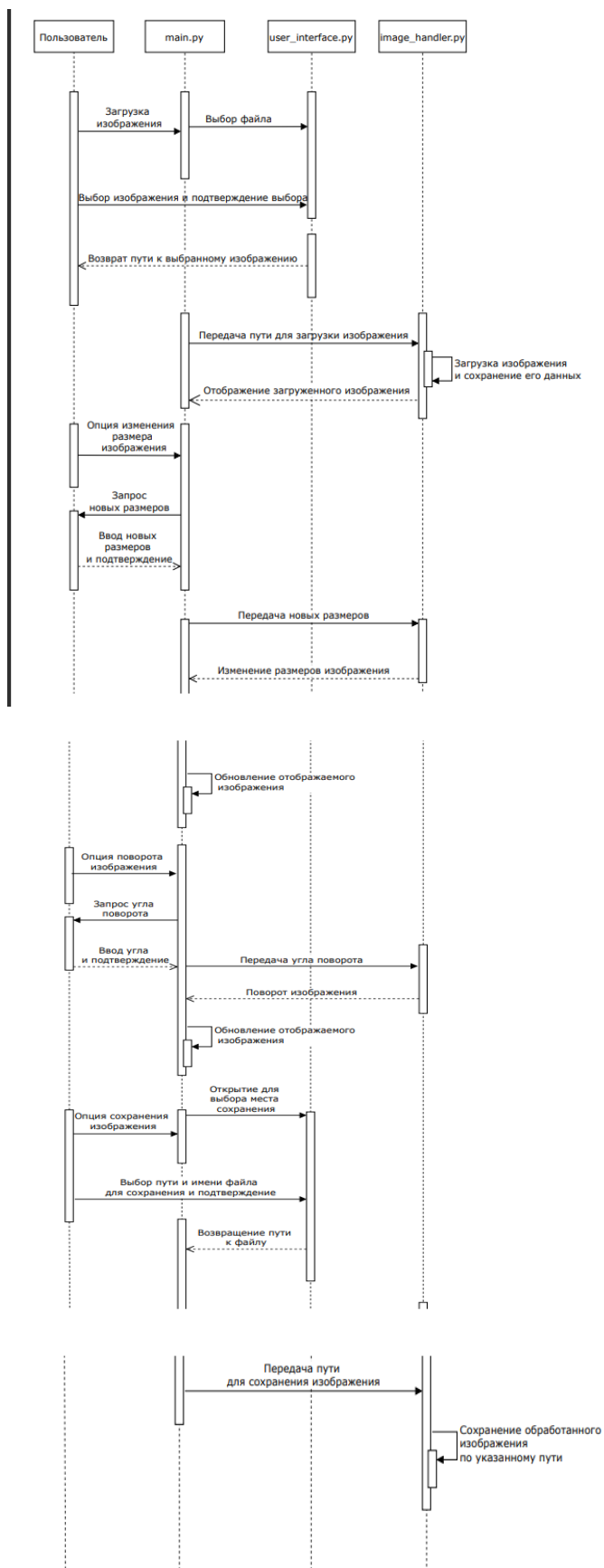


Рис.8 Диаграмма последовательностей

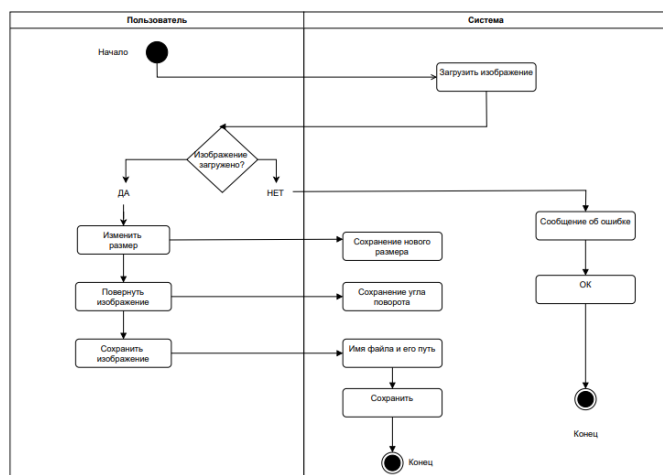


Рис.9 Диаграмма деятельности