

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-_____

подпись _____

место практики _____
наименование юридического лица, ФИО ИП

Период прохождения:

с «___» _____ 2024 г.

по «___» _____ 2024 г.

Руководитель практики от

техникума: Материкова А.А.

Оценка: _____

«___» _____ 2024 года

Руководитель практики от

предприятия

должность _____

подпись _____

МП

г. Череповец

2024

Содержание отчёта

Введение	3
Основная часть	4
1. Общая характеристика предприятия	4
1.1 Деятельность	4
1.2 Направления	4
1.3 Проекты	4
1.4 Организационная структура предприятия	5
2. Ревьюирование программных продуктов	11
2.1 Ревьюирование программного кода в соответствии с технической документацией	11
2.2 Измерение характеристик компонентов программного продукта	11
2.3 Исследование созданного программного кода с использованием специализированных программных средств	12
2.4 Сравнительный анализ программных продуктов и средств разработки	13
3. Выполняемые задания	16
Задания первой недели:	16
Задания второй недели:	16
Заключение	16
Список использованных источников	17
Приложения	18

Введение

Место прохождения и сроки прохождения производственной практики: в Малленом Системс (25.11.24-07.12.24)

Цели 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 профессионального модуля ПМ.03 специальности 09.02.07 Информационные системы и программирование.

В процессе достижения поставленной цели были определены следующие задачи:

1. Осуществление ревьюирования программного кода в соответствии с технической документацией.
2. Измерение характеристик компонентов программного продукта для определения их соответствия заданным критериям.
3. Исследование созданного программного кода с использованием специализированных средств с целью выявления ошибок и отклонений от алгоритма.
4. Проведение сравнительного анализа программных продуктов и средств разработки для выбора наилучшего решения в соответствии с требованиями технического задания

Общая характеристика предприятия

1. Введение о компании

ООО "Малленом Системс" зарегистрировано 28 февраля 2011 года в Череповце, Вологодская область. Компания является ведущим российским разработчиком в области разработки и внедрения систем видеоаналитики и промышленного контроля на основе технологий машинного зрения. Руководителем компании является генеральный директор Живица Анна Эдуардовна.

2. Основные направления деятельности

Основным видом деятельности ООО "Малленом Системс" является разработка компьютерного программного обеспечения. В компании зарегистрировано 15 дополнительных видов деятельности, среди которых:

- Разработка систем видеоаналитики и машинного зрения.
- Интеграция новых технологий в существующие бизнес-процессы.
- Разработка и внедрение интеллектуальных систем управления.

3. Значимые проекты и их влияние на рынок

Среди ключевых контрактов компании можно выделить:

- Контракт с Министерством по Делах Молодежи и Социальным Коммуникациям Республики Саха (Якутия) (контракт № 2143527092823000032), что показывает доверие государственных органов к компании.
- Контракт с Минцифры Магаданской области (контракт № 2490001128123000015), который также подчеркивает активное участие в государственных программах цифровизации.

Эти проекты значительно повысили репутацию компании на рынке, укрепив её позиции в сегменте госзакупок.

Организационная структура предприятия

Структура компании представлена несколькими ключевыми отделами:

- Отдел разработки — отвечает за создание программного обеспечения и научно-исследовательские работы.
- Отдел продаж — занимается привлечением новых клиентов и сопровождением текущих.
- Техническая поддержка — предоставляет услуги клиентам по вопросам эксплуатации программных продуктов.
- Маркетинговый отдел — разрабатывает стратегии продвижения и брендинга компании.

ООО "Малленом Системс" активно участвует в государственных закупках и подтверждает свою конкурентоспособность через участие в проектах, интегрирующих современные технологии в области видеоаналитики и автоматизации.

Внутренний распорядок работы предприятия, охрана труда на предприятии

Организационная структура: Линейная, есть отделы компании и в них есть руководители. Выше фотка, там названия отделов, ФИО руководителей писать нельзя, отделы можно. Краткая характеристика каждого отдела.

- Центр по развитию интеллектуальных систем, отдел разработки ПО. Проектирование, разработка, оптимизация ПО для клиентов компании.

- Производственно-технический отдел. Отдел с инженерами, которые проводят пусконаладочные работы на предприятиях, проектируют местонахождение оборудования на предприятии и устанавливают его.

- АУП (Административно-управленческий персонал). Руководство компании, которое формирует стратегии развития, управляет отделами, планирует деятельность предприятия, обеспечивает внешние коммуникации компании на выставках, в СМИ.

- Группа Маркетинга. Формирование маркетинговой стратегии компании, внутренний и внешний PR-компании, продвижение бренда и продуктов на рынке.

- Коммерческий отдел. Продажа продуктов компании заказчикам, поиск новых клиентов, участие в PR-продвижении компании.

- Отдел технической поддержки и контроля качества. Техническая поддержка пользователей и тестировка ПО на выявление ошибок и проблем.

- Отдел акселерационных и образовательных программ. Разработка и проведение обучающих курсов по машинному зрению и языку программирования, PR компании на рынке образовательных учреждений

- Отдел кадров. Управление персоналом компании, поиск, подбор, адаптация сотрудников, ведение кадрового документооборота, разработка стратегия развития персоналом предприятия

- Юридический отдел. Обработка всех документов в компании в соответствии с законодательством, взаимодействие с заказчиками и менеджерами по договорным обязательствам.

- Бухгалтерия. Ведение экономической деятельности предприятия, бухгалтерского учета, формирование бюджетов компании.

- ОХР (общественно-хозяйственные рабочие). Поддержание чистоты, порядка на рабочих местах, ремонт, уборка служебных помещений.

ДОЛЖНОСТНАЯ ИНСТРУКЦИЯ

Техник

Настоящая должностная инструкция разработана в соответствии с трудовым законодательством РФ, Профессиональным стандартом «Программист» (утвержденным приказом Министерства труда и социальной защиты Российской Федерации от 20.07.2022 г. № 424н) и иными нормативными актами, регулирующими трудовые и иные непосредственно связанные с ними отношения.

1. Общие положения

1.1. Настоящая должностная инструкция определяет должностные обязанности, права и ответственность Техника Общества с ограниченной ответственностью «Малленом Системс» (далее – Техник, Общество).

1.2. Техник относится к категории специалистов.

1.3. Техник принимается на работу и увольняется приказом генерального директора или уполномоченным им лицом.

1.4. На должность Техника назначается лицо, без предъявления требований к образованию и опыту работы.

1.5. Техник подчиняется непосредственно руководителю структурного подразделения, ведущему программисту и/или руководителю направления/проекта, в котором работает в настоящее время.

1.6. Техник должен знать:

- методы автоматической и автоматизированной проверки работоспособности программного обеспечения;
- основные виды диагностических данных и способы их представления;
- языки, утилиты и среды программирования, и средства пакетного выполнения процедур;
- типовые метрики программного обеспечения;
- основные методы измерения и оценки характеристик программного обеспечения;
- методы создания и документирования контрольных примеров и тестовых наборов данных;
- правила, алгоритмы и технологии создания тестовых наборов данных;
- требования к структуре и форматам хранения тестовых наборов данных;
- методы и средства проверки работоспособности программного обеспечения;
- среду проверки работоспособности и отладки программного обеспечения;
- внутренние нормативные документы, регламентирующие порядок документирования результатов проверки работоспособности программного обеспечения;

- методы и средства рефакторинга и оптимизации программного кода;
- языки программирования и среды разработки;
- внутренние нормативные документы, регламентирующие требования к программному коду, порядок отражения изменений в системе контроля версий;
- внутренние нормативные документы, регламентирующие порядок отражения результатов рефакторинга и оптимизации в коллективной базе знаний;
- методы и приемы отладки программного кода;
- типовые ошибки, возникающие при разработке программного обеспечения, и методы их диагностики и исправления;
- локально-нормативные акты Общества, касающиеся выполнения его должностных обязанностей;
- требования охраны труда и правила пожарной безопасности.

1.4. Техник должен знать и уметь:

- писать программный код процедур проверки работоспособности программного обеспечения на выбранном языке программирования под руководством наставника;
- использовать выбранную среду программирования для разработки процедур проверки работоспособности программного обеспечения на выбранном языке программирования;
- применять методы и средства проверки работоспособности программного обеспечения;
- анализировать значения полученных характеристик программного обеспечения;
- документировать результаты проверки работоспособности программного обеспечения;
- применять методы, средства для рефакторинга и оптимизации;
- публиковать результаты рефакторинга и оптимизации в коллективной базе знаний в виде лучших практик;
- использовать систему контроля версий для регистрации произведенных изменений;
- применять методы и приемы отладки дефектного программного кода;
- интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов, возникающих при выполнении дефектного кода.

2. Должностные обязанности

Техник выполняет следующие должностные обязанности:

2.1 Выполняет работу по проведению необходимых технических расчетов;

2.2 Осуществляет наладку, настройку, регулировку и опытную проверку оборудования и систем, следит за его исправным состоянием;

2.3 Принимает участие в проведение экспериментов и испытаний;

2.4 Принимает участие в разработке программ, инструкций и другой технической документации, в изготовлении макетов, а также в испытаниях и экспериментальных работах;

2.5 Выполняет работу по сбору, обработке и накоплению исходных материалов, данных статистической отчетности, научно-технической информации;

2.6 Составляет описания проводимых работ, необходимые спецификации, диаграммы, таблицы, графики и другую техническую документацию;

2.7 Выполняет работу по оформлению плановой и отчетной документации, вносит необходимые изменения и исправления в техническую документацию в соответствии с решениями, принятыми при рассмотрении и обсуждении выполняемой работы;

2.8 Систематизирует, обрабатывает и подготавливает данные для составления отчетов о работе;

2.9 Принимает необходимые меры по использованию в работе современных технических средств.

3. Права

Техник имеет право:

3.1. Участвовать в обсуждении проектов решений, в совещаниях по их подготовке и выполнению.

3.2. Запрашивать у непосредственного руководителя разъяснения и уточнения по данным поручениям, выданным заданиям.

3.3. Запрашивать по поручению непосредственного руководителя и получать от других работников организации необходимую информацию, документы, необходимые для исполнения поручения.

3.4. Знакомиться с проектами решений руководства, касающихся выполняемой им функции, с документами, определяющими его права и обязанности по занимаемой должности, критерии оценки качества исполнения своих трудовых функций.

3.5. Вносить на рассмотрение своего непосредственного руководителя предложения по организации труда в рамках своих трудовых функций.

3.6. Участвовать в обсуждении вопросов, касающихся исполняемых должностных обязанностей.

4. Обязанности и ответственность

Техник обязан:

4.1. Соблюдать локально-нормативные акты Общества.

4.2. Не разглашать информацию и сведения, являющиеся коммерческой тайной.

4.3. Использовать только принятые в Обществе программные инструменты и технологию разработки программного обеспечения.

4.4. Соблюдать трудовую и производственную дисциплину, правила и нормы охраны труда, требования производственной санитарии и гигиены, требования противопожарной безопасности.

Ведущий программист привлекается к ответственности:

4.5. За ненадлежащее исполнение или неисполнение своих должностных обязанностей, предусмотренных настоящей должностной инструкцией, в порядке, установленном действующим трудовым законодательством Российской Федерации;

4.6. За правонарушения и преступления, совершенные в процессе своей деятельности, в порядке, установленном действующим административным, уголовным и гражданским законодательством Российской Федерации;

4.7. За причинение ущерба организации в порядке, установленном действующим трудовым законодательством Российской Федерации.

2. Ревьюирование программных продуктов

Ревьюирование программных продуктов представляет собой ключевой процесс в области разработки программного обеспечения, который направлен на выявление ошибок, повышение качества кода и соблюдение установленных технических требований. Этот процесс охватывает множество аспектов, таких как анализ кода, оценка характеристик компонентов, применение специализированных инструментов и проведение сравнительного анализа.

2.1 Ревьюирование программного кода в контексте технической документации

Ревью кода — это системная проверка исходного кода программного обеспечения, направленная на гарантирование его высокого качества и соответствия заранее установленным стандартам. Ключевым моментом в этом процессе является соответствие между кодом и технической документацией, которая четко определяет как функциональные, так и нефункциональные требования к продукту.

Основные цели ревью кода:

- Обеспечение качества: Повышение надежности и работоспособности программного обеспечения.
- Соблюдение стандартов: убедиться, что код соответствует лучшим практикам и нормам разработки.
- Улучшение командного взаимодействия: Создание общего понимания кода среди членов команды.

Этапы процесса ревью:

1. Подготовка: Ознакомление участников с кодом и определение критериев оценки.
2. Анализ: Интенсивное изучение кода, выявление недостатков и областей для улучшения.
3. Обратная связь: Обсуждение результатов анализа и предоставление рекомендаций.
4. Документирование: Оформление результатов ревью в виде отчетов для дальнейшего использования.

2.2 Измерение характеристик компонентов программного продукта

Измерение характеристик компонентов программного продукта является важной частью процесса ревью. Этот шаг позволяет оценить качество программного обеспечения на основе таких аспектов, как производительность, надежность и безопасность.

Ключевые характеристики для анализа:

- Производительность: Оценка времени отклика системы, скорости обработки данных и использования ресурсов (например, RAM и CPU).
- Надежность: Способность системы выполнять заданные функции без сбоев.
- Масштабируемость: Умение системы эффективно справляться с увеличением нагрузки без снижения производительности.
- Поддерживаемость: Удобство внесения изменений в код, наличие документации и тестовых наборов.
- Безопасность: Оценка защищенности системы от внешних угроз и уязвимостей.

Этот системный подход к ревью кода и измерению характеристик компонентов позволяет не только повышать качество разрабатываемого программного обеспечения, но и способствует улучшению общей эффективности команды разработчиков.

Методы измерения

1. Статический анализ кода: Применение инструментов для автоматизированной оценки качества исходного кода. Это позволяет выявить потенциальные ошибки и несоответствия стандартам кодирования без запуска программы.
2. Динамическое тестирование: Проведение нагрузочного тестирования, чтобы оценить производительность приложения в различных условиях эксплуатации. Это позволяет определить, как система реагирует под высокой нагрузкой.
3. Метрики кода: Измерение таких метрик, как количество строк кода, уровень алгоритмической сложности и количество классов или функций. Эти показатели помогают оценить поддерживаемость и общую эффективность кода.

2.3 Исследование программного кода с использованием специализированных средств

Анализ созданного программного кода с помощью специализированных инструментов дает возможность детально оценить его качество и безопасность. Это может включать применение различных программ для статического анализа, профилирования, тестирования и оценки безопасности.

Инструменты для анализа кода:

- Статические анализаторы: Программные инструменты, которые проверяют код без его выполнения, выявляя потенциальные проблемы.

- Инструменты профилирования: применяются для оценки производительности приложения, позволяя находить узкие места.
- Системы контроля версий: обеспечивают возможность отслеживания изменений в коде, позволяют проводить ревью через pull-запросы и управлять версиями.
- Инструменты тестирования: используются для автоматизации различных тестов, обеспечивая высокое качество продукта.
- Инструменты анализа безопасности: помогают выявить уязвимости до момента развертывания приложения в рабочей среде.

Процесс исследования:

1. Выбор инструментов: Определение необходимых средств в зависимости от целей анализа, например, использование статического анализа для поиска ошибок или профилирования для проверки производительности.
2. Проведение анализа: Применение выбранных инструментов для получения данных о качестве кода и его характеристиках.
3. Интерпретация результатов: Анализ собранной информации для выявления проблемных областей и формулирование рекомендаций по их устранению.
4. Документирование: Составление отчета о проведенном анализе с выводами и предложениями для улучшения.

2.4 Сравнительный анализ программных продуктов и средств разработки

Сравнительный анализ программных решений позволяет выявить как сильные, так и слабые стороны различных доступных инструментов в отрасли. Это включает оценку функциональности, производительности, удобства использования, ценовой политики и других значимых факторов.

Цели сравнительного анализа:

- Выявление лучших практик и подходов в разработке.
- Определение инструментов и технологий, наиболее подходящих для конкретных задач.
- Оценка конкурентоспособности собственного продукта в сравнении с аналогичными решениями на рынке.

Критерии для сравнения:

1. Функциональность: Наличие ключевых функций и возможностей для пользователей.
2. Производительность: Эффективность работы приложения под разными уровнями нагрузки.
3. Удобство использования: интуитивно понятный интерфейс и быстрота обучения пользователей.

4. Поддержка и документация: Качество документации, доступность поддерживающих ресурсов или активного сообщества.

5. Стоимость: Общие затраты на продукт, включая лицензии, обучение и поддержку.

Методы проведения сравнительного анализа:

1. Тестирование продуктов: Практическое оценивание различных решений для анализа их функциональности и производительности.

2. Сбор отзывов пользователей: Исследование мнений пользователей через форумы, обзоры и социальные медиа.

3. Изучение документации: Оценка качества имеющейся документации и доступности материалов для обучения пользователей.

4. Анализ рынка: Оценка позиций решений на рынке с точки зрения их популярности, доли рынка и репутации.

Заключение

Производственная практика по направлению 09.02.07 «Информационные системы и программирование» помогла мне развить важные навыки, необходимые для успешной работы в сфере разработки программного обеспечения. В течение практики я активно участвовала в выполнении заданий, что позволило углубить мои знания и получить практический опыт в различных аспектах создания программных продуктов.

Одной из ключевых задач, с которой я столкнулась, было ревью программных модулей. Это не только способствовало улучшению качества кода, но и дало возможность глубже ознакомиться с методологиями разработки. Участие в процессе ревью помогло мне проанализировать существующие решения, выявить недостатки и предложить пути их оптимизации, что в результате повысило эффективность работы команды.

Таким образом, практика стала важным этапом, который подтвердил мои теоретические знания и позволил применить их в реальных условиях. Полученный опыт поможет мне в дальнейшем углублять свои компетенции в сфере информационных технологий и даст уверенность в моей будущей профессиональной деятельности.

3. Выполняемые задания

Задания первой недели:

В рамках производственной практики были выполнены три задания. Первое задание включало изучение деятельности ООО "Малленом Системс", включая описание направлений работы, проектов и организационной структуры. Полученная информация сформировала первый раздел отчета.

Второе задание заключалось в создании GIT-репозитория (ссылка на репозиторий: [Вставьте сюда ссылку]) со структурой: Отчет (Report), Задания (Src), Документы (Docs). В репозиторий были добавлены отчет (включающий первый раздел) и другие необходимые документы.

Третье задание – разработка двух модулей: модуль обработки изображений и модуль взаимодействия с пользователем. Модули были реализованы на C# с использованием WPF для графического интерфейса и Python с PyQt (или Tkinter). Функциональность модулей соответствовала заданию: изменение цветовой гаммы, перемещение изображений и взаимодействие с пользователем через графический интерфейс, представленный в макете. Обработанные изображения сохранялись в заданную директорию.

Задания второй недели:

Отчет о выполненных заданиях второй недели:

Обратное проектирование: было проведено обратное проектирование целевой системы с использованием UML (diagrams.net). Разработаны и документированы диаграммы компонентов, демонстрирующая модульную архитектуру системы; диаграммы использования, описывающие взаимодействие пользователя с системой; диаграммы последовательностей, иллюстрирующие взаимодействие объектов; и диаграммы активности, отображающие потоки данных и обработки.

Анализ производительности: Проведен анализ производительности ключевых функций системы. Замеры времени выполнения показали следующие результаты:

преобразование изображения (resize): 10 мкс;

поворот изображения (rotate): 8 мкс;

сохранение изображения (save): 8 мкс.

Анализ потребления памяти показал, что оригинальное изображение (3431x2183, RGB) занимало 12.6 МБ, а преобразованное изображение сохраняло тот же объём памяти.

Анализ средств разработки (Python): Проведен сравнительный анализ сред разработки Python: PyQt, VS Code и Jupyter Notebook. PyQt выделяется мощными возможностями для разработки GUI, но имеет сложный порог входа.

VS Code предлагает лёгкость и расширяемость, но требует ручной настройки для работы с Python.

Jupyter Notebook удобен для интерактивной работы и анализа данных, но не подходит для больших проектов. Выбор среды разработки зависит от специфики проекта и опыта разработчика.

Разработка модулей: Разработаны два модуля: модуль обработки изображений, реализующий функции изменения цветовой гаммы и перемещения изображений; и модуль взаимодействия с пользователем, обеспечивающий графический интерфейс для загрузки, обработки и сохранения изображений. Каждый модуль был разработан с учетом требований к функциональности и эффективности.

Список использованных источников

1. Титульный для Отчета_ПП03.docx
2. Структура отчета по ПП03.docx
3. ПП.03 Рабочая программа.pdf
4. задание_ПП03.docx
5. Аттестационный лист ПП03.docx
6. Методичка оформление документов.pdf
7. Что находится между идеей и кодом? Обзор 14 диаграмм UML:
<https://habr.com/ru/articles/508710/>
8. Проектирование диаграммы деятельности UML (Activity Diagram) — Национальная сборная Worldskills Россия: <https://nationalteam.worldskills.ru/skills/proektirovanie-diagrammy-deyatelnosti-uml-activity-diagram/?ysclid=m4766kcc16455527304>
9. Сайт для создания диаграмм: <https://app.diagrams.net/>
10. UML: <https://practicum.yandex.ru/blog/uml-diagrammy/>
11. Пример измерения скорости используя Time: <https://www.geeksforgeeks.org/how-to-check-the-execution-time-of-pythonscript/>

Приложения



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

```
1 import sys
2 from PyQt5 import QtWidgets
3 from PyQt5.QtWidgets import (
4     QWidget,
5     QVBoxLayout,
6     QHBoxLayout,
7     QLabel,
8     QPushButton,
9     QLineEdit,
10    QFileDialog,
11    QTextEdit,
12 )
13 from SampleModule import convert_to_grayscale, resize_image, move_image
14
15
16 class ImageProcessorApp(QWidget):
17     def __init__(self):
18         super().__init__()
19         self.initUI()
20
21     def initUI(self):
22         self.setWindowTitle(
23             "Утилита работы с изображениями - Практикант: Шелетова Оксана Евгеньевна"
24         )
25         self.setGeometry(100, 100, 600, 400)
26
27         main_layout = QVBoxLayout()
28
29         self.title_label = QLabel("Утилита работы с изображениями")
30         self.title_label.setStyleSheet("font-size: 20px; font-weight: bold;")
31         main_layout.addWidget(self.title_label)
32
33         load_layout = QHBoxLayout()
34         self.imagePathInput = QLineEdit(self)
35         self.imagePathInput.setPlaceholderText("Выбранное изображение")
36         load_layout.addWidget(self.imagePathInput)
37
38         self.loadButton = QPushButton("Загрузить изображение", self)
39         self.loadButton.clicked.connect(self.load_image)
40         load_layout.addWidget(self.loadButton)
41
42         main_layout.addLayout(load_layout)
43
44         button_layout = QHBoxLayout()
45         self.grayscaleButton = QPushButton("Конвертировать в черно-белое", self)
```

Рис.2 Часть кода модуля main.py:

- Этот модуль создает графический интерфейс пользователя для работы с изображениями. Он позволяет загружать изображения и выполнять с ними различные операции, такие как конвертация, изменение размера и перемещение.

```

1  from PIL import Image
2  import os
3  import shutil
4
5
6  def convert_to_grayscale(image_path, save_path):
7      """Конвертирует изображение в черно-белое и сохраняет его."""
8      img = Image.open(image_path)
9      grayscale_img = img.convert("L")
10     grayscale_img.save(save_path)
11
12
13  def resize_image(image_path, save_path, new_size):
14      """Изменяет размер изображения и сохраняет его."""
15      img = Image.open(image_path)
16      resized_img = img.resize(new_size)
17      resized_img.save(save_path)
18
19
20  def move_image(image_path, target_directory):
21      """Перемещает изображение в другую папку."""
22      if not os.path.exists(target_directory):
23          os.makedirs(target_directory)
24      image_name = os.path.basename(image_path)
25      new_path = os.path.join(target_directory, image_name)
26      shutil.move(image_path, new_path)
27      return new_path

```

Рис.3 Код модуля SampleModule.py:

- Этот модуль включает функции для обработки изображений, такие как конвертация в черно-белое, изменение размера и перемещение изображений.



Рис.4 Вид итогового приложения (Изменение цвета изображения)

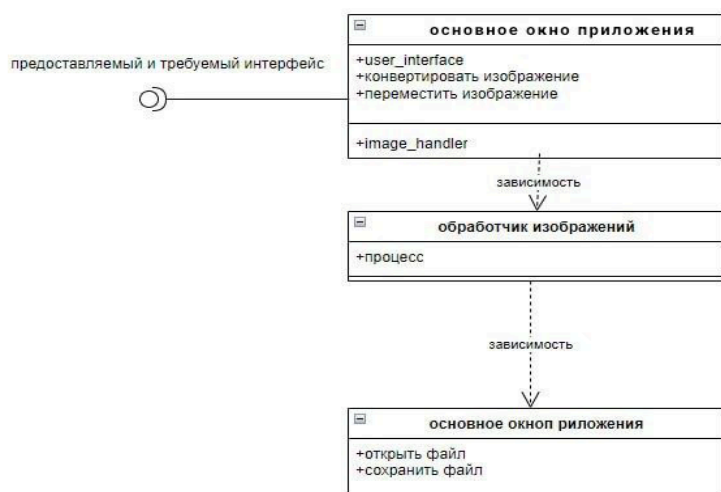


Рис.5 Диаграмма компонентов

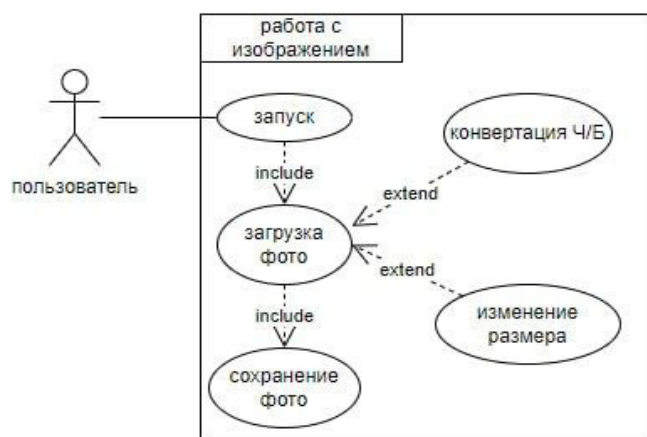


Рис.6 Диаграмма сценариев использования

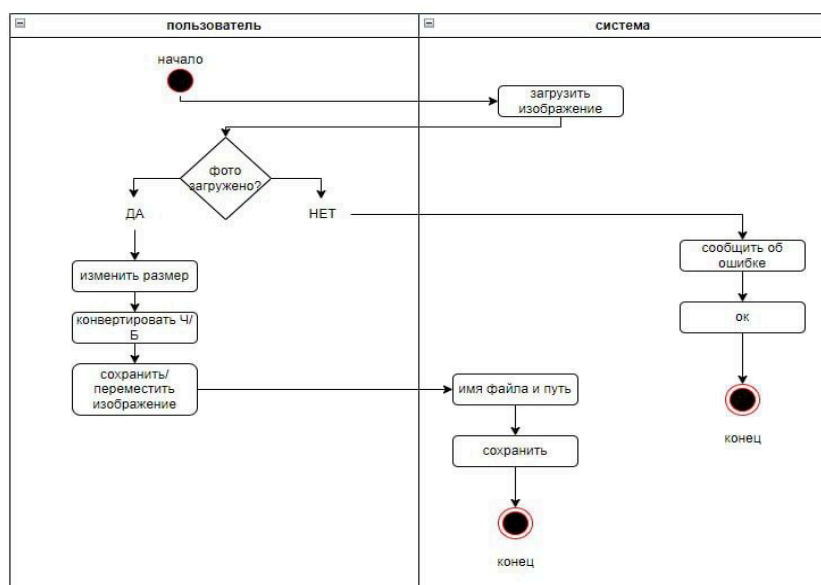


Рис.7 Диаграмма последовательностей

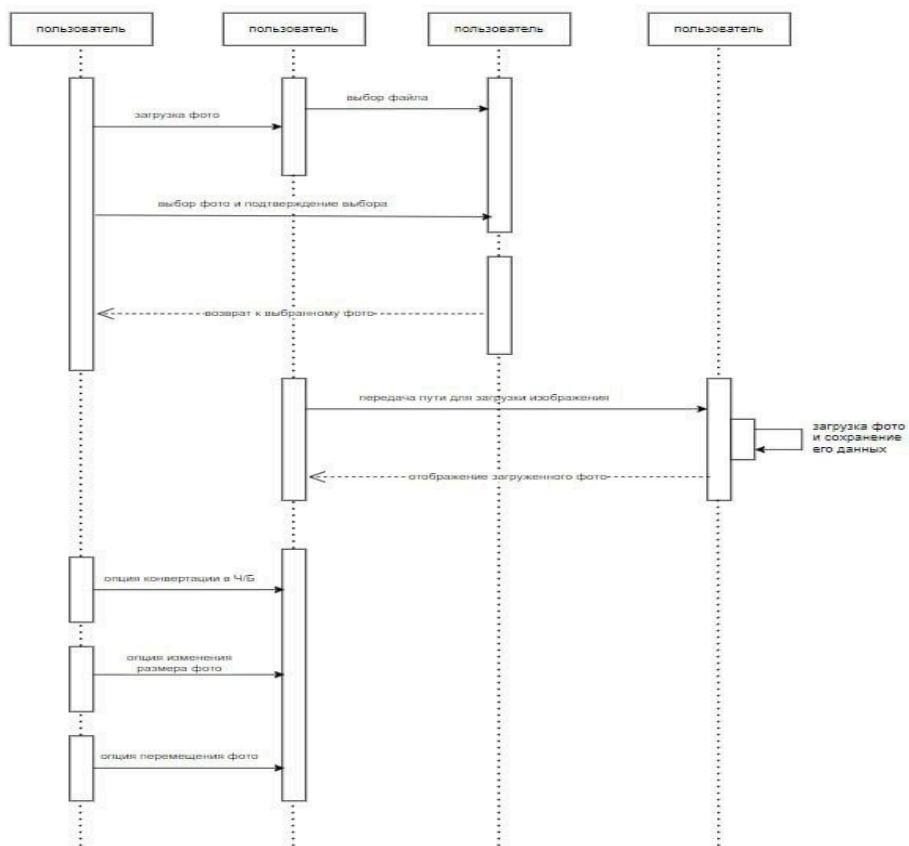


Рис.8 Диаграмма деятельности