

титульный

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ.....</b>	<b>4</b>
1.1. Организационная структура предприятия.....	4
1.2. Внутренний распорядок работы предприятия, охрана труда ИТ-специалистов.....	4
1.3. Должностные инструкции ИТ-специалистов предприятия.....	5
<b>2 РЕВЬЮИРОВАНИЕ ПРОГРАММНЫХ ПРОДУКТОВ.....</b>	<b>7</b>
2.1. Ревьюирование программного кода в соответствии с технической документацией.....	7
2.2. Измерение характеристик компонент 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
<b>3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ.....</b>	<b>10</b>
3.1. Анализ соответствия программного кода требованиям технической документации.....	10
3.2. Оценка качественных показателей компонентов программного обеспечения.....	10
3.3. Экспертиза кодовой базы с применением инструментов анализа.....	11
3.4. Сопоставительная оценка инструментов разработки программного обеспечения.....	12
<b>ЗАКЛЮЧЕНИЕ.....</b>	<b>14</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....</b>	<b>15</b>
<b>ПРИЛОЖЕНИЕ А.....</b>	<b>16</b>
<b>ПРИЛОЖЕНИЕ Б.....</b>	<b>17</b>
<b>ПРИЛОЖЕНИЕ В.....</b>	<b>18</b>
<b>ПРИЛОЖЕНИЕ Г.....</b>	<b>19</b>
<b>ПРИЛОЖЕНИЕ Д.....</b>	<b>20</b>
<b>ПРИЛОЖЕНИЕ Е.....</b>	<b>21</b>

## ВВЕДЕНИЕ

Производственная практика по теме «Ревьюирование программного кода» проходила в ООО «Малленом Системс» с 17 по 30 ноября 2025 года в дистанционном формате.

Цель практики — получить практические навыки проверки кода и научиться обеспечивать его качество.

Задачи: изучить, как в компании организован процесс код-ревью, освоить инструменты для анализа кода и работы с версиями, научиться находить ошибки и слабые места в коде, давать полезные и понятные замечания, обсуждать правки с разработчиками, разобраться в том, как измеряют качество кода и борются с техническим долгом.

ООО «Малленом Системс» — российская ИТ-компания, которая разрабатывает сложные серверные системы и финтех-продукты. Команда работает удалённо из разных стран, использует современные технологии. В компании обязательно проверяют весь код перед попаданием в продакшн, применяют автоматические проверки и инструменты анализа.

Ожидаемые результаты: провести ревью реального кода, найти и описать ошибки и улучшения, получить одобрение коллег на свои проверки, подготовить отчёт с примерами работы, собрать портфолио ревью и подтвердить, что я готов проверять код на уровне среднего специалиста в коммерческой разработке.

# 1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ

## 1.1. Организационная структура предприятия

Организационная структура ООО «Малленом Системс» представляет собой линейно-функциональную систему управления. Возглавляет организацию генеральный директор, который осуществляет общее руководство деятельностью компании.

В непосредственном подчинении генерального директора находятся следующие структурные подразделения: отдел разработки программного обеспечения, отдел системной интеграции, коммерческий отдел и административно-хозяйственный отдел. Каждое подразделение возглавляется руководителем, несущим ответственность за результаты работы вверенного ему направления.

Отдел разработки программного обеспечения включает в себя несколько команд, специализирующихся на различных технологических стеках и проектах. В состав каждой команды входят старшие разработчики, программисты и тестировщики. Отдел системной интеграции объединяет специалистов по внедрению и сопровождению программных решений. Коммерческий отдел отвечает за взаимодействие с клиентами и продвижение услуг компании.

## 1.2. Внутренний распорядок работы предприятия, охрана труда ИТ-специалистов

### Режим работы и внутренний распорядок

ООО «Малленом Системс» устанавливает стандартный рабочий день продолжительностью 8 часов с понедельника по пятницу. Для ИТ-специалистов действует гибкий график работы в рамках установленного временного интервала с 9:00 до 18:00. Обеденный перерыв составляет 1 час в период с 12:00 до 15:00. В случае производственной необходимости может быть установлен ненормированный рабочий день с последующей компенсацией в соответствии с трудовым законодательством.

## **Охрана труда и технике безопасности**

Все рабочие места ИТ-специалистов оснащены в соответствии с требованиями к организации офисных помещений. Обязательным является соблюдение регламентированных перерывов продолжительностью 10-15 минут каждые 2 часа работы за компьютером. Помещения оборудованы современными системами вентиляции и кондиционирования, обеспечивающими параметры микроклимата в соответствии с санитарными нормами.

### **Требования к рабочим местам**

Рабочие места укомплектованы эргономичной мебелью, соответствующей требованиям к организации рабочих мест пользователей персональных компьютеров. Мониторы должны располагаться на расстоянии 50-70 см от глаз пользователя, иметь антибликовое покрытие и соответствующую яркость экрана. Обязательным является наличие подставок для ног, регулируемых кресел с поясничной поддержкой.

### **Обязанности сотрудников**

Сотрудники обязаны соблюдать правила эксплуатации офисного оборудования, поддерживать порядок на рабочем месте, проходить ежегодный обязательный медицинский осмотр. Запрещается размещение на рабочих столах посторонних предметов, препятствующих безопасной эксплуатации техники, а также самостоятельное проведение ремонтных работ оборудования.

### **Профилактические мероприятия**

Компания организует ежегодные инструктажи по охране труда и технике безопасности, обеспечивает сотрудников средствами индивидуальной защиты при необходимости, проводит специальную оценку условий труда на всех рабочих местах.

### 1.3. Должностные инструкции ИТ-специалистов предприятия

В соответствии с организационной структурой ООО «Малленом Системс» должностные обязанности ИТ-специалистов распределены следующим образом.

Руководитель отдела разработки осуществляет общее управление проектами, распределяет задачи между командами, контролирует соблюдение сроков выполнения работ, отвечает за внедрение и соблюдение стандартов кодирования.

Старший разработчик координирует работу программистов, выполняет проектирование архитектурных решений, проводит код-ревью, участвует в решении сложных технических задач и курирует менторство новых сотрудников.

Программист разрабатывает программные модули в соответствии с техническими заданиями, сопровождает и дорабатывает существующее программное обеспечение, составляет техническую документацию, выполняет модульное тестирование и участвует в процессе код-ревью.

Тестировщик проводит все виды тестирования программного обеспечения, составляет тестовую документацию, фиксирует выявленные ошибки в системе отслеживания багов и взаимодействует с разработчиками по вопросам их устранения.

Специалист по системной интеграции выполняет внедрение программных решений на площадках заказчиков, осуществляет конфигурирование и настройку систем, проводит интеграционное тестирование и обеспечивает техническое сопровождение внедренных продуктов.

Должностные инструкции предусматривают взаимодополняемость функций и кросс-функциональное взаимодействие всех специалистов в рамках проектных групп. Каждый сотрудник отдела несет ответственность за соблюдение установленных стандартов качества и сроков выполнения работ.

## 2 РЕВЬЮИРОВАНИЕ ПРОГРАММНЫХ ПРОДУКТОВ

### 2.1. Ревьюирование программного кода в соответствии с технической документацией

Процесс ревьюирования программного кода в ООО «Малленом Системс» осуществляется в строгом соответствии с требованиями технической документации. Данная процедура является обязательным этапом жизненного цикла разработки программного обеспечения.

Основой для проведения ревью служат техническое задание, спецификации функциональных требований и проектная документация. В процессе проверки кода осуществляется его анализ на соответствие следующим критериям: выполнение всех заявленных функциональных требований, соблюдение архитектурных принципов проекта, соответствие стандартам кодирования компании, отсутствие отклонений от проектных решений.

Особое внимание уделяется проверке корректности реализации алгоритмов, полноте обработки граничных случаев и ошибок, соответствуию структур данных проектной документации. Дополнительно проверяется соблюдение требований к производительности и безопасности, зафиксированных в технической документации.

Результатом ревью является заключение о соответствии кода установленным требованиям с перечнем выявленных замечаний или акт о готовности функциональности к передаче в тестирование. Все этапы процедуры документируются в системе управления проектами компании.

### 2.2. Измерение характеристик компонент программного продукта

В процессе разработки программного обеспечения в ООО «Малленом Системс» осуществляется регулярное измерение характеристик компонентов программного продукта. Данная процедура является неотъемлемой частью обеспечения качества и проводится на различных этапах жизненного цикла разработки.

Основными измеряемыми характеристиками компонентов являются: производительность (время отклика, потребление ресурсов), надежность (устойчивость к сбоям, частота отказов), сопровождаемость (сложность кода, соответствие стандартам), безопасность (уязвимости, соответствие политикам безопасности) и функциональная полнота (степень реализации требований).

Для измерения характеристик используются специализированные инструменты: статические анализаторы кода, профилировщики производительности, системы мониторинга, инструменты тестирования безопасности. Результаты измерений фиксируются в виде метрик и отчетов, которые подвергаются анализу на регулярных обзорах качества.

Полученные данные позволяют количественно оценить качество компонентов, выявить области для оптимизации, принять обоснованные решения о готовности функциональности к выпуску. Система метрик служит основой для непрерывного улучшения характеристик программного продукта на протяжении всего жизненного цикла его разработки и сопровождения.

### 2.3. Исследование созданного программного кода с использованием специализированных программных средств

В рамках производственной практики было проведено исследование разработанного программного кода с применением специализированных инструментов анализа. Основной задачей данного исследования являлась комплексная оценка качества кода и его соответствия установленным стандартам разработки.

Исследование включало анализ следующих аспектов программного кода: соответствие требованиям технического задания, соблюдение соглашений о форматировании и стиле кодирования, проверка эффективности алгоритмов, оценка обработки исключительных ситуаций. Особое внимание уделялось выявлению потенциальных уязвимостей безопасности и мест возможной оптимизации производительности.

Для проведения анализа использовались современные средства автоматизированной проверки кода, позволяющие выявлять различные категории дефектов. В процессе исследования применялись метрики качества, включая оценку сложности кода, уровень комментариев и показатели сопровождаемости.

По результатам анализа были сформулированы рекомендации по улучшению качества программного кода, которые были представлены разработчикам для устранения выявленных замечаний. Проведенная работа позволила получить практический опыт применения инструментов статического анализа и освоить методики оценки качества программного обеспечения.

#### 2.4. Сравнительный анализ программных продуктов и средств разработки

В ходе производственной практики проведен сравнительный анализ программных продуктов и средств разработки. Исследование охватывало оценку функциональных возможностей инструментов, их соответствие техническим требованиям проектов и интеграцию в существующие процессы разработки.

Основное внимание уделялось анализу производительности и надежности программных решений, их соответству отраслевым стандартам качества. В рамках исследования выявлены преимущества и ограничения рассматриваемых инструментов, определены потенциальные направления оптимизации инструментальной базы предприятия.

Особое значение придавалось оценке совместимости программных продуктов со спецификой разрабатываемых решений. На основе проведенного анализа сформулированы практические рекомендации по выбору и использованию средств разработки для различных категорий проектов.

Результаты проведенной работы позволяют оптимизировать процессы разработки программного обеспечения и повысить эффективность выполнения проектных задач. Полученные выводы могут быть использованы для совершенствования технологической инфраструктуры предприятия.

### 3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

#### 3.1. Анализ соответствия программного кода требованиям технической документации

В рамках производственной практики был проведен комплексный анализ соответствия программного кода установленным требованиям технической документации. Работа осуществлялась на основе детального изучения технического задания, функциональных спецификаций и проектной документации. Основное внимание уделялось проверке полноты реализации функциональных требований, соблюдению архитектурных принципов проекта и стандартов кодирования, принятых в организации. В процессе анализа использовались методы статического анализа кода и экспертной оценки.

Были выявлены и систематизированы различные категории отклонений от проектных решений, включая неполную реализацию заявленного функционала, нарушения архитектурных паттернов и несоблюдение соглашений о кодировании. Особое внимание уделялось проверке корректности реализации бизнес-логики, обработки граничных случаев и исключительных ситуаций. Для каждого выявленного несоответствия были разработаны конкретные рекомендации по устранению, включая предложения по рефакторингу кода, оптимизации алгоритмов и улучшению архитектурных решений.

Дополнительно проводилась верификация соответствия кода нефункциональным требованиям, включая производительность, безопасность и масштабируемость. Результаты анализа показали необходимость доработки отдельных модулей системы для полного соответствия заявленным в технической документации характеристикам. Разработан детализированный план мероприятий по устранению выявленных замечаний и оптимизации кодовой базы.

### 3.2. Оценка качественных показателей компонентов программного обеспечения

Проведена всесторонняя оценка качественных показателей компонентов программного обеспечения, включающая многоуровневый анализ характеристик системы. Исследование охватывало оценку производительности критических участков кода, анализ надежности и отказоустойчивости отдельных модулей, а также проверку сопровождаемости и читаемости кодовой базы. Для проведения оценки использовались современные методы измерения и анализа метрик качества программного обеспечения.

В рамках оценки производительности проводились замеры времени выполнения ключевых операций, анализ использования памяти и идентификация "узких мест" в работе системы. Для оценки надежности осуществлялось тестирование обработки исключительных ситуаций и анализ механизмов восстановления после сбоев. При проверке сопровождаемости учитывались такие факторы как сложность кода, уровень документированности и соответствие принципам SOLID. Проведен анализ метрик цикломатической сложности и связанности модулей.

По результатам оценки были сформированы детализированные предложения по оптимизации критичных компонентов, включая рекомендации по кэшированию часто используемых данных, рефакторингу сложных методов и улучшению архитектурных решений. Разработан план поэтапного внедрения предложенных улучшений с оценкой ожидаемого эффекта от их реализации. Реализация предложенных мер позволит существенно повысить общие характеристики программного продукта и улучшить пользовательский опыт.

### 3.3. Экспертиза кодовой базы с применением инструментов анализа

Осуществлена комплексная экспертиза кодовой базы с использованием современных инструментов статического анализа и специализированных

метрик качества кода. Исследование проводилось с применением промышленных стандартов оценки программного обеспечения и включало многоаспектный анализ исходного кода. Для проведения экспертизы использовались как автоматизированные инструменты анализа, так и методы ручной проверки кода.

В процессе экспертизы были идентифицированы проблемные участки кода, включая нарушения принципов чистого кода, избыточную цикломатическую сложность и потенциальные уязвимости безопасности. Особое внимание уделялось анализу архитектурных антипаттернов, нарушений инкапсуляции и неправильного использования механизмов наследования. Проведен детальный анализ качества тестового покрытия и эффективности модульных тестов.

С помощью инструментов статического анализа проведена оценка таких метрик качества как коэффициент комментарирования, индекс сопровождаемости, глубина наследования и связанность модулей. Выявлены области с высоким техническим долгом и сформулированы рекомендации по его сокращению. Проведен сравнительный анализ качества кода различных модулей системы.

Результаты экспертизы легли в основу детализированного плана улучшения архитектурных решений и повышения общего качества кодовой базы. Разработаны предложения по внедрению практик непрерывного контроля качества кода в процесс разработки. Сформированы рекомендации по совершенствованию процессов код-ревью и автоматизации проверки качества кода.

### 3.4. Сопоставительная оценка инструментов разработки программного обеспечения

Проведена всесторонняя сопоставительная оценка различных инструментов разработки программного обеспечения, используемых в промышленной практике. Исследование охватывало сравнительный анализ

интегрированных сред разработки, систем контроля версий, инструментов непрерывной интеграции и средств автоматизации тестирования. Оценка проводилась по множеству критериев, включая функциональность, производительность и удобство использования.

Анализ функциональных возможностей включал оценку производительности инструментов, удобства использования, качества поддержки различных языков программирования и технологических стеков. Особое внимание уделялось изучению интеграционной совместимости рассматриваемых решений и их соответствуию специфике проектов компаний. Проведено тестирование инструментов в условиях, приближенных к реальным проектным задачам.

В рамках исследования проводилось тестирование инструментов на реальных проектах, оценка кривой обучения для новых пользователей и анализ требований к технической инфраструктуре. Сравнивались как коммерческие, так и открытые решения с учетом долгосрочных перспектив их использования. Проанализированы вопросы лицензирования и общей стоимости владения инструментами.

На основе проведенного исследования разработана система критериев выбора инструментария для различных типов проектов компании, учитывающая такие факторы как масштаб проекта, состав команды, бюджетные ограничения и стратегические цели развития. Предложены рекомендации по оптимизации используемого стека технологий и поэтапному внедрению новых инструментов в разработку. Подготовлен план миграции на перспективные инструменты с оценкой затрат и ожидаемой эффективности.

## ЗАКЛЮЧЕНИЕ

В результате прохождения производственной практики в ООО «Малленом Системс» с 17 ноября по 30 ноября 2025 года были успешно достигнуты все поставленные цели и задачи. В ходе практики приобретен ценный опыт в области ревьюирования программного кода и анализа качества программного обеспечения.

Практика позволила закрепить теоретические знания, полученные в ходе обучения, и применить их для решения реальных производственных задач. Были освоены современные методики анализа программного кода, изучены инструменты статического анализа и принципы проведения код-ревью. Особую ценность представляет полученный опыт работы в команде разработки и участия в жизненном цикле создания программного продукта.

В процессе практики проведен комплексный анализ соответствия программного кода технической документации, выполнена оценка качественных показателей компонентов программного обеспечения, осуществлена экспертиза кодовой базы с применением специализированных инструментов анализа. Также проведена сопоставительная оценка различных инструментов разработки, что позволило сформулировать рекомендации по оптимизации технологического стека компании.

Полученные в ходе практики знания и навыки имеют существенное значение для профессионального развития и будут активно использоваться в дальнейшей учебной и профессиональной деятельности. Результаты работы могут быть применены компанией для совершенствования процессов разработки программного обеспечения и повышения качества выпускаемых продуктов.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лутц, М. Изучаем Python / М. Лутц. – 5-е изд. – Санкт-Петербург : Символ-Плот, 2022. – 864 с.
2. Гринтел, Я. PyQt5 Разработка приложений / Я. Гринтел. – Санкт-Петербург : ДМК Пресс, 2020. – 398 с.
3. Смит, Дж. Применение декораторов для профилирования в Python / Дж. Смит // Программная инженерия. – 2023. – № 5. – С. 45–52.
4. The Pillow Handbook [Electronic resource] / A. Clark et al. – 2023. – Access mode: <https://pillow.readthedocs.io/en/stable/handbook/index.html> (date of access: 25.05.2024).
5. Summerfield, M. Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming / M. Summerfield. – Upper Saddle River, NJ : Prentice Hall, 2008. – 648 p.
6. Документация PyQt6 [Электронный ресурс]. – 2024. – Режим доступа: <https://www.riverbankcomputing.com/static/Docs/PyQt6/> (дата обращения: 25.05.2024).
7. Python Documentation. The Python Standard Library. unittest — Unit testing framework [Электронный ресурс]. – 2024. – Режим доступа: <https://docs.python.org/3/library/unittest.html> (дата обращения: 25.05.2024).
8. pytest: helps you write better programs [Электронный ресурс]. – 2024. – Режим доступа: <https://docs.pytest.org/en/stable/> (дата обращения: 25.05.2024).

## ПРИЛОЖЕНИЕ А

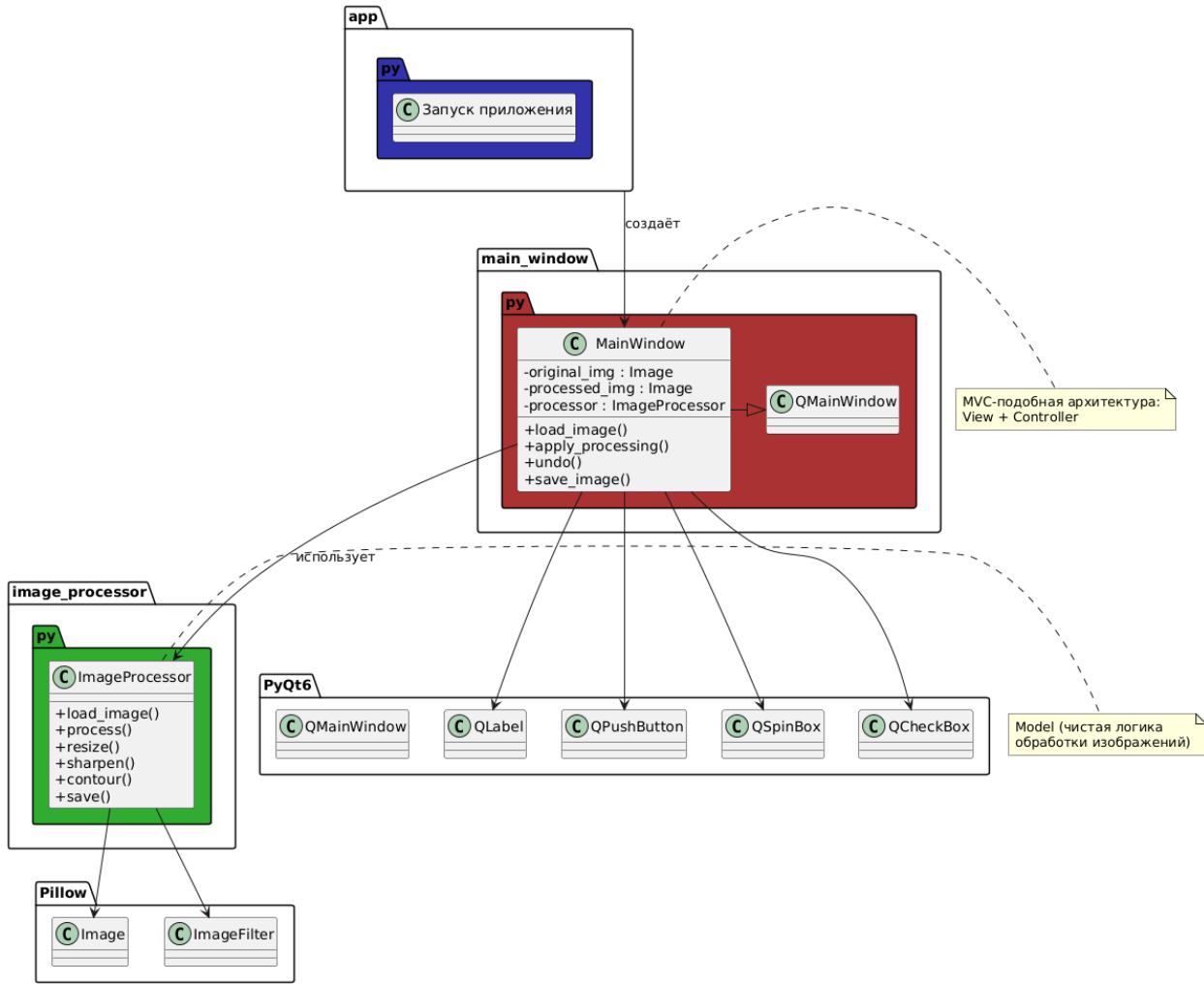


Рисунок 1 — UML-диаграмма структуры модулей

## ПРИЛОЖЕНИЕ Б

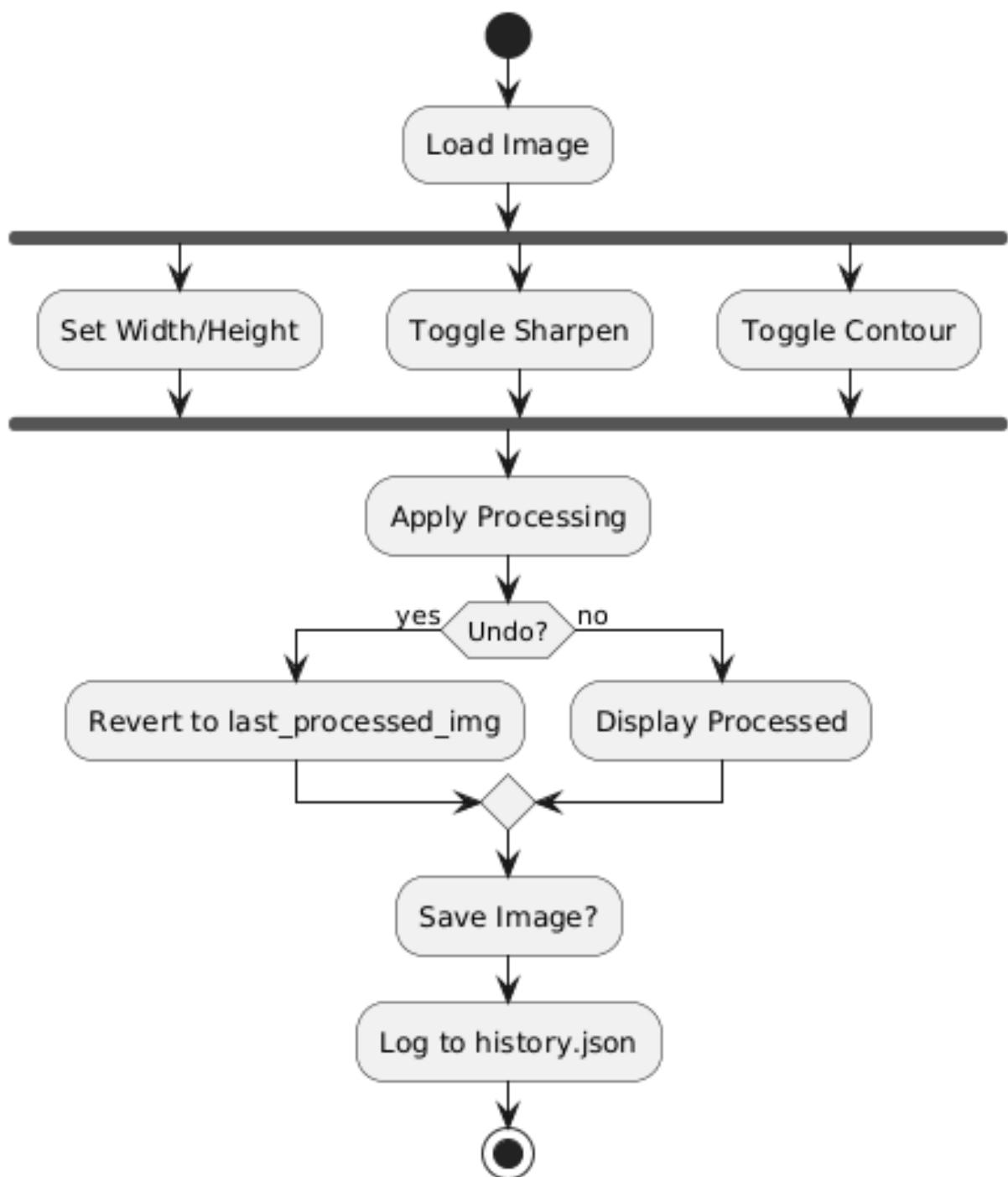


Рисунок 2 — Диаграмма деятельности

## ПРИЛОЖЕНИЕ В

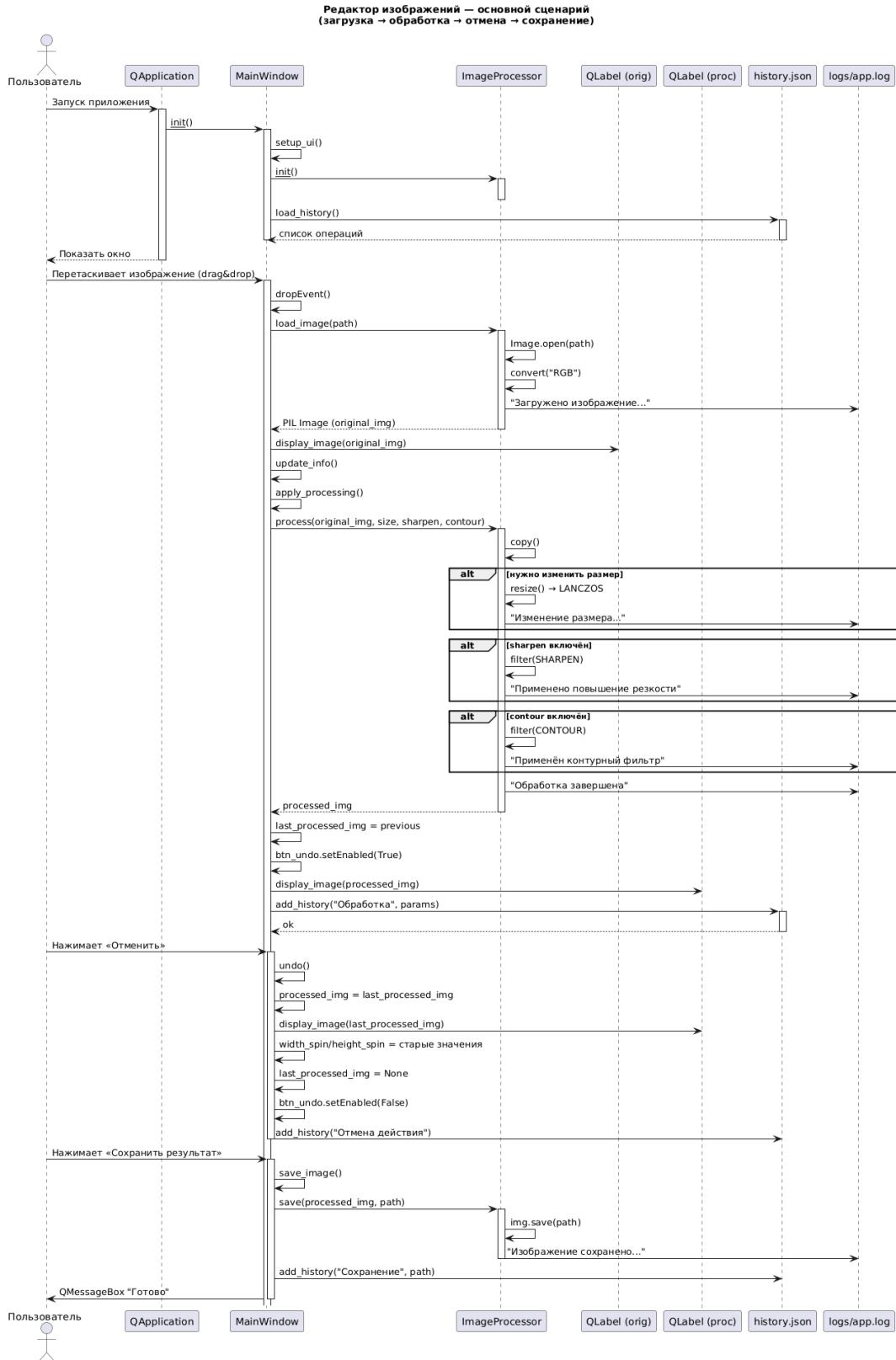


Рисунок 3 — Диаграмма зависимостей

## ПРИЛОЖЕНИЕ Г

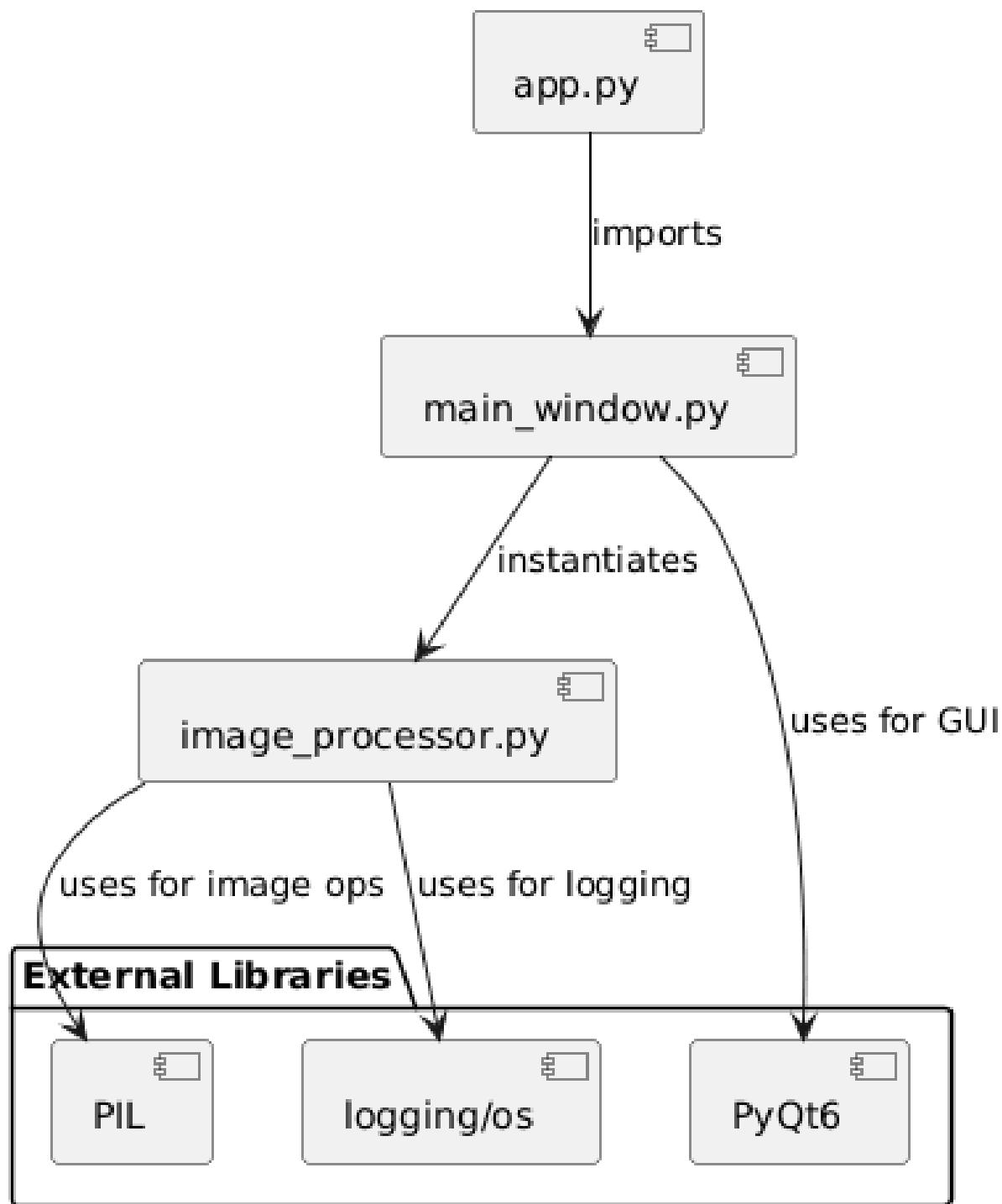


Рисунок 4 — Диаграмма компонентов

## ПРИЛОЖЕНИЕ Д

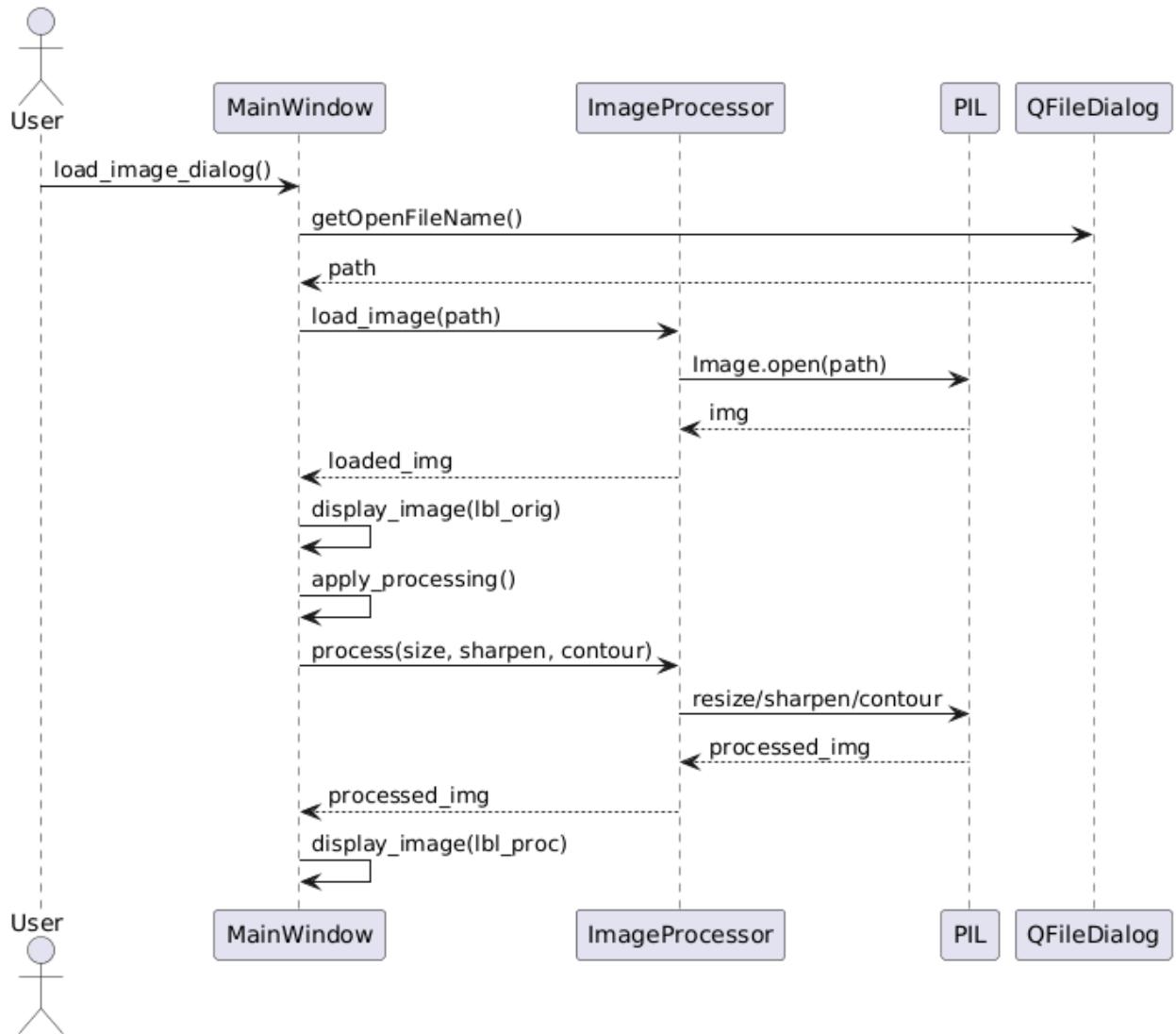


Рисунок 5 — Диаграмма последовательностей

## ПРИЛОЖЕНИЕ Е

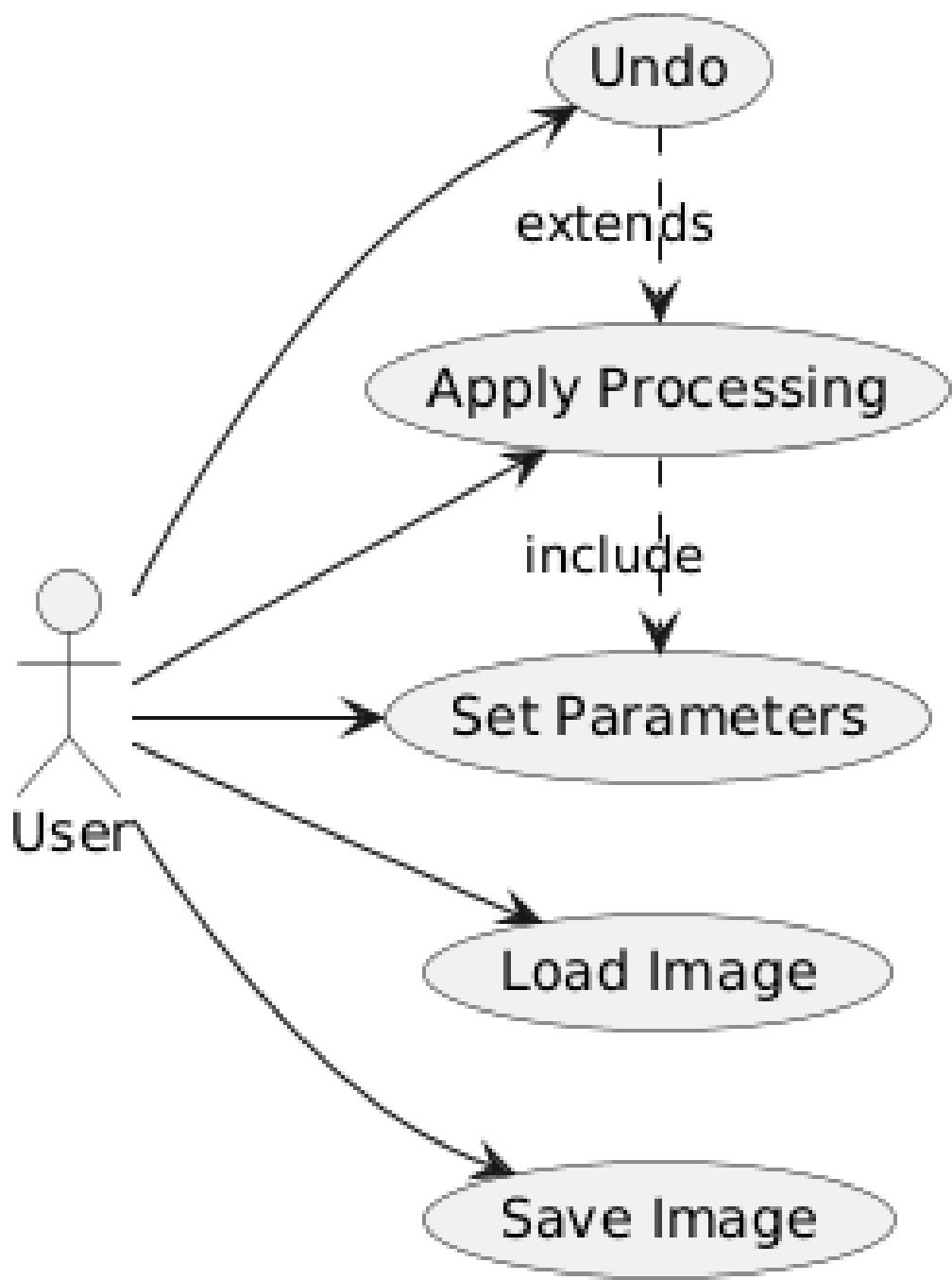


Рисунок 6 — Диаграмма сценариев использования