# 2 **Проектирование задачи**

# **2.1 Моделирование проекта**

Диаграммы вариантов использования описывают взаимоотношения и зависимости между группами вариантов использования и действующих лиц, участвующими в процессе.

Диаграммы вариантов использования предназначены для упрощения взаимодействия с будущими пользователями системы, с клиентами, и особенно пригодятся для определения необходимых характеристик системы. Другими словами, диаграммы вариантов использования говорят о том, что система должна делать, не указывая применяемые методы.

Диаграмма вариантов использования, представлена на рисунке 2.1.

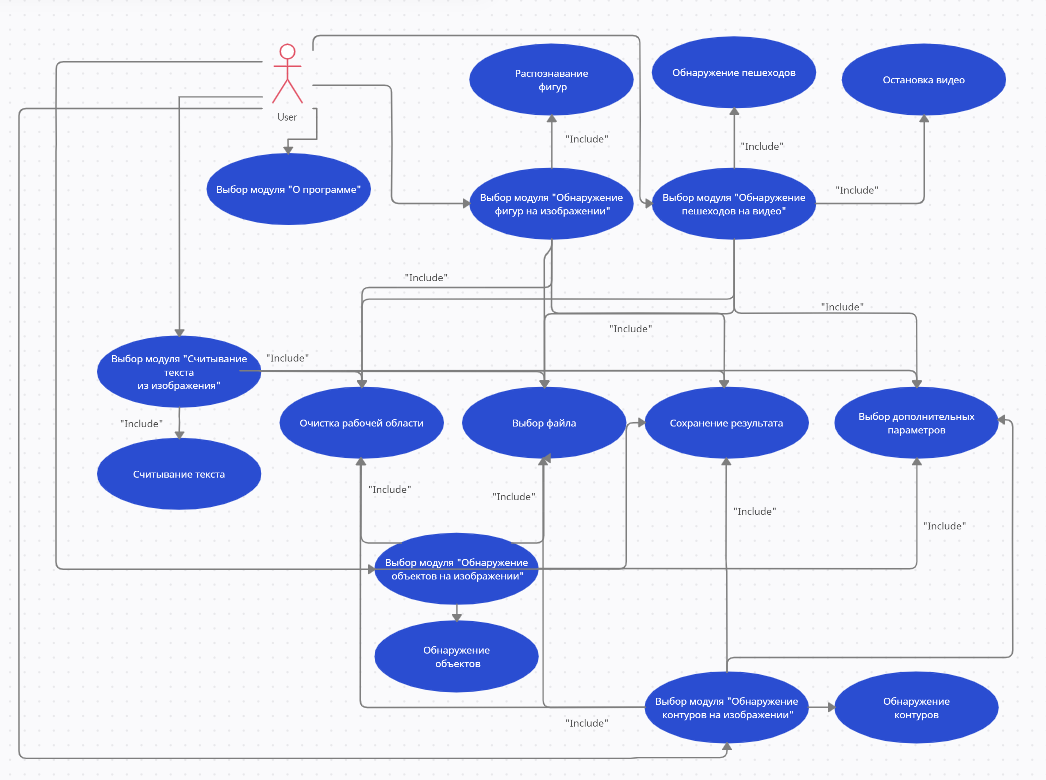


Рисунок 2.1 – Диаграмма вариантов использования

Источник: собственная разработка

В диаграмме вариантов использования, представленной на рисунке 2.1, отображен функционал курсового проекта.

Здесь присутствуют варианты использования, при выборе определенного модуля приложения становятся доступными еще несколько действий, такие как выбор файла, очистка рабочей области и так далее. Конечным результатом может являться изображение, текст, или видео.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

Деятельность может содержать входящие и/или исходящие дуги, показывающие потоки управления и потоки данных. Если поток соединяет две деятельности, он является потоком управления.

Для создания диаграммы деятельности используются следующие узлы:

узел управления (control node) – это абстрактный узел действия, которое координирует потоки действий;

начальный узел деятельности (или начальное состояние деятельности) (activity initial node) является узлом управления, в котором начинается поток (или потоки) при вызове данной деятельности извне;

конечный узел деятельности (или конечное состояние деятельности) (activity final node) является узлом управления, который останавливает (stop) все потоки данной диаграммы деятельности, на диаграмме может быть более одного конечного узла;

конечный узел потока (или конечное состояние потока) (flow final node) является узлом управления, который завершает данный поток, на другие потоки и деятельность данной диаграммы это не влияет;

Точка разделенияобеспечивает разделение одного потока на несколько параллельных потоков.

Точка слиянияобеспечивает синхронизацию нескольких параллельных потоков.

Диаграмма деятельности представлена на рисунке 2.2.

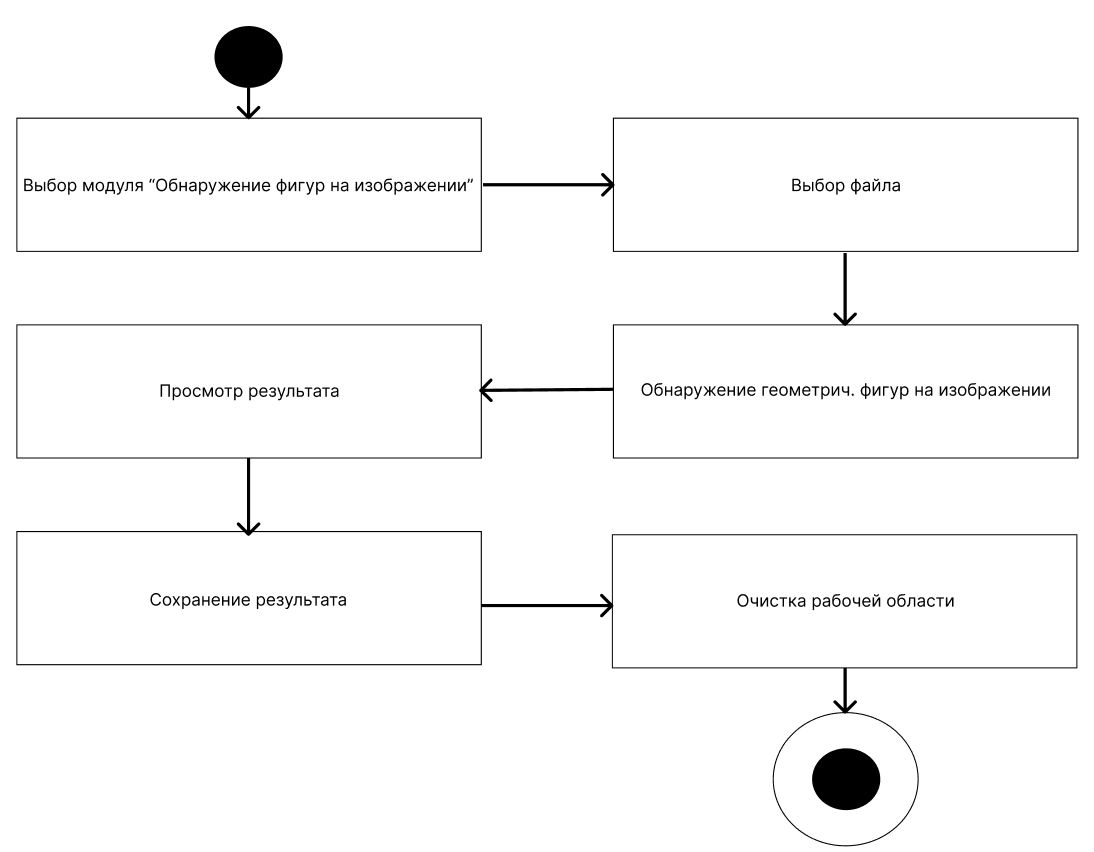


Рисунок 2.2 – Диаграмма деятельности модуля «Обнаружение фигур на изображении»

Источник: собственная разработка

В диаграмме, представленной на рисунке 2.2, размещается алгоритм работы модуля «Обнаружение фигур на изображении», сам модуль распознает геометрические фигуры на изображении c помощью компьютерного зрения, обводит результат и подписывает фигуры, имеется возможность сохранения результата в файл с расширением JPG или PNG.

Диаграмма последовательности ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) sequence diagram) - [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), на которой для некоторого набора объектов на единой временной оси показан жизненный цикл (создание-деятельность-уничтожение) и взаимодействие (отправка запросов и получение ответов). Используется в языке [UML](https://ru.wikipedia.org/wiki/UML).

Основными элементами диаграммы последовательности являются обозначения [объектов](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) (прямоугольники с названиями объектов), вертикальные «линии жизни» ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) lifeline), отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами.

Диаграмма последовательности представлена на рисунке 2.3.

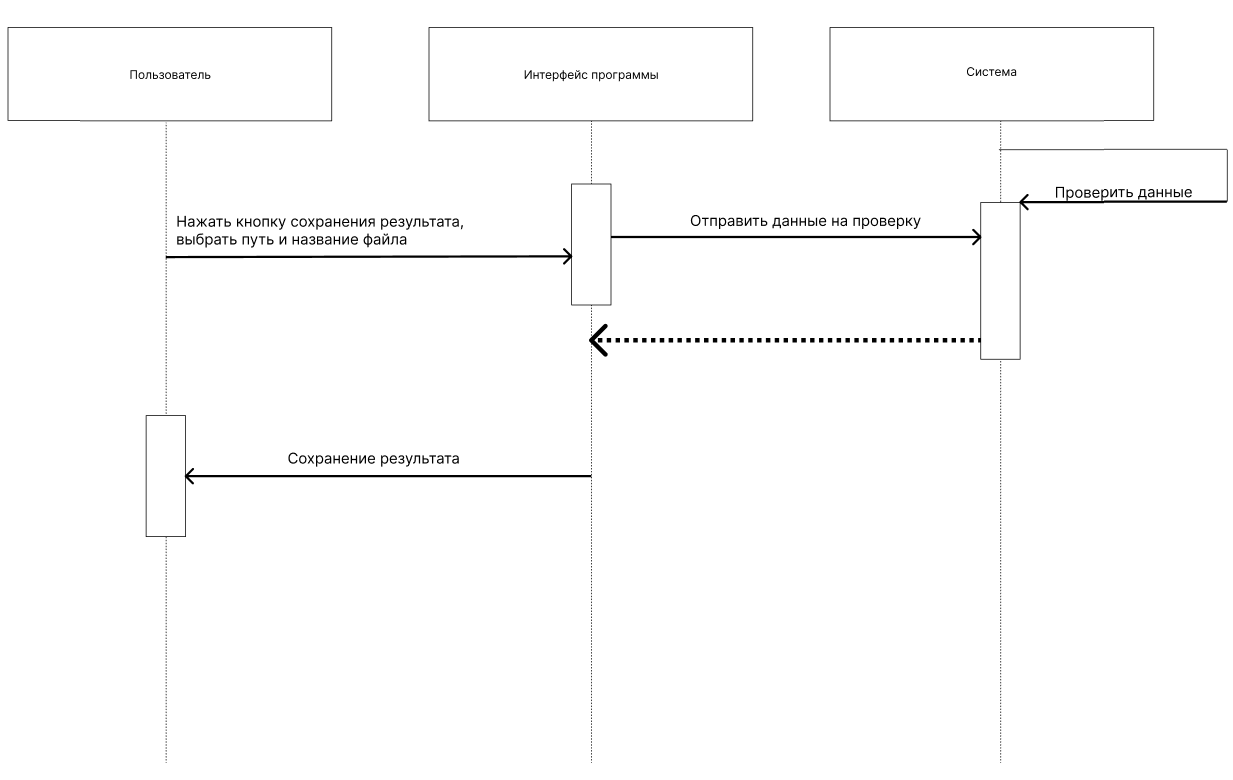


Рисунок 2.3 – Диаграмма последовательности (сохранение результата)

Источник: собственная разработка

На диаграмме последовательности, представленной на рисунке 2.3, отображена полная последовательность действий, исполняемая пользователем, для реализации той или иной функции. В данном случае при сохранении результата, у которого задан путь сохранения и имя файла интерфейс передает данные системе, затем данные проверяются.

Если проверка прошла успешно, то результат сохраняется в указанную директорию.

**2.2 Описание системы меню**

Система меню представлена в виде схемы в соответствии с рисунком 2.4.



Рисунок 2.4 – Схема системы меню

Источник: собственная разработка

Изначально открыта главная страница приложения, представленная на рисунке 2.5.

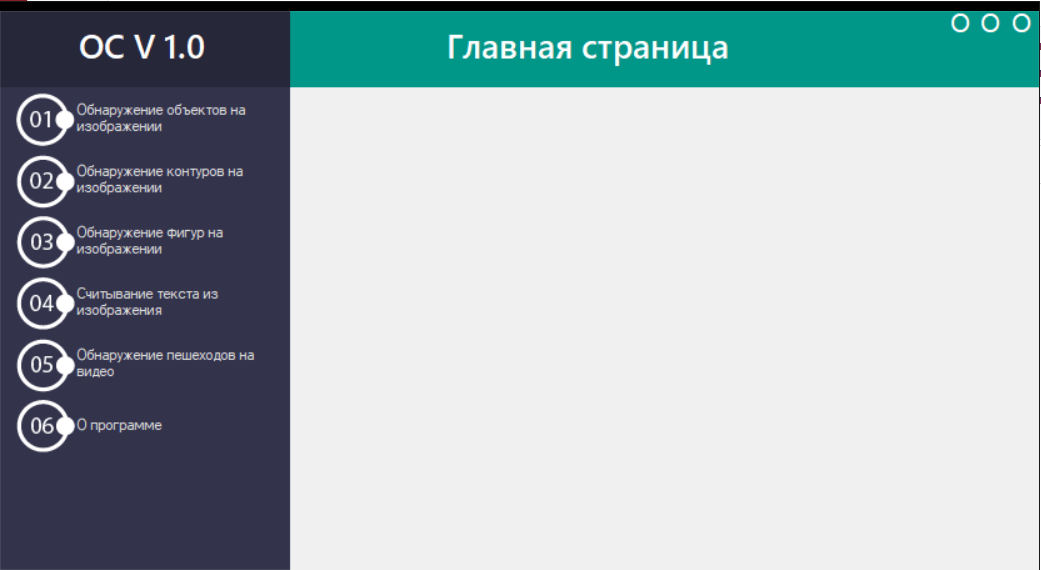


Рисунок 2.5 – Главная страница приложения

Источник: собственная разработка

Для открытия других модулей необходимо нажать на кнопки, расположенные на боковой панели приложения, при нажатии на кнопки отображаются формы со всеми элементами управления.

Результат открытия модуля «Обнаружение пешеходов на видео» представлен на рисунке 2.6

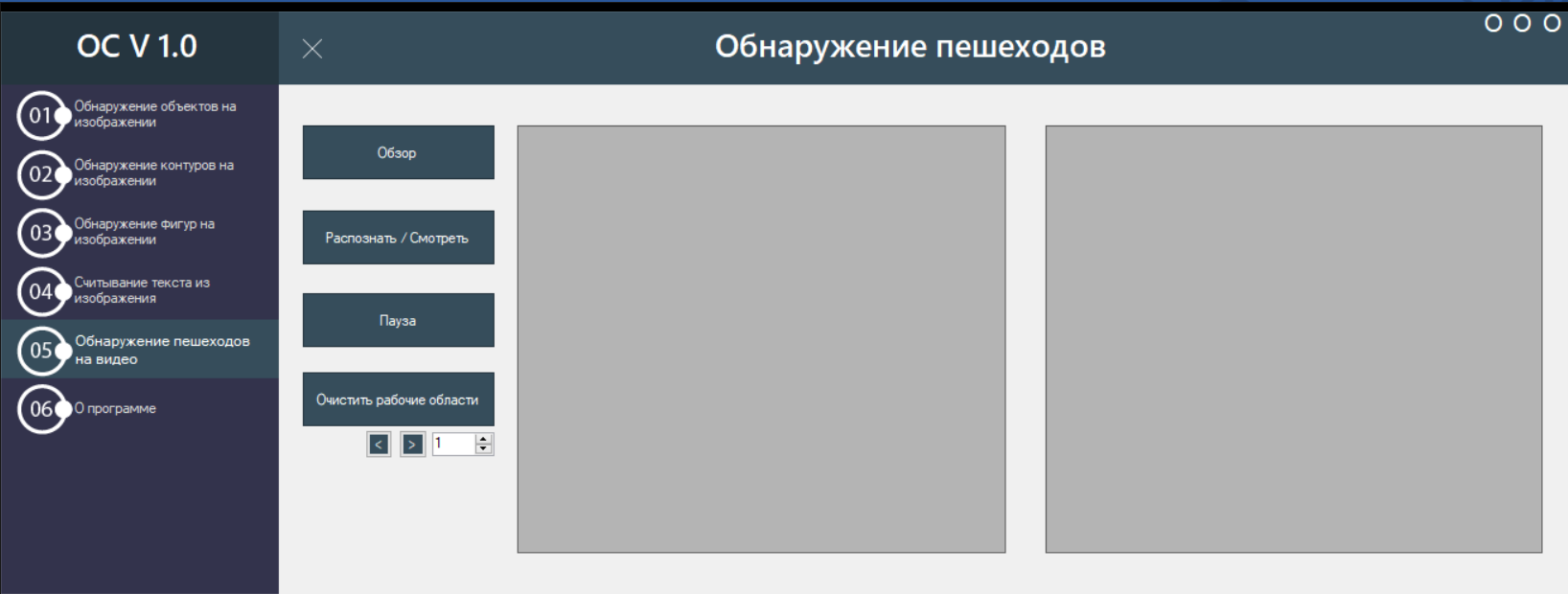


Рисунок 2.6 – модуль «Обнаружение пешеходов на видео»

Источник: собственная разработка

**2.3 Выбор и обоснование среды разработки**

.NET Framework – программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общеязыковая среда исполнения Common Language Runtime (CLR), которая подходит для различных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду. В настоящее время .NET Framework развивается в виде .NET.

Программа для .NET Framework, написанная на любом поддерживаемом языке программирования, сначала переводится компилятором в единый для .NET промежуточный байт-код Common Intermediate Language (CIL) (ранее назывался Microsoft Intermediate Language, MSIL). В терминах .NET получается сборка, англ. assembly. Затем код либо исполняется виртуальной машиной Common Language Runtime (CLR), либо транслируется утилитой NGen.exe в исполняемый код для конкретного целевого процессора. Использование виртуальной машины предпочтительно, так как избавляет разработчиков от необходимости заботиться об особенностях аппаратной части. В случае использования виртуальной машины CLR встроенный в неё JIT-компилятор «на лету» (just in time) преобразует промежуточный байт-код в машинные коды нужного процессора. Современная технология динамической компиляции позволяет достигнуть высокого уровня быстродействия. Виртуальная машина CLR также сама заботится о базовой безопасности, управлении памятью и системе исключений, избавляя разработчика от части работы.

Архитектура .NET Framework описана и опубликована в спецификации Common Language Infrastructure (CLI), разработанной Microsoft и утверждённой ISO и ECMA. В CLI описаны типы данных .NET, формат метаданных о структуре программы, система исполнения байт-кода и многое другое.

Объектные классы .NET, доступные для всех поддерживаемых языков программирования, содержатся в библиотеке Framework Class Library (FCL). В FCL входят классы Windows Forms, ADO.NET, ASP.NET, Language Integrated Query, Windows Presentation Foundation, Windows Communication Foundation и другие. Ядро FCL называется Base Class Library (BCL).