**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ**

**ПО ДИСЦИПЛИНЕ «ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНЫХ СИСТЕМ»**

**НА ТЕМУ: *«ВСПОМОГАТЕЛЬНАЯ ПОДСИСТЕМА ИДЕНТИФИКАЦИИ АВТОМОБИЛЯ ПО НОМЕРНОМУ ЗНАКУ НА ПРОПУСКНОМ ПУНКТЕ»***

Студент ИУ6-53 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мамадаев И.М.**

(Подпись, дата)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фетисов М.В.**

(Подпись, дата)

*2018 г.*

**Задание(здесь будет подписанный лист)**

**РЕФЕРАТ**

Расчётно-пояснительная записка 31 страниц, 24 рисунка, 4 таблицы, 5 источников, 3 приложения.

Объектом разработки является прототип программы, предназначенной для автоматического распознания автомобильного номера по его отсканированному (сфотографированному) изображению и авторизации пользователей, имеющих доступ к ПИАНЗ.

Цель работы – создание описанного выше приложения, обеспечивающего:

* Регистрацию пользователей;
* Авторизацию в программе;
* Выбор изображения с автомобильным номером;
* Получение результатов в виде распознанного номера и времени распознания.

В процессе работы проводилось исследование предметной области и проектирование необходимого приложения, в результате чего было создано приложение.

Поставленная цель достигается средствами языка программирования C# и библиотекой компьютерного зрения для распознания объектов с изображений «emguCV». В качестве среды разработки используется «Visual Studio 2017». Для контроля версий используется интернет-ресурс «GitHub». Хранение данных пользователей было организовано с помощью средств стандартной СУБД для «Visual Studio» - Microsoft SQL Server.

Приложение позволит упростить работу сотрудника охраны, находящегося на контрольно-пропускном пункте со шлагбаумом. ПИАНЗ должна ускорить и повысить надежность процесса пропуска автомобилей на частную территорию, облегчить работу сотрудника охраны.

Ключевые слова: приложение, авторизация, распознание, автомобили, номера, охрана, C#, emguCV.

Содержание

[ВВЕДЕНИЕ 5](#_Toc532210562)

[1 Анализ технических требований и уточнение спецификаций 6](#_Toc532210563)

[1.1 Анализ задания, выбор технологии, языка и среды разработки 6](#_Toc532210564)

[1.2 Модель жизненного цикла программы 6](#_Toc532210565)

[1.3 Разработка терминологии проекта 8](#_Toc532210566)

[1.4 Разработка диаграммы вариантов использования 8](#_Toc532210567)

[1.5 Объектная декомпозиция предметной области 12](#_Toc532210568)

[1.6 Разработка базы данных 14](#_Toc532210569)

[1.7 Разработка структурной схемы 14](#_Toc532210570)

[2 Проектирование структуры и компонентов программного продукта 16](#_Toc532210571)

[2.1 Основания выбора вида интерфейса 16](#_Toc532210572)

[2.2 Разработка графа состояний интерфейса 16](#_Toc532210573)

[2.3 Разработка методов основных форм 19](#_Toc532210574)

[2.4 Разработка форм интерфейса 20](#_Toc532210575)

[3 Тестирование приложения 28](#_Toc532210576)

[3.1 Тестирование граничных условий 28](#_Toc532210577)

[3.2 Тестирование методом эквивалентных разбиений 28](#_Toc532210578)

[3.3 Тестирование причинно-следственных связей 30](#_Toc532210579)

[ЗАКЛЮЧЕНИЕ 32](#_Toc532210580)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 33](#_Toc532210581)

[ПРИЛОЖЕНИЕ А. Техническое задание 34](#_Toc532210582)

[ПРИЛОЖЕНИЕ Б. Руководство пользователя 35](#_Toc532210583)

[ПРИЛОЖЕНИЕ В. Фрагменты исходного кода программы 36](#_Toc532210584)

# ВВЕДЕНИЕ

Цель данной работы – разработка «Подсистемы идентификации автомобиля по номерному знаку на пропускном пункте» с использованием основных методов анализа, проектирования, реализации и тестирования программных систем.

Для достижения поставленной цели выделяются следующие задачи:

–анализ предметной области и выбор технических средств реализации;

–анализ технологий, применяемых для распознавания текста с изображения;

–реализация пользовательского интерфейса;

–реализация авторизации и базы данных;

–реализация модуля распознавания автомобильного номера с изображения;

Разработка ПИАНЗ является актуальной, так как в настоящее время все профессии, требующие монотонной длительной деятельности человека, постепенно заменяются автоматизированными системами, позволяющими выполнять эту работу компьютерам. Человек в таких профессиях остается лишь для контроля работы компьютера и его замены в случае отказа.

Основное назначение «Подсистемы идентификации автомобиля по номерному знаку на пропускном пункте» заключается в упрощении работы сотрудника охраны, находящегося на контрольно-пропускном пункте со шлагбаумом. Использование ПИАНЗ уменьшит необходимость нахождения сотрудника охраны на пропускном пункте и создаст перспективу полного отказа от сотрудников охраны на КПП.

«Подсистема идентификации автомобиля по номерному знаку на пропускном пункте» может быть легко модифицирована и улучшена, так как реализация интерфейсной и логической части находятся в разных модулях.

# Анализ технических требований и уточнение спецификаций

## Анализ задания, выбор технологии, языка и среды разработки

Согласно требованиям задания требуется разработать программный продукт, имеющий авторизацию, ограничивающую доступ пользователей, не имеющих доступа к ПИАНЗ, а также имеющий графический интерфейс пользователя, объединяющий в себе остальные несколько самостоятельных компонентов программы.

В связи с этим был выбран объектно-ориентированный подход, позволяющий использовать такие свойства, как наследование и инкапсуляция, с использованием которых легко реализовать интерфейс любой сложности.

При проектировании и реализации программного продукта будет использоваться нисходящий подход, в данном случае он является предпочтительным, так как в предметной области используется большое количество классов и их методов, на которых основан функционал всего приложения. Их взаимодействие будет легче отладить и тестировать при наличии визуального отображения интерфейса.

## Модель жизненного цикла программы

Для данной разработки была выбрана спиральная модель жизненного цикла (рисунок 1.2.1). В соответствии с данной схемой программное обеспечение создается не сразу, а итерационно с использованием метода прототипирования, базирующегося на создании прототипов.

Основным достоинством данной схемы является то, что, начиная с некоторой итерации, на которой обеспечена определенная функциональная полнота, продукт можно предоставлять пользователю, что позволяет сократить время до появления первых версий программного продукта и создаёт благоприятные условия для долгосрочной поддержки программного продукта.



Рисунок 1.2.1 – Спиральная модель жизненного цикла

Для реализации программного продукта был выбран язык C#, являющийся полноценным современным языком программирования, поддерживающим необходимые в данной разработке возможности. Разработка проводилась в среде «Visual Studio 2017». Использование быстрой библиотеки «emguCV» (computer vision), которая может работать без доступа в Интернет, позволило использовать возможности «компьютерного зрения», сохраняя при этом кроссплатформенность разрабатываемого продукта.

Хранение данных пользователей было реализовано с помощью стандартной для «Visual Studio» СУБД «Microsoft SQL Server».

## Разработка терминологии проекта

Для грамотной разработки и реализации проекта необходимо разработать терминологию. С помощью выбранных терминов будет описываться вся деятельность приложения во время его разработки, основной деятельности и поддержки.

Для реализации деятельности сотрудника охраны, достаточно ввести модель «Пользователь», имеющей доступ к основному функционалу ПИАНЗ.

Для корректной организации регистрации новых пользователей необходима модель «Администратор», которая помимо доступа к функционалу ПИАНЗ будет также иметь возможность подтверждать или отклонять новые регистрационные заявки.

На основании данных выводов был составлен глоссарий ПИАНЗ (Таблица 1.3.1).

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| Администратор | Сторона, организующая регистрацию новых пользователей, обеспечивающая подтверждение или отклонение заявок новых пользователей. |
| Заявка | Пользовательская заявка на регистрацию. |
| Пользователь | Сторона, получающая книги и несущая ответственность за их своевременное возвращение.  Синонимы: сотрудник КПП. |

Таблица 1.3.1 – Глоссарий программного продукта

## Разработка диаграммы вариантов использования

Разрабатываемая ПИАНЗ предусматриваем однопользовательскую архитектуру, но разграничивающую доступ пользователей с различными правами.

Предметная область предполагает процесс распознания автомобильного номера с изображения, при этом данный процесс включает в себя взаимодействие с файловой системой платформы и получение графического отображения выбранного для распознания изображения.

Доступ к процессу распознания ограничивается авторизацией, которая осуществляется одним из пунктов в главном меню. Также в меню находится возможность регистрации для неавторизованных пользователей. Для пользователя типа «Администратор» определены привилегированные права, позволяющие подтвердить или отклонить регистрацию пользователей из соответствующей вкладки в главном меню. Таким образом, выделяются два варианта использования:

-главное меню, предполагающее:

* регистрацию;
* авторизацию;
* подтверждение учетной записи;
* справку;
* выход.

-распознание автомобильного номера, предполагающее:

* выбор изображения для распознания;
* отображения выбранного изображения;
* отображение распознанного номера в виде текста;
* отображение времени, потраченного на процесс распознания.

Таким образом, отобразим эти варианты использования на соответствующей диаграмме на рисунке 1.4.1.

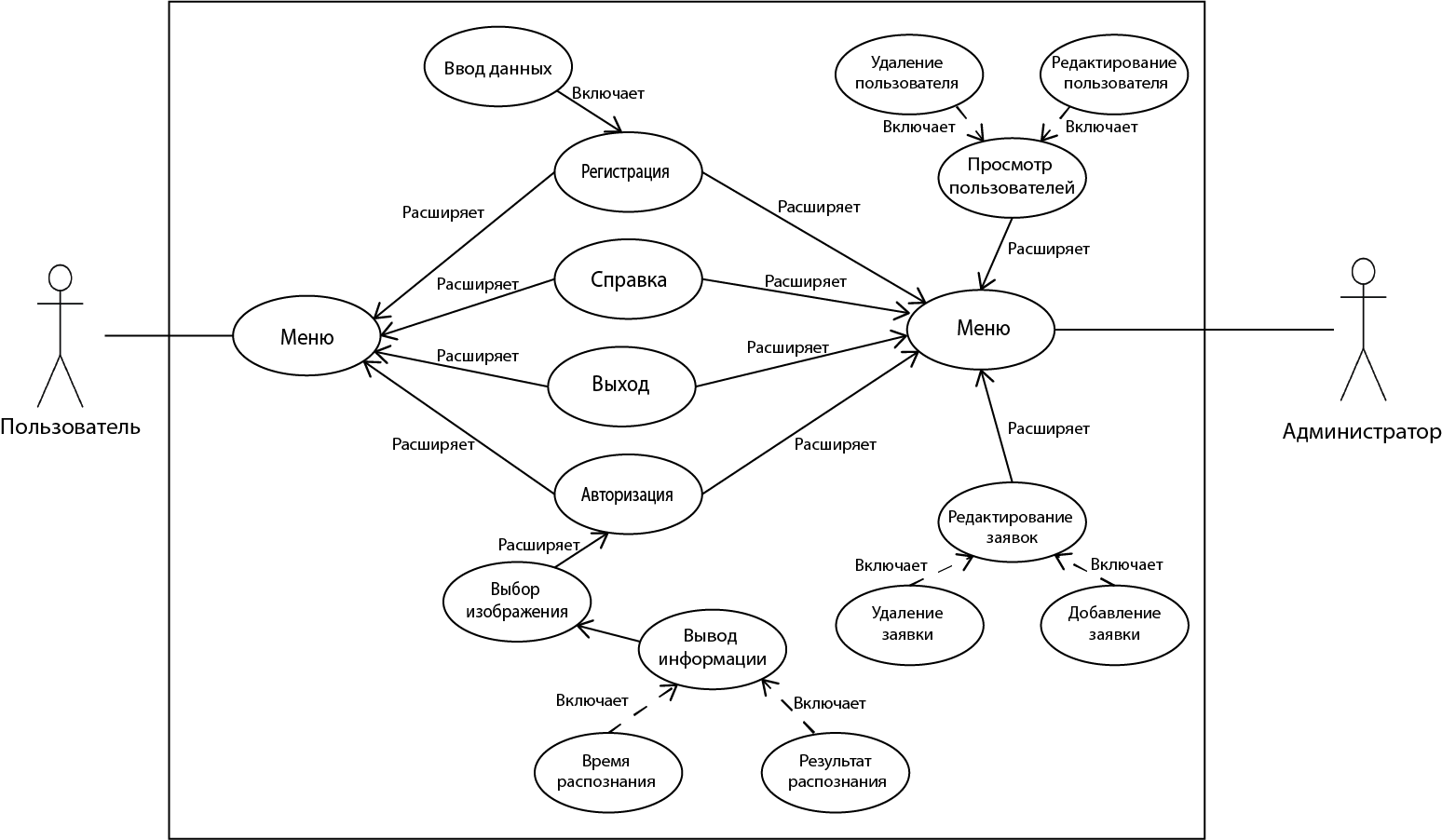


Рисунок 1.4.1 – Диаграмма вариантов использования

На диаграмме изображены семь основных вариантов использования – регистрация и авторизация пользователя, процесс распознания, просмотр пользователей, редактирование заявок, справка и выход.

Остальные варианты использования предполагают расширение основных вариантов. В их количество входят - ввод данных, выбор изображения, вывод времени распознания и результата распознания, удаление/добавление заявки и удаление пользователя.

## Объектная декомпозиция предметной области

Для реализации описанного приложения понадобится множество схожих между собой объектов, требуемых для реализации пользовательского интерфейса: кнопки, поля для ввода/вывода, поле для отображения изображения.

В связи с этим были составлены диаграммы классов, представленные на рисунках 1.5.1-1.5.2, которые также будут использованы в реализации ПИАНЗ.

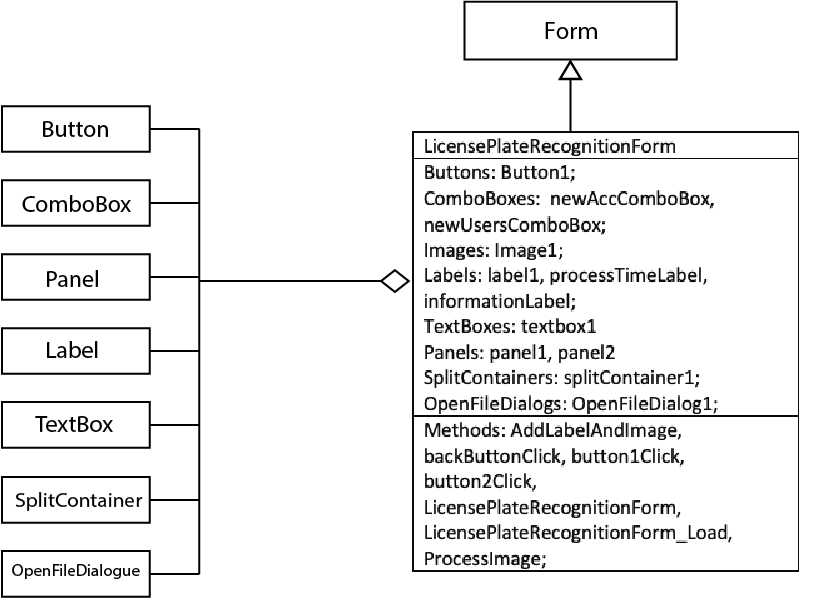


Рисунок 1.5.1 – Диаграмма классов формы «License Plate Recognition»

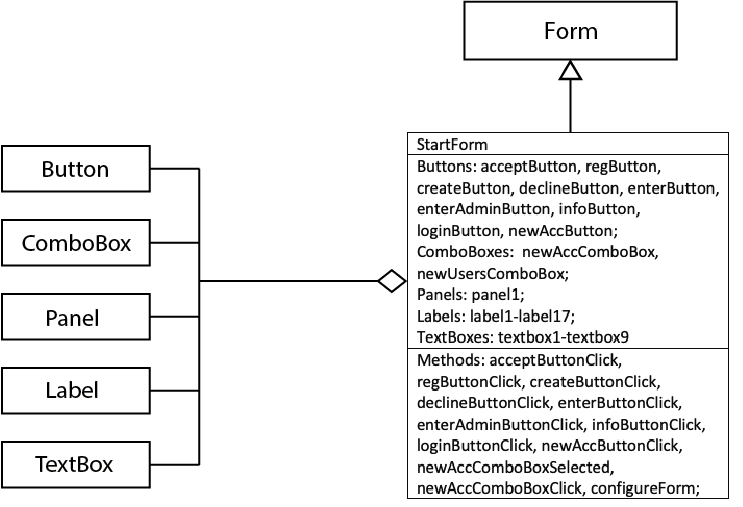


Рисунок 1.5.2 – Диаграмма классов формы «StartForm»

## Разработка базы данных

Для реализации авторизации необходимо где-то хранить данные пользователей, имеющий доступ к ПИАНЗ. В базе данных достаточно одной таблицы «Persons» (см рисунок 1.7.1), в которой будут следующие поля:

* id – уникальное идентификатор пользователя (первичный ключ);
* fullname – фамилия, имя, отчество пользователя;
* username – логин пользователя, использующийся для авторизации;
* password – пароль пользователя, использующийся для авторизации;
* email – адрес электронной почты.

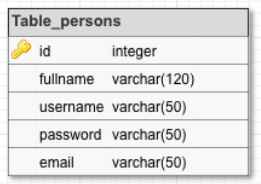


Рисунок 1.6.1 – Диаграмма базы данных

## Разработка структурной схемы

В проекте было выделено три области: область для хранения исходных изображений для распознания, область для хранения служебных файлов и изображений, а также область для хранения файлов-классов. Поскольку хранение всех файлов и изображений в одной директории ухудшает технологичность программного продукта, было решено разделить файлы по следующим папкам:

* Папка «\_pictures» - содержит изображения и файлы для реализации пользовательского интерфейса;
* Папка «\_numbers» - находится в папке «\_pictures» и содержит изображения автомобильных номеров, она открывается по умолчанию при выборе изображения в приложении;
* Папка «controllers» - содержит всей файлы, реализующие как логику программы (файлы с расширением «.cs»), так и её интерфейс (формы);
* Папка «tessdata» - является программно-сгенерированной с помощью библиотеки «emguCV» и содержит данные, используемые для распознания букв и цифр с изображения;

Структурная схема проекта приведена на рисунке 1.4.1

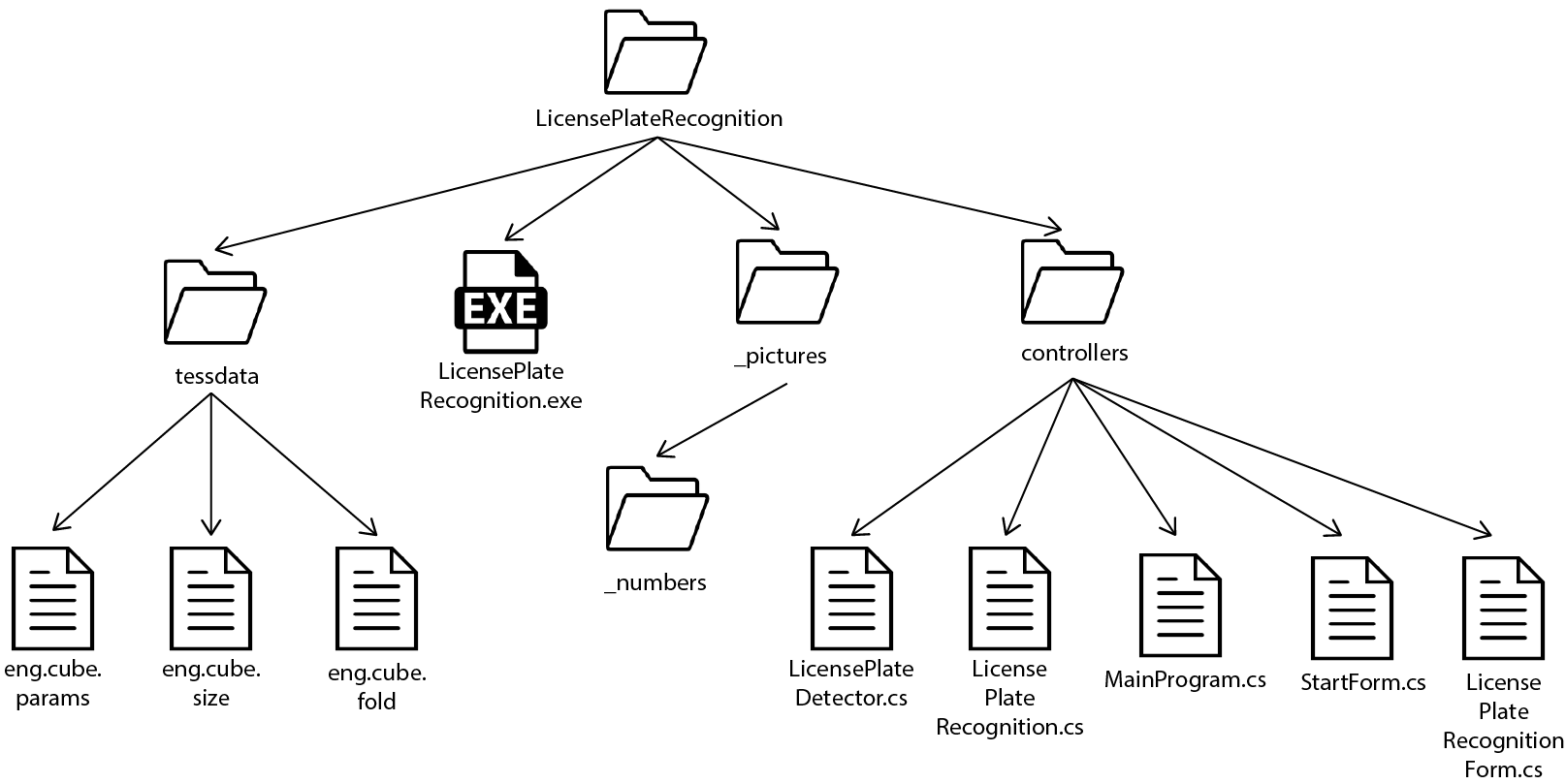


Рисунок 1.7.1 – Структурная схема проекта

# Проектирование структуры и компонентов программного продукта

## Основания выбора вида интерфейса

Интерфейс разрабатываемого программного продукта предназначен для узкого круга пользователей, имеющих доступ к ПИАНЗ, а так как в разрабатываемом приложении акцент делается на повторяющуюся операцию распознания номера с изображения, поэтому был выбран процедурно-ориентированный способ проектирования пользовательского интерфейса

В связи с тем, что при разработке программного продукта был выбран нисходящий способ проектирования, то первоочередной задачей ставится реализация пользовательского, а функционал компонентов временно замещается заглушками.

## Разработка графа состояний интерфейса

На основании предпроектных исследований и изучении работы ПИАНЗ было решено использовать интерфейс, соответствующий графу состояний интерфейса, представленному на рисунке 2.1.1.

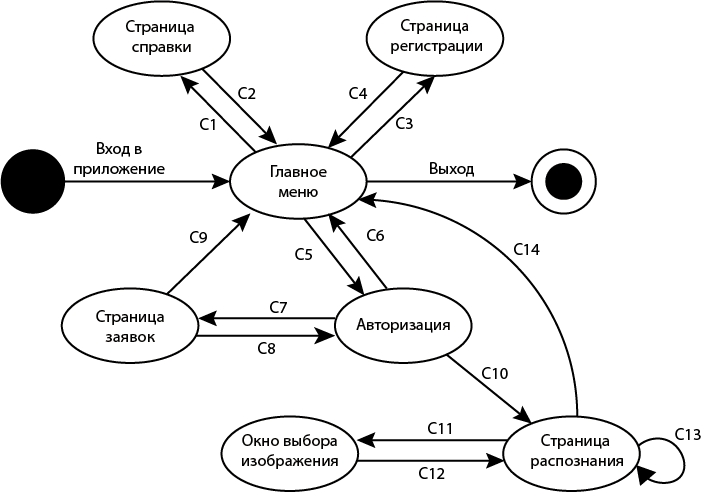


Рисунок 2.2.1 – Граф состояний интерфейса

Обозначения на графе состояний интерфейса:

С1 — нажатие кнопки «Справка»

С2 — возвращение в главное меню

С3 — нажатие кнопки «Регистрация»

С4 — Заполнение формы и возвращение в главное меню

С5 — нажатие кнопки «Авторизация»

С6 — Неуспешная авторизация и возвращение в главное меню

С7 — нажатие кнопки «Заявки» после авторизации

С8 — возвращение со страницы «Заявки» по окончанию редактирования

С9 — нажатие кнопки «Назад»

С10 — переход после успешной авторизации пользователя на страницу распознания

С11 — нажатие на «Загрузить»

С12 — нажатие кнопки «Выбрать» или «Отмена» или иконки «крестик»

С13 — повторный переход на страницу распознания после очередной итерации распознания изображения

С14 — нажатие иконки «крестик»

Детализуем диалог «Страница распознания» (рисунок 2.2.2), так как именно он является ключевым диалогом в данном приложении. Данный диалог является управляемым системой, т.к. пользователю предоставляется всего 3 варианта действия: выбор изображения, вывод результата и выход.



Рисунок 2.2.2 – Граф диалога «Распознание»

## Разработка методов основных форм

Список методов, реализуемых на различных формах приведен на рисунке 2.3.1.

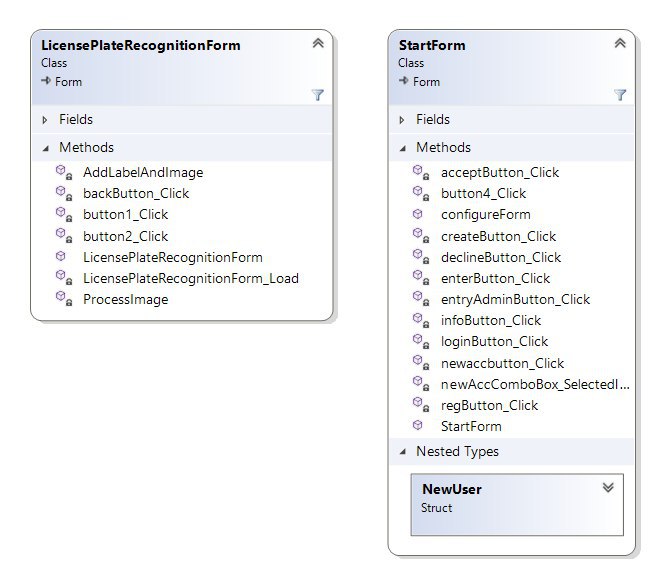


Рисунок 2.3.1 – Методы соответствующих классов

Таблица, отображающая поля структуры, необходимой для работы с базой данных приведена на рисунке 2.3.2

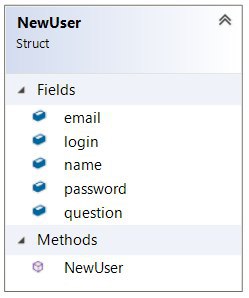


Рисунок 2.3.2 – Схема структуры

## Разработка форм интерфейса

Исходя из логики, описанной в предыдущих пунктах, интерфейс разрабатываемой программы будет состоять из двух основных частей: главная страница, содержащая в себе страницу авторизации, страницу регистрации, страницу справки и кнопку выхода, и страница распознания, содержащая форму для вывода изображения, форму для вывода результатов распознания, форму для выбора изображения.

На всех страницах первой части постоянно будет находится меню перехода на другую страницу в левой части.

Неавторизованному пользователю для перехода на часть распознания необходимо авторизоваться, перейдя на страницу авторизации через нажатие кнопки «Войти».

Полученный в результате проектирования интерфейс приведен на рисунках 2.4.1-2.4.13

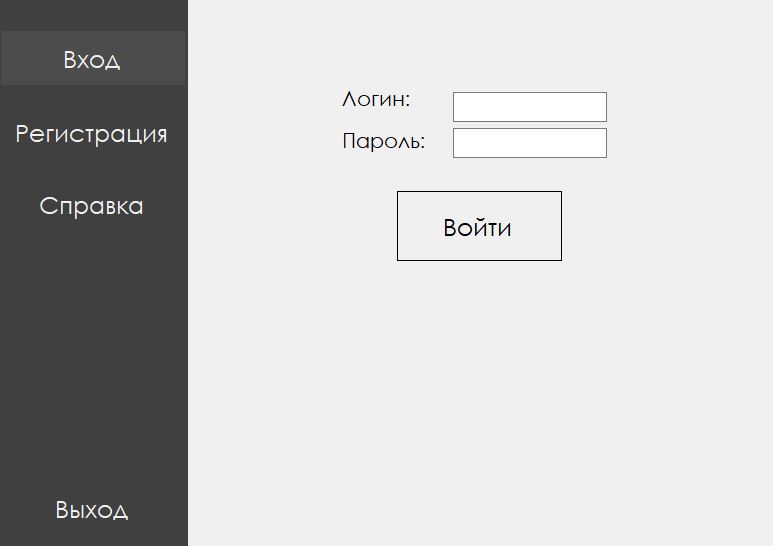


Рисунок 2.4.1 – Форма авторизации пользователя

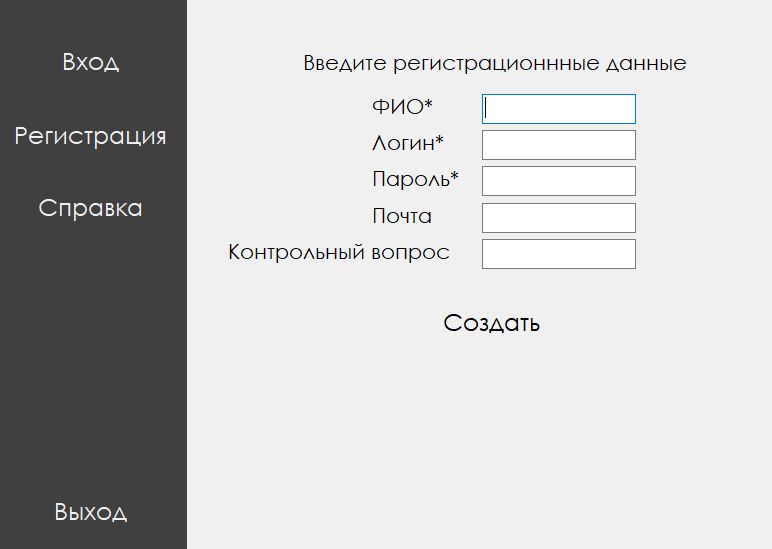


Рисунок 2.4.2 – Форма регистрации пользователя

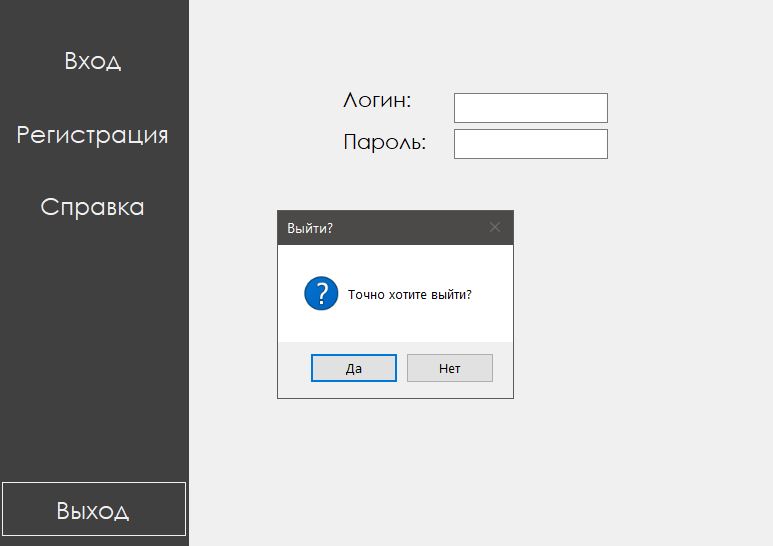
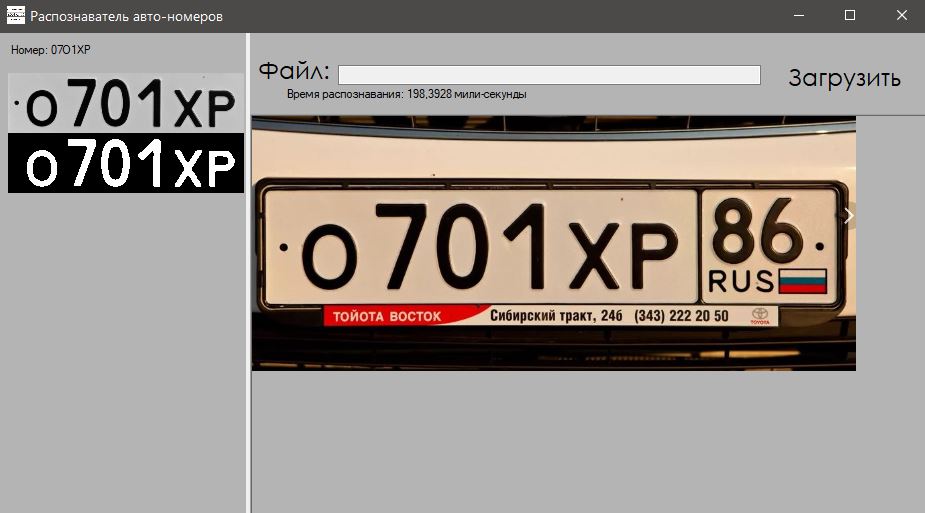


Рисунок 2.4.3 – Форма выхода из приложения

 Рисунок 2.4.4 – Форма распознания с шаблоном

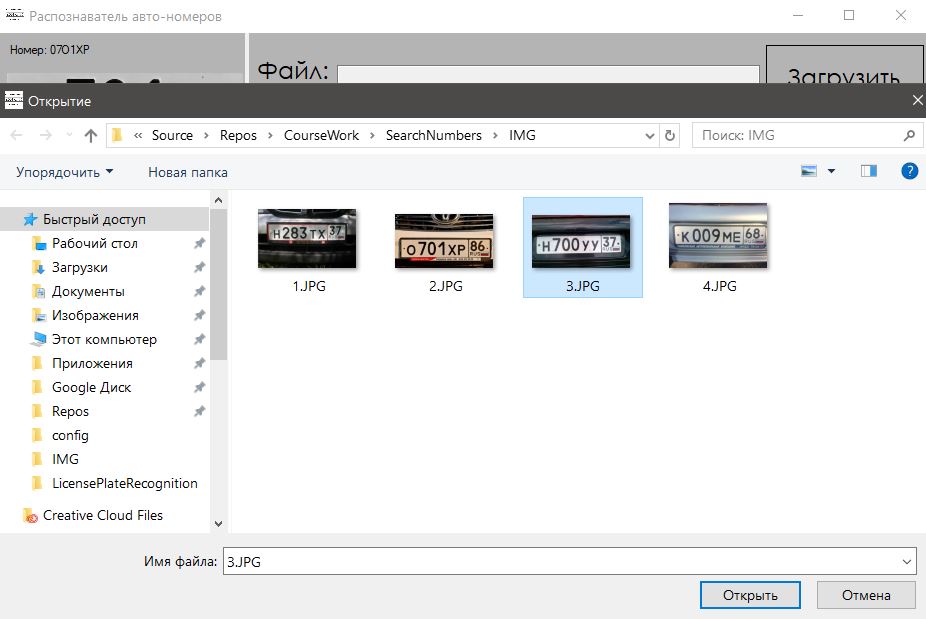


Рисунок 2.4.5 – Форма выбора изображения



Рисунок 2.4.6 – Форма при выборе изображения



Рисунок 2.4.7 – Форма при выборе изображения

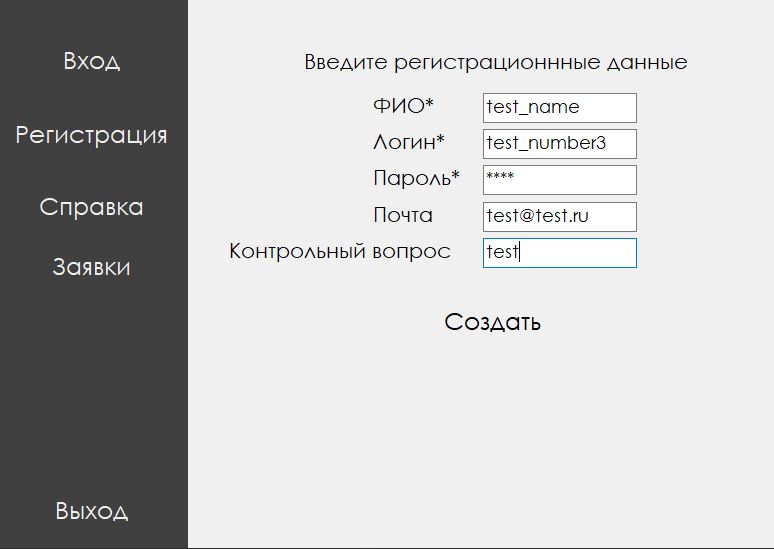


Рисунок 2.4.8 – Форма регистрации при вводе данных

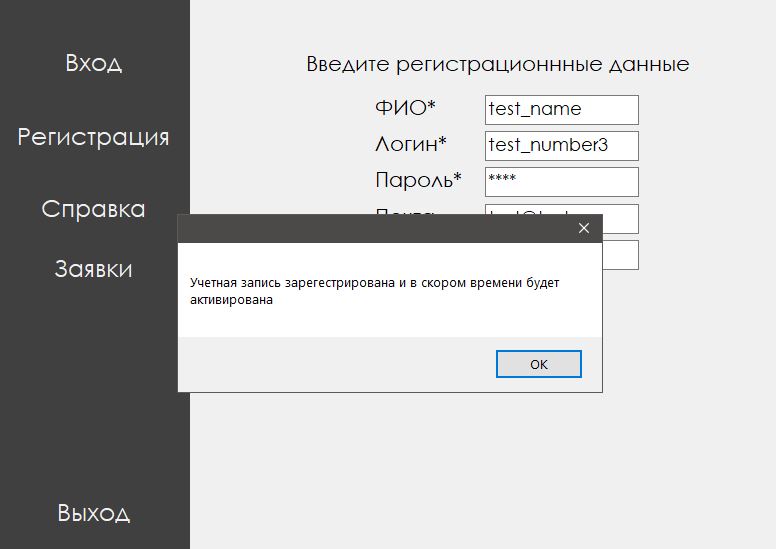


Рисунок 2.4.9 – Форма регистрации при отправке заявки

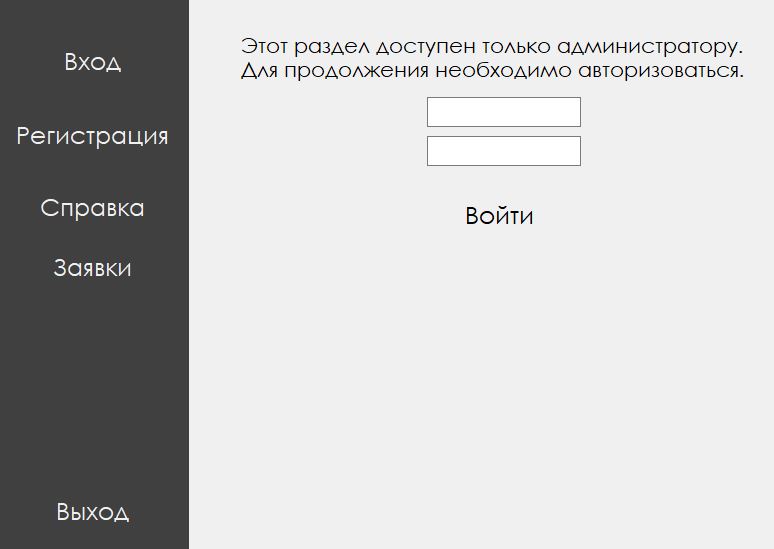


Рисунок 2.4.10 – Форма заявок для администратора

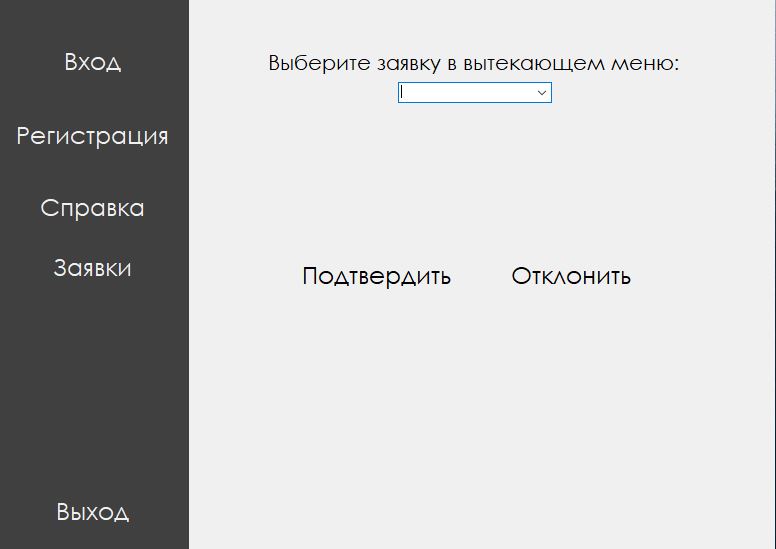


Рисунок 2.4.11 – Форма заявок после авторизации администратора

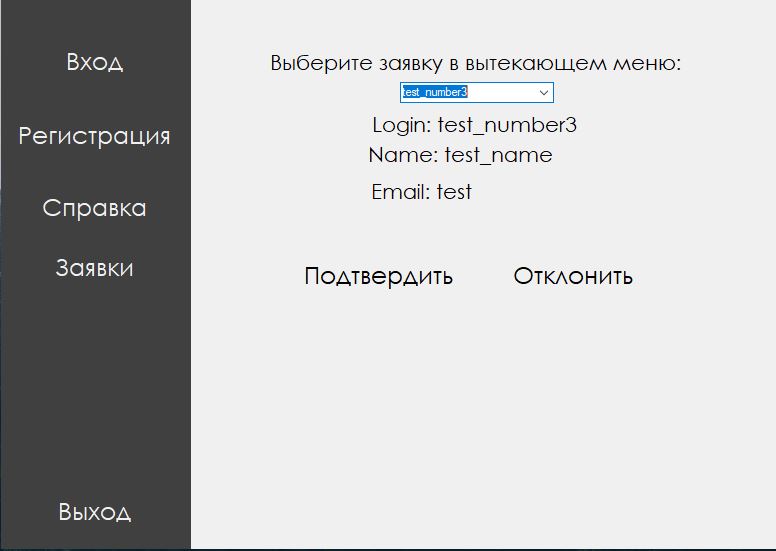


Рисунок 2.4.12 – Форма заявок – вывод информации о пользователе

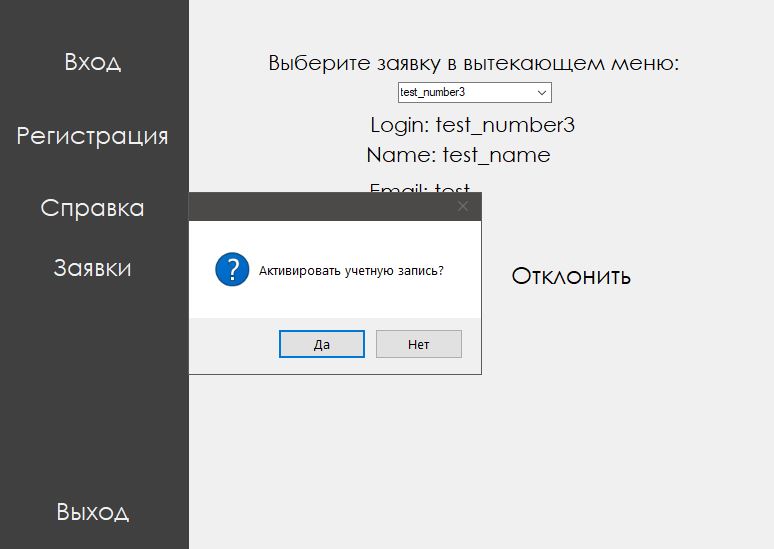


Рисунок 2.4.13 – Форма заявок – активация учетной записи

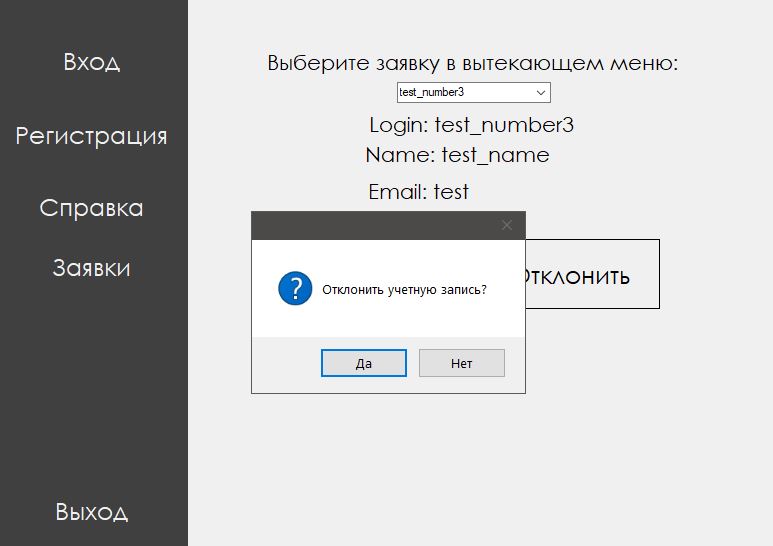


Рисунок 2.4.13 – Форма заявок – отклонение учетной записи

# Тестирование приложения

Опишем проведенное тестирование по принципу «Черного ящика», рассмотрев некоторые виды функционального тестирования.

## Тестирование граничных условий

Ввод данных в приложении осуществляется только для добавляемых или редактируемых пользователей.

Проведем тестирование граничных условий для этой модели.

Таблица 3.1.1 - Тестирование граничных условий

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Ожидаемый результат** | **Результат** | **Вывод** |
| Ввод в поля ФИО, почты, пароля, вопроса пустых значений | Уведомление о некорректном вводе (пустые поля) | Уведомление о некорректном вводе (пустые поля) | Успешно |
| Ввод во второе поля ввода пароля пароль, отличный от первого | Уведомление о неверном пароле | Уведомление о неверном пароле | Успешно |
| Ввод в поле логина некорректных символов | Уведомление о некорректном вводе логина | Уведомление о некорректном вводе логина | Успешно |

## Тестирование методом эквивалентных разбиений

Определим классы эквивалентности. Выделим минимальную длину вводимых данных для всех текстовых полей. Получим 2 класса эквивалентности: верный – более 5 символов, и неверный – 4 символа и меньше. В случае, если не введено ни одного символа, будет выведено сообщение об ошибке. Помимо этого, введем проверку формата электронной почты пользователя.

Таблица 3.2.1 – Тестирование методом эквивалентных разбиений

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Ожидаемый результат** | **Результат** | **Вывод** |
| Ввод в поле пароля при регистрации значение длинной менее 5 символов | Уведомление о некорректном вводе пароля | Уведомление о некорректном вводе пароля | Успешно |
| Ввод в поле логина при регистрации значение длинной менее 5 символов | Уведомление о некорректном вводе логина | Уведомление о некорректном вводе логина | Успешно |
| Ввод в поле почты данные, не подходящие под шаблон почты | Уведомление о некорректном вводе почты | Уведомление о некорректном вводе почты | Успешно |
| Оставить поле «Вопрос» пустым | Нет никакого уведомления, так как вопрос может отсутствовать | Отсутствие уведомления | Успешно |

## Тестирование причинно-следственных связей

В данном части тестирования большую часть работы занимает исследование модуля распознания и немного модуля авторизации. Результат анализа причинно-следственных связей приведен на таблице 3.3.1.

Таблица 3.3.1 - Анализ причинно-следственных связей

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Ожидаемый результат** | **Результат** | **Вывод** |
| Ввод при регистрации занятого логина | Сообщение о занятости логина | Сообщение о занятости логина | Успешно |
| Открытие списка текущих заявок при их отсутствии | Вывод пустого поля | Вывод пустого поля | Успешно |
| Первое открытие окна «Распознание» без изображения | Страница содержит шаблон со стандартной информацией | Страница содержит шаблон со стандартной информацией | Успешно |
| Выбор изображения без автомобильного номера | Вывод не содержит никакого распознанного номера | Вывод не содержит никакого распознанного номера | Успешно |
| Выбор изображения с несколькими автомобильными номерами | Вывод нескольких автомобильных номеров | Вывод нескольких автомобильных номеров | Успешно |
| Выбор изображения с искаженным номером | Вывод номера, полученного с помощью библиотеки распознания | Вывод номера, полученного с помощью библиотеки распознания | Успешно |

# ЗАКЛЮЧЕНИЕ

В результате данной разработки получен прототип программного продукта «Вспомогательная подсистема идентификации автомобиля по номерному знаку на пропускном пункте», соответствующий всем требованиям технического задания. В его программную реализацию включена возможность расширения функционала и последующего улучшения алгоритма распознания.

Таким образом, можно сделать вывод о том, что приложение получилось технологичным, так как:

* Хорошо проработана модель программы, в связи с тем, что на этапе проектирования были выделены основные задачи, реализованные в приложении впоследствии;
* Приложение обладает проработанным пользовательским интерфейсом, что расширяет круг пользователей программы;
* Программа написана в соответствии со стилями руководства используемого языка. В коде программы присутствуют необходимые комментарии, имена переменных, функций и классов осмысленны, присутствуют отступы и пустые строки для облегчения чтения программы;
* Везде, где возможно, присутствует повторное использование кода;
* Разработанные модули обладают достаточной независимостью;
* Модули программы выделены таким образом, что они могут использоваться для других задач или являться общими для различных объектов.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Иванова Г.С. Технология программирования: учебник / Г.С. Иванова. 3-е изд., стер. М.: КНОРУС, 2016. 334 с. (Бакалавриат). ISBN 978-5-04734-7
2. «A Real-Time Mobile Vehicle License Plate Detection and Recognition» /  Kuo-Ming Hung and Ching-Tang Hsieh : гистограммный подход при распознавании номеров [Электронный ресурс]. URL: <http://www2.tku.edu.tw/~tkjse/13-4/09-IE9722.pdf>
3. «Automated Number Plate Recognition Using Hough Lines and Template Matching» /  Saqib Rasheed, Asad Naeem and Omer Ishaq: поиск номеров через HOG-дескрипторы вертикальных линий [Электронный ресурс].
4. Крашенинников В. Р. Основы теории обработки изображений: Сборник лабораторных работ. Ульяновск: УлГТУ, 2004.
5. Цикл статей на «Habr.com» [Электронный ресурс], посвященных распознанию автомобильных номеров в деталях. URL: https://habr.com/company/recognitor/blog/225913/

# ПРИЛОЖЕНИЕ А. Техническое задание

# ПРИЛОЖЕНИЕ Б. Руководство пользователя

# ПРИЛОЖЕНИЕ В. Фрагменты исходного кода программы