**Министерство образования Приморского края**

**КРАЕВОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ПРИМОРСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

Отделение экономики и сервиса

Специальность *09.02.04*

Информационные системы (по отраслям)

Работа допущена к защите:

Завотделением \_\_\_\_\_\_\_ А.А.Савина

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**КУРСОВАЯ ПРОЕКТ**

**по теме:**

**БЮРО НАХОДОК И ЗАБЫТЫХ ВЕЩЕЙ**

**ОП 06 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

|  |  |
| --- | --- |
| **Выполнил:** | **Руководитель:** |
| студент группы 1121 | преподаватель |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.А.Воробьёв | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А.Пушкин |
|  |  |
| Оценка, полученная на защите КР, | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Владивосток 2020

**Министерство образования Приморского края**

**КРАЕВОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ПРИМОРСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

**Отделение экономики и сервиса**

**З А Д А Н И Е**

на курсовой проект

Заведующий отделением \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Савина

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

|  |
| --- |
| *студенту Воробьёву Николаю Анатольевичу 1121 группы* |

(фамилия, имя, отчество)

на тему: «Бюро находок и забытых вещей»

|  |
| --- |
|  |

Вопросы, подлежащие разработке (исследованию):

|  |
| --- |
| 1. Изучение предметной области. Выделение объектов и связей между ними. |
| 1. Написание технического задания на разработку системы |
| 1. Разработка классов для хранения и оперирования данными |
| 1. Разработка системы сохранения и загрузки данных |
| 1. Разработка форм для ввода и вывода данных |
| 1. Разработка специфичного функционала согласно тематике приложения |
| 1. Тестирование приложения на предмет ошибок |
|  |

Основные и другие источники информации, используемые для разработки темы

|  |
| --- |
| 1. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы |
| 1. Стиллмен Э., Грин Дж. - Изучаем C#. Включая C# 7.0 |
| 1. Петцольд Ч. - Программирование для Microsoft Windows на С#. |
|  |

Срок предоставления работы «03» июня 2020 г.

Дата выдачи задания «06 » апреля 2020 г.

Руководитель КП \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (и.о. фамилия)

Задание получил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ:**

[ВВЕДЕНИЕ 3](#_Toc42009548)

[1 Проектирование приложения 5](#_Toc42009549)

[1.1 Характеристика объектов 5](#_Toc42009550)

[1.2 Требования к программе 6](#_Toc42009551)

[1.2.1 Платформа клиентских частей приложения 6](#_Toc42009552)

[1.2.2 Требования к функциональным характеристикам 6](#_Toc42009553)

[1.3 Схема классов 9](#_Toc42009554)

[2. Разработка приложения 14](#_Toc42009555)

[2.1 Описание интерфейса приложения 14](#_Toc42009556)

[2.1.1 Список объявлений 14](#_Toc42009557)

[2.1.2 Редактирование объявлений 15](#_Toc42009558)

[2.1.3 Список стран 17](#_Toc42009559)

[2.1.4 Редактирование страны 18](#_Toc42009560)

[2.1.5 Список городов 18](#_Toc42009561)

[2.1.6 Редактирование города 20](#_Toc42009562)

[2.1.7 Список категорий 20](#_Toc42009563)

[2.1.8 Редактирование категории 21](#_Toc42009564)

[2.1.9 Список подкатегорий 22](#_Toc42009565)

[2.1.10 Редактирование подкатегории 24](#_Toc42009566)

[2.1.11 Отчёты 24](#_Toc42009567)

[2.2 Диаграммы потоков данных 26](#_Toc42009568)

[2.3 Пользовательские сценарии 28](#_Toc42009569)

[2.3.1 Создание объявления 28](#_Toc42009570)

[2.3.2 Просмотр объявлений 28](#_Toc42009571)

[2.3.3 Просмотр отчётов 29](#_Toc42009572)

[ЗАКЛЮЧЕНИЕ 30](#_Toc42009573)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31](#_Toc42009574)

**ВВЕДЕНИЕ**

В нашей жизни часто теряются ценные вещи: выпавшие ключи или документы, различные украшения, машины и потерявшиеся домашние питомцы. Благодаря добрым людям, которые хотели помочь как-то улучшить поиски и увеличить вероятность, что потерянные вещи будут найдены и отданы владельцам, на свет появились бюро находок.

Раньше бюро находок были весьма популярны. Большинство людей, обнаружившие пропажи, без проблем могли отнести их в бюро, как и люди, которые что-то или кого-то потеряли, могли спокойно обратиться туда. Но проблемы этих бюро были в том, что, если, к примеру, люди были в отпуске в каком-то городе и у них потерялась собака, добрый человек, найдя эту собаку, на ошейнике обнаружит другой город и решит отвезти собаку туда или даже может по ошибке отвезти в другое место. Люди, обратившиеся в бюро, не смогут получить информацию о своей собаке может даже через полгода.

Некоторые люди, потеряв какие-нибудь вещи, пытаются размещать объявления о пропаже на различных сайтах, в социальных сетях и т. д. Но вероятность, что человек, обнаруживший эти вещи, сможет найти это объявление, крайне мала.

Бывают ещё бюро находок в супермаркетах, торговых центрах и т. д. Это хорошая идея, поскольку, забыв что-то в том же торговом центре, можно легко обратить в бюро, а не искать ближайшую контору бюро находок. Но бывает и так, что люди просто не понят, в каком именно супермаркете они потеряли свои вещи, и потому, в данной ситуации найти их будет гораздо сложнее.

Для упрощения работы бюро находок в данном курсовом проекте будет спроектировано и разработано приложение "Бюро находок и забытых вещей". Это приложение будет хранить в себе информацию о всех найденных и потерянных вещях, питомцах и т. д. Но, всё же, сами конторы бюро находок нужно оставить, дабы люди, не пользующиеся интернетом, смогли просто передать найденную вещь или написать объявление о пропаже, после чего, эти объявления будут переданы на хранение в приложение.

Также благодаря этому приложению не обязательно нужно будет идти за пропавшим предметом в бюро, а можно будет договориться о встрече с обнаружившим эту вещь.

Ещё плюсом приложения будет является возможность отслеживать потерянные вещи в различных городах и странах, дабы расширить область поисков пропавших вещей и их владельцев.

Следовательно, целью проекта является разработка приложения бюро находок. Для достижения этой цели нужно выполнить следующие задачи:

1. Рассмотреть характеристику объектов;
2. Сформировать требования к приложению;
3. Изобразить схемы классов;
4. Спроектировать базы данных;
5. Описать интерфейс приложения;
6. Изобразить диаграммы потоков данных;
7. Спроектировать алгоритм работы приложения;
8. Создать пользовательский сценарий.

**1 Проектирование приложения**

**1.1 Характеристика объектов**

Бюро находок занимается следующей работой:

1) Хранит в канторе потерявшиеся вещи или информацию о них;

2) Создаёт объявления о пропавших вещях;

3) Принимает потерявшиеся вещи от людей, обнаруживших их;

4) Возвращает потерявшиеся вещи владельцам;

В ходе проектирования данного курсового проекта были выявлены следующие сущности:

1) Категория. О ней следует хранить следующую информацию:

1.1) Название категории;

1.2) Список подкатегорий этой категории.

2) Подкатегория. О ней следует хранить следующую информацию:

2.1) Название подкатегории;

2.2) Категорию, которой подкатегория принадлежит.

3) Страна. О ней следует хранить следующую информацию:

3.1) Название страны;

3.2) Список городов этой страны.

4) Город. О нём следует хранить следующую информацию:

4.1) Название города;

4.2) Страну, которой город принадлежит.

5) Объявление. О нём следует хранить следующую информацию:

5.1) Заголовок объявления;

5.2) Категория, которой принадлежит потерянная вещь;

5.3) Подкатегория, которой принадлежит потерянная вещь;

5.4) Страна, в которой была потеряна или найдена вещь;

5.5) Город, в котором была потеряна или найдена вещь;

5.6) Адрес, где можно получить или отдать потерянную вещь;

5.7) Описание потерянной или найденной вещи;

5.8) Номер телефона, нашедшего или потерявшего вещь;

5.9) Контактное лицо, обнаружившее или потерявшее вещь;

5.10) Статус обнаружения (найдено, потеряно);

5.11) Время, когда объявление было добавлено;

5.12) Статус заявки (открыта, закрыта).

Приложение проектируются для работы одного пользователя под управлением операционной системы Windows.

**1.2 Требования к программе**

**1.2.1 Платформа клиентских частей приложения**

- Windows – 7 и выше

- .NET Framework 4.7 и ее доработанный выпуск (4.7.1 и 4.7.2).

**1.2.2 Требования к функциональным характеристикам**

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

**1.2.2.1 Предоставление контента**

При запуске программы должна запускаться главная форма под названием “Бюро находок и забытых вещей”. Она должна содержать следующую информацию:

- Список с объявлениями о пропаже или обнаружении вещей, животных и т.д.

- Возможность поиска по критериям определённых объявлений.

- Возможность просматривать объявления из списка постранично с помощью прокрутки и кнопок.

- Возможность открывать формы “Добавление объявления”, “Список стран”, “Список городов”, “Список категорий”, “Список подкатегорий”, “Информация об объявлении”, “Отчёты”.

**1.2.2.2 Функционал “Добавление объявления”**

- Должен производиться ввод заголовка объявления, адреса, телефона, контактного лица, описания найденной вещи; выбор категории, подкатегории, страны, города, статуса обнаружения.

- После введённых данных должна быть возможность их сохранить и закрыть форму.

**1.2.2.3 Функционал “Список стран”**

- Должно производится отображения всех добавленных стран.

- Должна быть возможность добавить новую страну, редактировать или удалить существующею.

- Возможность просматривать страны из списка постранично с помощью прокрутки и кнопок.

**1.2.2.4 Функционал “Список городов”**

- Должно производится отображения всех добавленных городов при выборе определённой страны.

- Должна быть возможность добавить новый город выбранной стране, редактировать или удалить существующий.

- Возможность просматривать города из списка постранично с помощью прокрутки и кнопок.

**1.2.2.5 Функционал “Список категорий”**

- Должно производится отображения всех добавленных категорий.

- Должна быть возможность добавить новую категорию, редактировать или удалить существующею.

- Возможность просматривать категории из списка постранично с помощью прокрутки и кнопок.

**1.2.2.6 Функционал “Список подкатегорий”**

- Должно производится отображения всех добавленных подкатегорий при выборе определённой категории.

- Должна быть возможность добавить новую подкатегорию выбранной категории, редактировать или удалить существующею.

- Возможность просматривать подкатегории из списка постранично с помощью прокрутки и кнопок.

**1.2.2.7 Функционал “Информация об объявлении”**

- Должно производится отображения всех введённых данных при создании объявления.

- Должна быть возможность редактирования введённых данных объявления.

- Должна быть возможность закрыть объявление.

**1.2.2.8 Функционал “Редактирование страны”**

- Должно производится отображение названия редактируемой страны или, при создании новой, пустое поле.

- Должна быть возможность редактирования названия страны.

- После введённых данных должна быть возможность их сохранить и закрыть форму.

**1.2.2.9 Функционал “Редактирование города”**

- Должно производится отображение названия редактируемого города или, при создании нового, пустое поле.

- Должна быть возможность редактирования названия города.

- После введённых данных должна быть возможность их сохранить и закрыть форму.

**1.2.2.10 Функционал “Редактирование категории”**

- Должно производится отображение названия редактируемой категории или, при создании новой, пустое поле.

- Должна быть возможность редактирования названия категории.

- После введённых данных должна быть возможность их сохранить и закрыть форму.

**1.2.2.11 Функционал “Редактирование подкатегории”**

- Должно производится отображение названия редактируемой подкатегории или, при создании новой, пустое поле.

- Должна быть возможность редактирования названия подкатегории.

- После введённых данных должна быть возможность их сохранить и закрыть форму.

**1.2.2.12 Функционал “Отчёты”**

- Должна быть возможность отображения объявлений по статусу заявки.

- Должна быть возможность отображения объявлений по промежутку времени.

- Должна быть возможность отображения объявлений по своей дате.

- Должен производится счёт количества найденных по критериям объявлений.

- Возможность просматривать объявления из списка постранично с помощью прокрутки и кнопок.

**1.3 Схема классов**

Классы Category, SubCategory, Country, City, Advertisement предназначены для хранения данных.

Интерфейс IViewRow предназначен для ослабления связей между классами Category, SubCategory, Country, City, Advertisement.

Класс ListViewViewer предназначен для отображения на listview данных, передаваемых с помощью интерфейса IViewRow.

Классы CategoryForm, SubcategoryForm, CountryForm, CityForm, MainForm, ReportForm предназначены для отображения списков объектов, хранящих данные.

Классы AddCategoryForm**,** AddSubcategoryForm, AddCountryForm, AddCityForm, AddAdvertisementForm предназначены для добавления, редактирования и удаления объектов, хранящих данные.

Классы CategoryDB, CountryDB, DiscoveredDB, SubCategoryDB, CityDB, AdvertisementDB, FilterDB предназначены для формирования списков объектов, хранящих данные.

Интерфейс IDB<T> предназначен для ослабления связей между классами CategoryDB, CountryDB, DiscoveredDB, SubCategoryDB, CityDB, AdvertisementDB, FilterDB.

Класс Paginator<T, V> предназначен для отображения коллекции данных постранично и для возможности эти страницы просматривать.



Рисунок 1 – Схема классов

На рисунке 1 изображена схема классов.

Классы Category, SubCategory, Country, City, Advertisement связываются благодаря реализации интерфейса IViewRow.

Интерфейс IViewRow связывается с классом ListViewViewer благодаря связи ассоциация, потому что IViewRow используется в перечислителе в качестве типа данных, который является параметром метода ViewData.

Класс ListViewViewer связывается с классами CategoryForm, SubcategoryForm, CountryForm, CityForm, MainForm, ReportForm благодаря связи композиция, поскольку в этих классах созданы экземпляры класса ListViewViewer.

Класс CategoryForm связывается с классом AddCategoryForm благодаря связи композиция, поскольку в CategoryForm создан экземпляр класса AddCategoryForm.

Класс SubcategoryForm связывается с классом AddSubcategoryForm благодаря связи композиция, поскольку в SubcategoryForm создан экземпляр класса AddSubcategoryForm.

Класс CountryForm связывается с классом AddCountryForm благодаря связи композиция, поскольку в CountryForm создан экземпляр класса AddCountryForm.

Класс CityForm связывается с классом AddCityForm, благодаря связи композиция, поскольку в CityForm создан экземпляр класса AddCityForm.

Классы CategoryDB, SubCategoryDB, CountryDB, CityDB, AdvertisementDB, FilterDB связываются благодаря реализации интерфейса IDB<T>.

Интерфейс IDB<T> связывается с классом Paginator<T, V> благодаря связи агригация, поскольку в конструктор класса Paginator<T, V> предаётся шаблонный экземпляр интерфейса IDB<T>.

Классы CategoryDB, CountryDB, DiscoveredDB связываются с классом AdvertisementDB благодаря связи композиция, поскольку в AdvertisementDB созданы экземпляры этих классов.

Классы CategoryForm, SubcategoryForm, CountryForm, CityForm, AddAdvertisementForm, ReportForm, CategoryDB, CountryDB, DiscoveredDB, AdvertisementDB, FilterDB, Paginator<T, V> связываются с классом MainForm благодаря связи композиция, поскольку в MainForm созданы экземпляры этих классов.

Классы CategoryDB, CountryDB, DiscoveredDB, AdvertisementDB связываются с классом AddAdvertisementForm благодаря связи агригация, поскольку в класс AddAdvertisementForm предаются в конструктор экземпляры этих классов.

Класс SubCategoryDB связан с классом CategoryDB благодаря связи агригация, поскольку в класс SubCategoryDB передаётся экземпляр класса CategoryDB.

Класс CityDB связан с классом CountryDB благодаря связи агригация, поскольку в класс CityDB передаётся экземпляр класса CountryDB.

Класс Paginator<T, V> связан с классами CategoryForm, SubcategoryForm, CountryForm, CityForm, ReportForm, MainForm с помощью связи композиция, так как в этих классах создан экземпляр класса Paginator<T, V>.

**2. Разработка приложения**

**2.1 Описание интерфейса приложения**

**2.1.1 Список объявлений**

На рисунке 2 изображена форма “Список объявлений”.

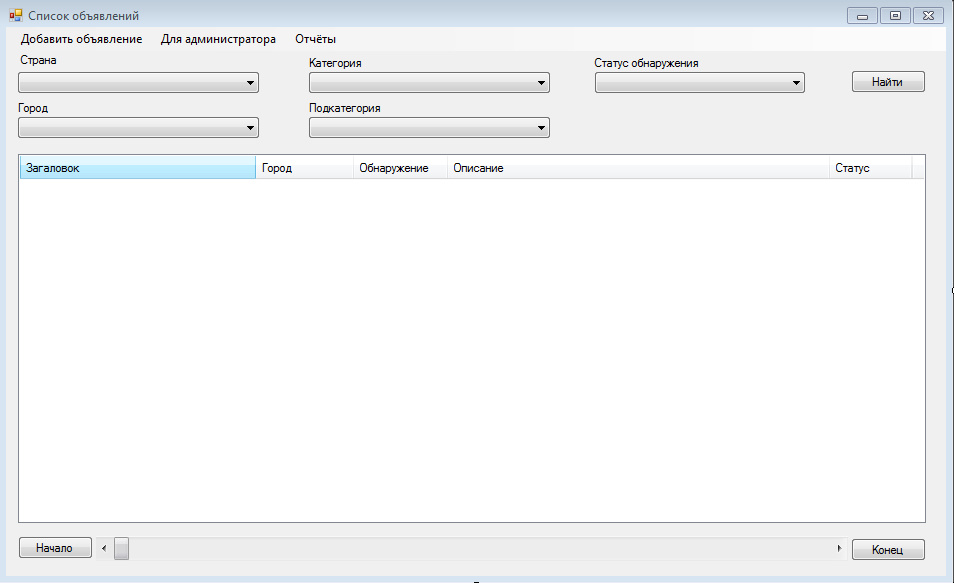
****

Рисунок 2 – Форма “Список объявлений”

Список предназначен для отображения созданных объявлений.

Выпадающие списки “Страна”, “Город”, “Категория”, “Подкатегория”, “Статус обнаружения” предназначены для выбора значений, по которым будут искаться объявления, соответствующие поля которых будут равны выбранным значениям.

Кнопка “Найти” предназначена для передачи объявлений, соответствующих выбранным в выпадающих списках значениях, в список для их последующего отображения в нём.

Кнопка “Начало” предназначена для перемотки списка на первую страницу.

Кнопка “Конец” предназначена для перемотки списка на последнюю страницу.

Полоса прокрутки предназначена для перемотки списка постранично.

Вкладки в меню “Добавить объявление” и “Отчёты” предназначены для открытия форм, соответствующих своему названию.

Вкладка в меню “Для администратора” является папкой, содержащей вкладки “Добавить страну”, “Добавить город”, “Добавить категорию”, “Добавить подкатегорию”, которые открывают соответствующие названиям формы со списками.

Двойное нажатие на элемент списка предназначено для открытия формы “Редактирование объявлений” с переданными в неё данными из выбранного элемента списка.

**2.1.2 Редактирование объявлений**

На рисунке 3 изображена форма “Редактирование объявления”.

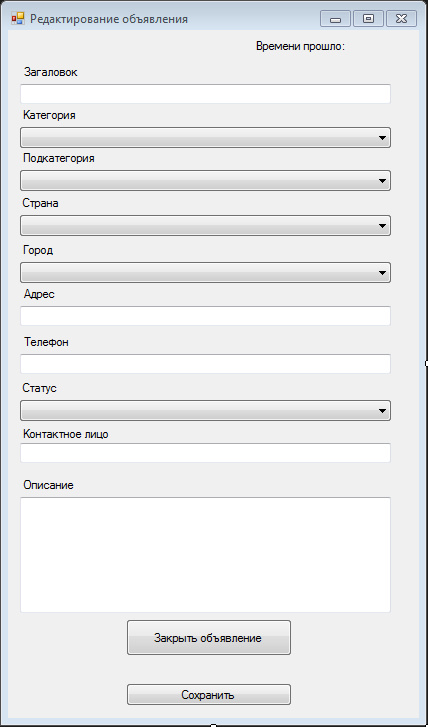
****

Рисунок 3 – Форма “Редактирование объявления”

Текстовые поля “Заголовок”, “Адрес”, “Телефон”, “Контактное лицо”, “Описание” предназначены для ввода в них пользователем данных о потерянной или найденной вещи, животном и т.д.

Выпадающие списки “Категория”, “Подкатегория”, “Страна”, “Город”, “Статус” предназначены для выбора в них данных, соответствующих потерянной или найденной вещи, животном и т.д.

Кнопка “Сохранить” предназначена для записи всех введённых в текстовые поля и выбранных в выпадающих списках данных в новое или старое объявление и для закрытия формы.

Кнопка “Закрыть объявление” предназначена для изменения статуса объявления с “Открытое” на “Закрытое” и для запрещение последующего изменения данных в объявлении.

Надпись: “Времени прошло:” предназначена для отображения количества дней, часов и т.д. с создания объявления.

**2.1.3 Список стран**

На рисунке 4 изображена форма “Список стран”.

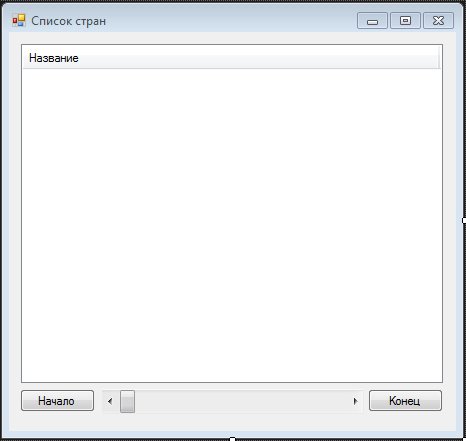
****

Рисунок 4 – Форма “Список стран”

Список предназначен для отображения добавленных стран.

Кнопка “Начало” предназначена для перемотки списка на первую страницу.

Кнопка “Конец” предназначена для перемотки списка на последнюю страницу.

Полоса прокрутки предназначена для перемотки списка постранично.

Вкладки “Добавить”, “Редактировать”, “Удалить” предназначены для соответствующих названиям действий со странами.

**2.1.4 Редактирование страны**

На рисунке 5 изображена форма “Редактирование страны”.

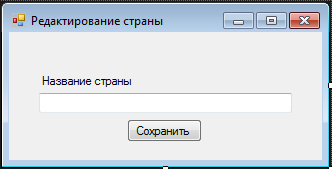
****

Рисунок 5 – Форма “Редактирование страны”

Текстовое поле “Название страны” предназначено для введения или изменения пользователем соответствующих названию данных о стране.

Кнопка “Сохранить” предназначена для сохранения или изменения данных о стране и закрытия формы.

**2.1.5 Список городов**

На рисунке 6 изображена форма “Список городов”.

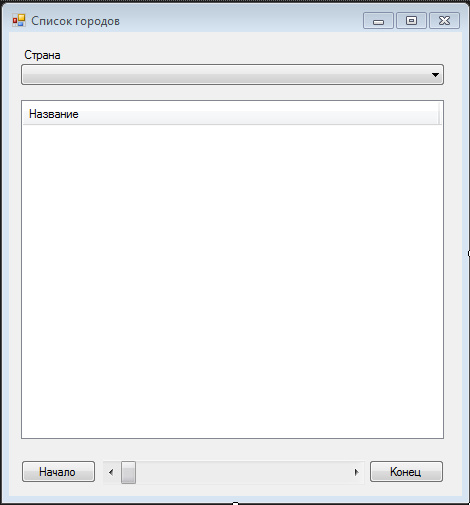


Рисунок 6 – Форма “Список городов”

Список предназначен для отображения всех добавленных городов определённой страны.

Выпадающий список “Страна” используется как фильтр для отображения городов в списке.

Кнопка “Начало” предназначена для перемотки списка на первую страницу.

Кнопка “Конец” предназначена для перемотки списка на последнюю страницу.

Полоса прокрутки предназначена для перемотки списка постранично.

Вкладки “Добавить”, “Редактировать”, “Удалить” предназначены для соответствующих названиям действий с городами выбранной в выпадающем списке страны.

**2.1.6 Редактирование города**

На рисунке 7 изображена форма “Редактирование города”.

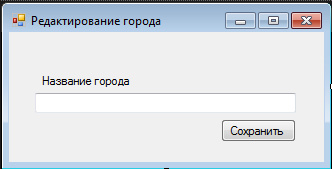


Рисунок 7 – Форма “Редактирование города”

Текстовое поле “Название города” предназначено для введения или изменения пользователем соответствующих названию данных о городе.

Кнопка “Сохранить” предназначена для сохранения или изменения данных о городе и закрытия формы.

**2.1.7 Список категорий**

На рисунке 8 изображена форма “Список категорий”.

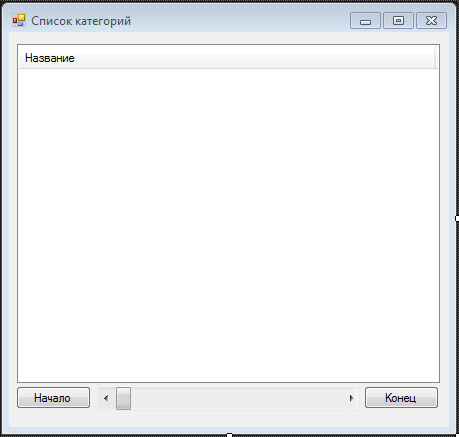


Рисунок 8 – Форма “Список категорий”

Список предназначен для отображения добавленных категорий.

Кнопка “Начало” предназначена для перемотки списка на первую страницу.

Кнопка “Конец” предназначена для перемотки списка на последнюю страницу.

Полоса прокрутки предназначена для перемотки списка постранично.

Вкладки “Добавить”, “Редактировать”, “Удалить” предназначены для соответствующих названиям действий с категориями.

**2.1.8 Редактирование категории**

На рисунке 9 изображена форма “Редактирование категории”.

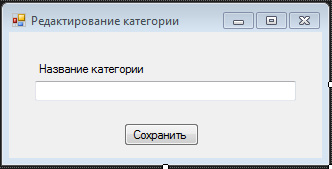


Рисунок 9 – Форма “Редактирование категорий”

Текстовое поле “Название категории” предназначено для введения или изменения пользователем соответствующих названию данных о категории.

Кнопка “Сохранить” предназначена для сохранения или изменения данных о категории и закрытия формы.

**2.1.9 Список подкатегорий**

На рисунке 10 изображена форма “Список подкатегорий”.

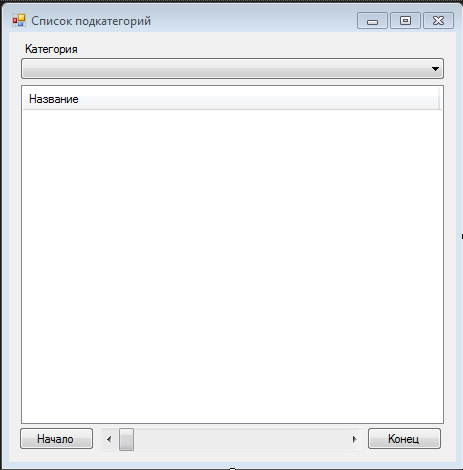


Рисунок 10 – Форма “Список подкатегорий”

Список предназначен для отображения всех добавленных подкатегорий определённой категории.

Выпадающий список “Категория” используется как фильтр для отображения подкатегорий в списке.

Кнопка “Начало” предназначена для перемотки списка на первую страницу.

Кнопка “Конец” предназначена для перемотки списка на последнюю страницу.

Полоса прокрутки предназначена для перемотки списка постранично.

Вкладки “Добавить”, “Редактировать”, “Удалить” предназначены для соответствующих названиям действий с подкатегориями выбранной в выпадающем списке категории.

**2.1.10 Редактирование подкатегории**

На рисунке 11 изображена форма “Редактирование подкатегории”.

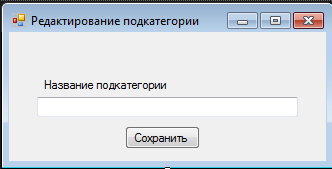


Рисунок 11 – Форма “Редактирование подкатегории”

Текстовое поле “Название подкатегории” предназначено для введения или изменения пользователем соответствующих названию данных о подкатегории.

Кнопка “Сохранить” предназначена для сохранения или изменения данных о подкатегории и закрытия формы.

**2.1.11 Отчёты**

На рисунке 12 изображена форма “Отчёты”.

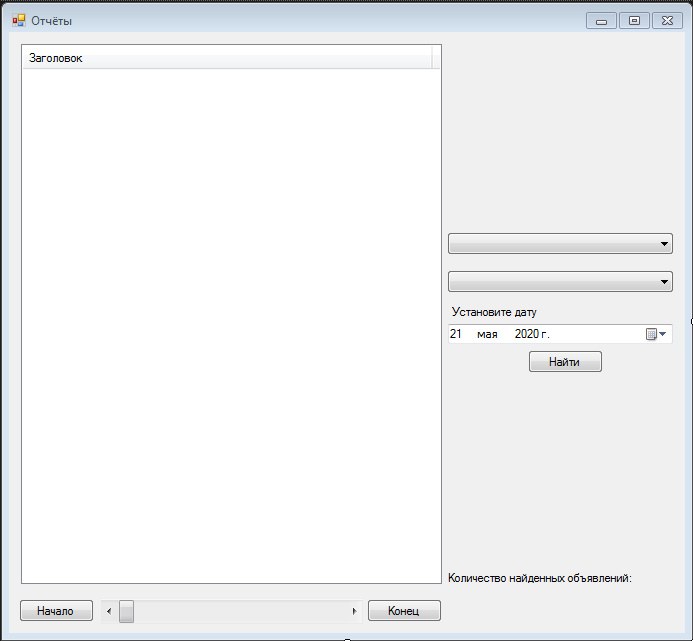


Рисунок 12 – Форма “Отчёты”

Список предназначен для отображения объявлений по статусу и временному промежутку.

Кнопка “Начало” предназначена для перемотки списка на первую страницу.

Кнопка “Конец” предназначена для перемотки списка на последнюю страницу.

Полоса прокрутки предназначена для перемотки списка постранично.

Первый выпадающий список используется как фильтр, указывающий на статус заявки.

Второй выпадающий список используется как фильтр, указывающий определённый промежуток времени, когда было создано объявление.

Поле выбора даты используется, когда во втором выпадающем списке выбрано “Своя дата”, как фильтр, указывающий определённый день, когда было создано объявление.

Кнопка “Найти” предназначена для передачи критериев из выпадающих списков и поля выбора даты в список, для отображения объявлений, соответствующих этим критериям.

Надпись: “Количество найденных объявлений:” предназначена для подсчёта объявлений, отображённых в списке.

Двойное нажатие на элемент списка предназначено для открытия формы “Редактирование объявлений” с переданными в неё данными из выбранного элемента списка.

**2.2 Диаграммы потоков данных**

На рисунке 14 изображена диаграмма “Работа с данными”.



Рисунок 14 – Работа с данными

1) При нажатии пользователем в контекстном меню любой из форм со списком, происходит создание пустого экземпляра класса.

2) Созданный пустой экземпляр класса сохраняется в файл.

3) Происходит запрос на введение данных пользователем.

4) После нажатия пользователем на кнопу “Сохранить”, происходит сохранение данных в созданный экземпляр класса.

5) Происходит сохранение изменённого экземпляра класса в файл.

6) Происходит изъятие всех экземпляров класса из файла.

7) По количеству экземпляров класса создаются пустые экземпляры строк списка.

8) Экземпляры класса и экземпляры строк списка передаются на отображение данных.

На рисунке 15 изображена диаграмма “Сохранение данных в файл”.



Рисунок 15 – Сохранение данных в файл

1) Созданный экземпляр класса добавляется в коллекцию экземпляров класса.

2) Пересоздаётся файл.

3) Коллекция экземпляров класса записывается в файл.

На рисунке 16 изображена диаграмма “Загрузка данных из файла”.



Рисунок 16 – Загрузка данных из файла

1) Происходит открытие файла.

2) Происходит считывание данных с файла.

3) Считанные данные с файла сохраняются в коллекцию экземпляров класса.

**2.3 Пользовательские сценарии**

**2.3.1 Создание объявления**

1) Нажать на вкладку в меню “Добавить объявление”.

2) В текстовые поля “Заголовок”, “Адрес”, “Телефон”, “Контактное лицо”, “Описание” ввести данные.

3) В выпадающих списках “Категория”, “Подкатегория”, “Страна”, “Город”, “Статус” выбрать данные.

4) Нажать на кнопку “Сохранить”.

**2.3.2 Просмотр объявлений**

1) В выпадающих списках “Страна”, “Категория”, “Статус обнаружения” выбрать данные или оставить пустыми.

2) При выборе данных в выпадающих списках “Страна”, “Категория”, появляются выпадающие списки “Город”, “Подкатегория” соответственно.

3) В выпадающих списках “Город”, “Подкатегория” выбрать данные или оставить пустыми.

4) Нажать на кнопку “Найти”.

5) С помощью полосы прокрутки или без неё найти нужное объявление.

6) Двойным щелчком мыши нажать на нужное объявление.

7) Изменить данные об объявлении или оставить неизменными.

8) Нажать на кнопку “Сохранить” или, если нужно закрыть заявку, на кнопку “Закрыть заявку”.

**2.3.3 Просмотр отчётов**

1) Нажать на вкладку в меню “Отчёты”.

2) Выбрать в выпадающих списках нужные критерии поиска.

3) При выборе во втором выпадающем списке значения “Своя дата”, появляется надпись: “Установить дату” и поле выбора даты.

4) Выбрать в поле выбора даты нужное значение.

5) Нажать на кнопку “Найти”.

6) С помощью полосы прокрутки или без неё найти нужное объявление.

7) Двойным щелчком мыши нажать на нужное объявление.

8) Изменить данные об объявлении или оставить неизменными.

9) Нажать на кнопку “Сохранить” или, если нужно закрыть заявку, на кнопку “Закрыть заявку”.

**ЗАКЛЮЧЕНИЕ**

В приложении удалось реализовать следующие возможности:

1) Просматривать объявления, страны, города определённой страны, категории, подкатегории определённой категории при помощи постраничной прокрутки.

2) Добавлять объявления, категории, страны, города определённой стране, подкатегории определённой категории.

3) Редактирования объявления, категории, страны, города, подкатегории.

4) Удалять категории, страны, города, подкатегории.

5) Закрывать объявления.

6) Выбирать критерии, по которым будут отображаться объявления.

7) Просматривать объявления по статусу и времени создания.

8) Сохранять созданные объявления, категории, страны, города, подкатегории в файлы.

9) Загружать созданные объявления, категории, страны, города, подкатегории из файлов**.**

Данное приложение поможет облегчить работу:

1) Людям, потерявших или нашедших вещь, животное и т.д.

2) Людям, работающих в бюро находок.

3) Возможностью хранения огромного количества объявлений.

4) Возможностью просматривать объявления и узнать количество объявлений, созданных за определённый промежуток времени.

5) Возможностью по критериям быстро находить нужные объявления.

6) Возможностью добавления, редактирования, удаления стран, городов, категорий и подкатегорий.

7) Возможностью закрывать объявления, если вещь, животное и т.д. были отданы владельцам.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

**Ресурсы сети Интернет**

1) С# для новичков: развеиваем мифы и пишем простого чат-бота [Электронный ресуср]. https://skillbox.ru/media/code/si\_sharp\_dlya\_novichkov\_razveivaem\_mify/, дата 29.05.2020.

2) UML-диаграммы классов [Электронный ресурс]. https://prog-cpp.ru/uml-classes/, дата 29.05.2020.

3) Автономный установщик 4.7 платформа.NET Framework для Windows [Электронный ресурс]. https://support.microsoft.com/ru-ru/help/3186497/the-net-framework-4-7-offline-installer-for-windows, дата 29.05.2020.

4) Диаграммы потоков данных / Леоненков А. [Электронные ресурс]. https://it.wikireading.ru/5086, дата 29.05.2020.

5) Как работает бюро находок [Электронный ресурс]. https://expert.ru/2019/06/8/kak-rabotaet-byuro-nahodok/, дата 29.05.2020.

6) Обзор Windows Forms [Электронный ресурс]. https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/windows-forms-overview, дата 29.05.2020.

7) Принципы работы онлайнового бюро находок [Электронный ресурс]. https://kudago.com/all/news/printsipyi-rabotyi-onlajnovogo-byuro/, дата 29.05.2020.

8) Пошаговое руководство. Обработка данных (C#) [Электронный ресурс]. https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/sql/linq/walkthrough-manipulating-data-csharp, дата 29.05.2020.

9) Руководство по программированию на C# [Электронный ресурс]. https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/, дата 29.05.2020.

10) Что такое DFD (диаграммы потоков данных) [Электронный ресурс]. https://habr.com/ru/company/trinion/blog/340064/, дата 29.05.2020.

**Лист нормоконтроля**

|  |  |
| --- | --- |
|  | Автор работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |
|  | Нормоконтроль  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) (ФИО)  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |