МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПЕУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет очный   
Кафедра программной инженерии

Утверждаю

Заведующий кафедрой ПИ

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_ Н.В. Пацей\_

подпись инициалы и фамилия

“\_\_\_ ”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г.

**ЗАДАНИЕ**

**к курсовому проектированию**

**по дисциплине** "Объектно-ориентированное программирование"

|  |  |
| --- | --- |
| Специальность: 1-98 01 03 Программное обеспечение информационных технологий  Студент: Храмых В.О. | Группа: 6 |
| **Тема: Программное средство «Торговая площадка»** | |

**1. Срок сдачи студентом законченной работы**: "май 2019 г."

**2. Исходные данные к проекту:**

**2.1**. Функционально ПС должно:

* Управлять базой данных (добавление пользователей, администрирование, управление товарами продажи со стороны админа)
* Выбор товаров по определённым критериям
* Поиск по названию товара, названию фирмы
* Возможность настройки своей учётной записи из самого приложения

**2.2.** Программное средство должно быть выполнено с использованием ООП, взаимодействовать с базой данных, реализовано под ОС Windows. Отображение, бизнес логика и хранилище данных должны быть максимально независимы друг от друга для возможности расширения. Язык разработки С#. Управление программой должно быть интуитивно понятным и удобным. Использовать архитектурные и шаблоны проектирования.

**3. Примерное содержание расчетно-пояснительной записки**

(перечень вопросов подлежащих разработке)

* Введение
* Постановка задачи
* Обзор литературы и аналогов (алгоритмы решения, обзор литературы, прототипы)
* Проектирование программного средства (концепция, модель, структура, выполняемые функции и взаимосвязь всех компонентов, диаграммы UML)
* Разработка модели данных программного средства
* Руководство пользователя
* Тестирование программного средства
* Заключение
* Список используемых источников
* Приложения

**4. Форма представления выполненной курсовой работы:**

* + Теоретическая часть курсового проекта должны быть представлены в формате MS Word. Оформление записки должно быть согласно правилам.
  + Необходимые схемы, диаграммы и рисунки допускается делать в MS Office Visio, VS или других редакторах.
  + Листинги программы представляются частично в приложении.
  + К записке необходимо приложить CD (DVD), который должен содержать: пояснительную записку, листинги и инсталляцию проекта.

#### Календарный план

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта | Примечание |
| 1 | Введение | 19.02.2019 |  |
| 2 | Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования | 12.03.2019 |  |
| 3 | Анализ и проектирование архитектуры приложения (построение диаграмм, проектирование бизнес-слоя, представления и данных) | 26.03.2019 |  |
| 4 | Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса | 2.04.2019 |  |
| 5 | Кодирование программного средства | 23.04.2019 |  |
| 6 | Тестирования и отладка программного средства | 30.04.2019 |  |
| 7 | Оформление пояснительной записки | 7.05.2019 |  |
| 2 | Сдача проекта | 20.05.2019 |  |

**5. Дата выдачи задания** 21.01.2019

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Радиванович Д.А.*

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата и подпись студента)

Введение

Программное средство «Торговая площадка» – это программное средство, которое позволяет вести учёт предложений о продаже чего-либо (телефоны, ноутбуки, услуги и т.д.). Приложение имеет 3 типа пользователей: администратор, пользователь, модератор. Они имеют разные полномочия. Администраторы проверяют предложения, которые поступают на проверку перед публикацией. Так же они могут удалять предложения, которые уже неактуальны. Пользователи же могут следить за списком предложений, выполнять сортировку по дате, названию, цене. Так же будет реализована авторизация всех пользователей, т.е. перед тем, как предложить свой товар, нужно будет зарегистрироваться. Хранение всех пользователей будет осуществлено в базе данных, и при авторизации или регистрации будет непосредственное обращение к ней. Также вся информация о товаре будет хранится в базе данных.

В ходе использования это программы, пользователи смогут получать информацию о интересующих их товарах и продавцах (характеристика товара, номер телефона продавца). Язык разработки проекта – C#. При выполнении курсового проекта будут использованы принципы и приемы ООП. Также будут использоваться технологии Windows Presentation Foundation(WPF) и MS SQL.

# Аналитический обзор литературы

Для выполнения этого курсового проекта не обошлось без использования дополнительной литературы. Таковой является информация на сайте <https://metanit.com/>. На этом сайте имеется много информации, нужной для разработки этого курсового проекта. Так же использовалась информация с сайта <http://professorweb.ru/>.

Самый большое количество информации было получено с официального сайта <https://docs.microsoft.com/en-us/dotnet/csharp/>. Статьи с <https://habr.com/ru/> были также очень полезны.

В качестве аналога я выбрал сайт [https://catalog.onliner.by](https://catalog.onliner.by/), т.к. он максимально приближен к моей задумке. Краткий обзор этого сервиса представлен ниже.

Для посетителей Onliner.by доступен  [каталог](http://catalog.onliner.by/) товаров. Вот так выглядит страница с товарами (рисунок 1.1).

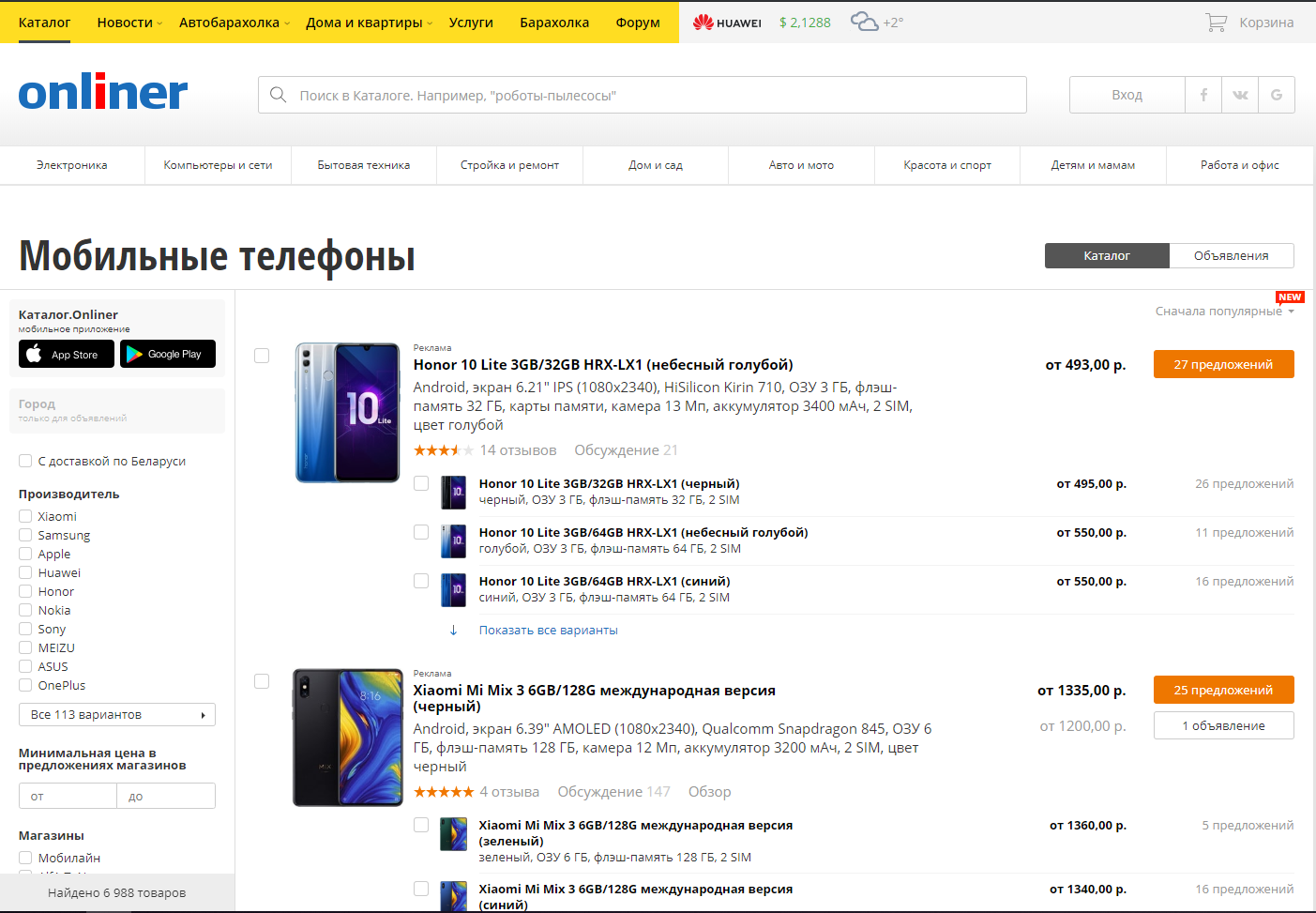


Рисунок 1.1 – раздел «Мобильные телефоны»

Все параметры для подбора товаров находятся в левой части экрана. Например, если речь идет о [телефонах](http://catalog.onliner.by/mobile), здесь с помощью checkbox можно выбрать среди наиболее популярных брендов, размеров и разрешений экрана, объемов оперативной и встроенной памяти и т. д. (рисунок 1.2).

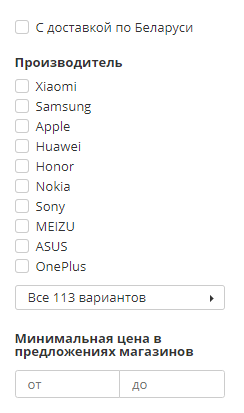


Рисунок 1.2 – параметры в каталоге

Если же вам нужно указать какой-то редкий параметр, его можно либо задать вручную, либо выбрать из списка. Дополнительные условия для поиска и всплывающие подсказки помогают в этом.

Так же здесь есть такая фишка – динамическое обновление контента по мере выбора тех или иных параметров. Если раньше для того, чтобы вывести на экран страницу товаров с заданными параметрами, надо было нажать кнопку «Подобрать», то теперь изменения происходят на лету. Пока вы маркируете checkbox, работаете со списками или вводите показатели вручную, данные на странице моментально обновляются в соответствии с заданными условиями (рисунок 1.3).

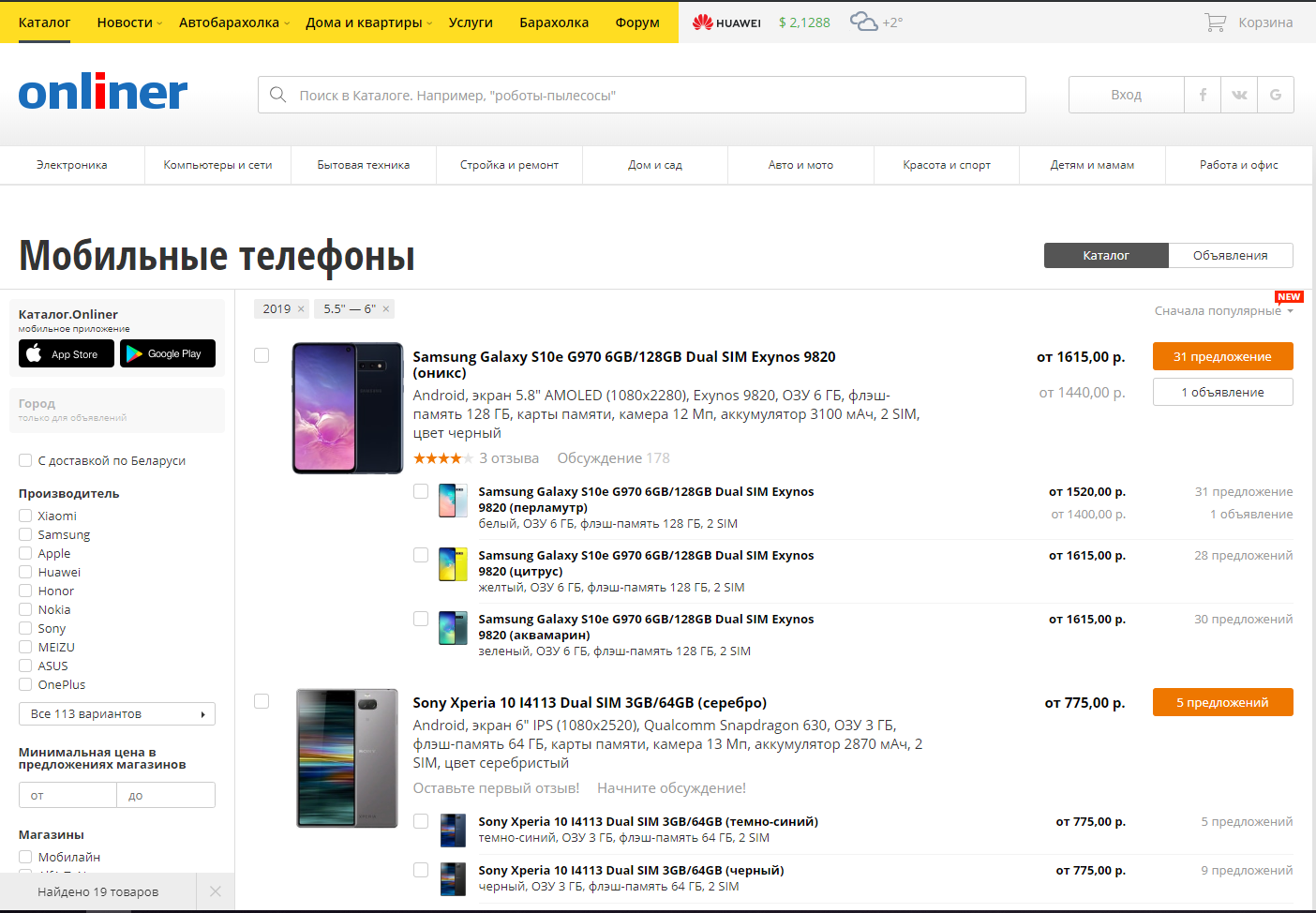
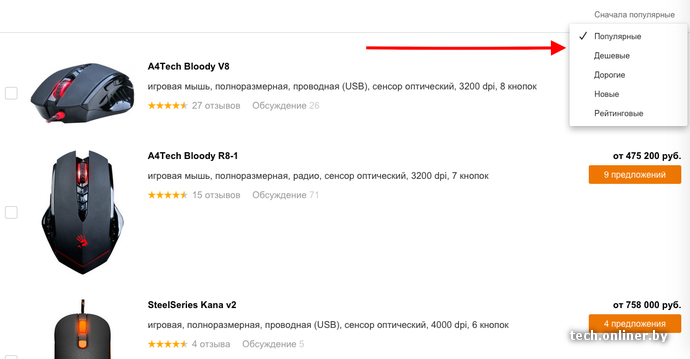


Рисунок 1.3 – Выбор параметров

Выбранные параметры выводятся вверху страницы с товарами. Чтобы убрать ненужные параметры, не обязательно искать их в левой панели и снимать «галочки». Достаточно «закрыть» их прямо наверху страницы.

Имеется сортировка по популярности, новизне, стоимости и рейтингу. Соответствующая опция расположена в правом верхнем углу страницы. По умолчанию задействован фильтр «Сначала популярные».

[[](https://content.onliner.by/news/2015/09/large/2ef92fb5f20c27cad94594c5162a4dba.png)](https://content.onliner.by/news/2015/09/large/2ef92fb5f20c27cad94594c5162a4dba.png)

Я решил сделать проект на эту тему, так как не видел ещё desktop-версии подобного приложения для Windows.

Этот сайт является образцом для моего проекта (<https://catalog.onliner.by/>).

# Анализ требований к программному средству и разработка функциональных требований

Задачей курсового проекта является разработка приложения, предназначенного осуществления торговли, а также ряда дополнительных операций: добавления товаров/услуг, редактирования, поиска, удаления информации. Информацию о товарах и услугах вводится непосредственно с самого приложения и проверяется администратором.

Программный продукт должен быть реализовать на объектно-ориентированном языке программирования C# с использованием технологии WPF. Windows Presentation Foundation (WPF) — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Графической технологией, лежащей в основе WPF, является DirectX, в отличие от Window Forms, где используется GDI/GDI+. Производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX.

Также существует урезанная версия CLR, называющаяся WPF/E, она же известна как Silverlight.

База данных была разработана в Microsoft SQL Server.

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире.

SQL Server характеризуется такими особенностями как:

* Производительность. SQL Server работает очень быстро.
* Надежность и безопасность. SQL Server предоставляет шифрование данных.
* Простота. С данной СУБД относительно легко работать и вести администрирование.

Для организации баз данных MS SQL Server использует реляционную модель, которая предполагает хранение данных в виде таблиц, каждая из которых состоит из строк и столбцов. Каждая строка хранит отдельный объект, а в столбцах размещаются атрибуты этого объекта.

Для взаимодействия с базой данных применяется язык SQL (Structured Query Language). Клиент (например, внешняя программа) отправляет запрос на языке SQL

должным образом интерпретирует и выполняет запрос, а затем посылает клиенту результат выполнения.

Основной используемый язык запросов —Transact-SQL — реализован на структурированном языке запросов (SQL) с расширениями.

Для осуществления связи между базой данных и приложением на C# необходим посредник, которым будет являться технология Entity Framework. Она основана на платформе .NET Framework и предназначена для работы с данными.

Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Эта технология предоставляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Entity Framework предполагает три возможных способа взаимодействия с базой данных:

* Database first: Entity Framework создает набор классов, которые отражают модель конкретной базы данных;
* Model first: сначала разработчик создает модель базы данных, по которой затем Entity Framework создает реальную базу данных на сервере;
* Code first: разработчик создает класс модели данных, которые будут храниться в базе данных, а затем Entity Framework по этой модели генерирует базу данных и ее таблицы.

В курсовом проектировании для разработки был выбран первый подход.

Приложение необходимо реализовать с использованием шаблона проектирования MVVM и с организацией взаимодействия с базой данных. В качестве клиентской части необходимо реализовать оконное приложение с использованием стандартных и иных библиотек. Приложение должно иметь возможность запускаться без использования интегрированных средств разработки.

Для разработанного продукта необходимо предусмотреть следующие задачи:

* Главное поле для отображения данных;
* Интуитивно понятный интерфейс;
* Механизм поиска товаром по названию, категории, цене;
* Возможность просмотра информации о товаре и о продавце.

По итогу работы над продуктом, его необходимо протестировать и отладить, проанализировать его возможности и, по необходимости, внести дополнения или улучшения проекта, с возможностью дальнейшего расширения проекта.

Программное средство предоставляет пользователю следующие функциональные возможности:

* Идентификация пользователей
* Размещение объявлений разного типа
* Сортировка объявлений по различным критериям
* Вход в приложение, использую социальные сети
* Хранение и отображение объявлений
* Управление своими объявлениями (редактирование, удаление)
* Сравнение объявлений (визуальное)

Модератор может всё то же самое, и плюс ко всему:

* Проверка поданных объявлений
* Удаление объявлений
* Просмотр информации о пользователях

Администратор может всё то же, что и модератор, и плюс ко всему:

* Удаление пользователей любого типа
* Изменение привилегий у любого пользователя

Наглядно это видно на UML-схеме, изображённой на рисунке 3.6



Рисунок 2.1 – «UML-диаграмма возможностей»

# Проектирование программного средства

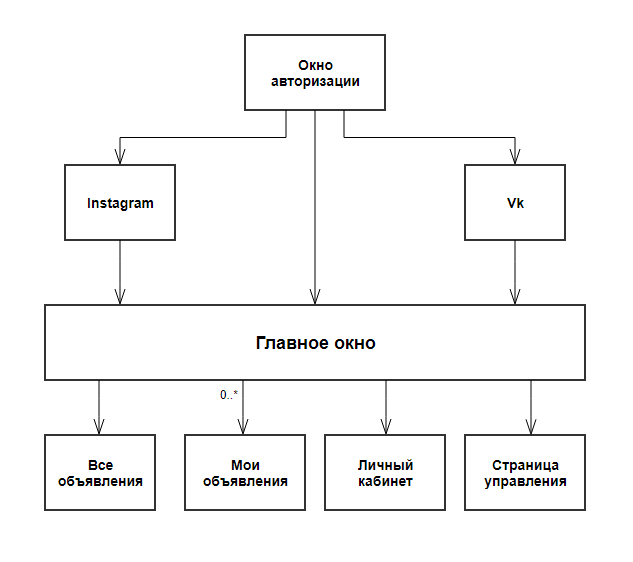


Рисунок 3.1 – «Авторизация и главное окно»

На рисунке 3.1 представлена общая схема работы приложения.

При запуске приложения у нас появляется окно авторизации, которое требует ввода логина и пароля для начала работы.

В случае, если вас нету своей учётной записи, её можно создать, нажав на кнопку «Регистрация» в окне авторизации. Вы будете направлены в окно регистрации, где вам нужно будет заполнить все поля. После успешной регистрации, вам нужно будет вернутся в окно авторизации и ввести ваши данные, указанные при регистрации. Так же поддерживается вход через социальные сети «ВКонтакте» и «Instagram».

Нажав на соответствующие кнопки в окне авторизации, у вас появится окно, где нужно будет ввести данные для входа в учётную запись, через которую производится вход.

После ввода логина и пароля в соответствующих полях, проверяется правильность ввода данных.

В случае неверного ввода данных, выведется соответствующее сообщение. В ином случае окно авторизации закроется и откроется главное окно приложения.

В главном окне работа идём в страничном режиме.

Рассмотрим каждую страницу в отдельности.

Страница «Все объявления» (рисунок 3.2).

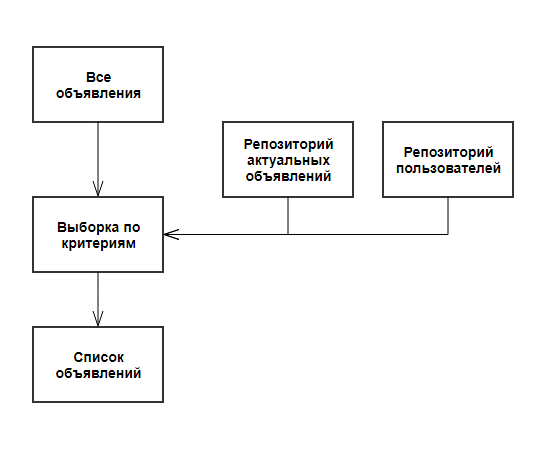


Рисунок 3.2 – «Все объявления»

Эта страница содержит список всех актуальных объявлений.

Так же она содержит механизм поиска объявления по определённым критериям.

Осуществить поиск можно по таким критериям:

* Слово в названии или описании
* Продавец
* Цена
* Регион
* Категорий
* Комбинации вышеперечисленных

На основе критериев поиска, формируется результирующий набор При нажатии мы видим более подробную информацию о объявлении и получаем возможность его закрепления, чтобы была возможность сравнить несколько объявлений между собой.

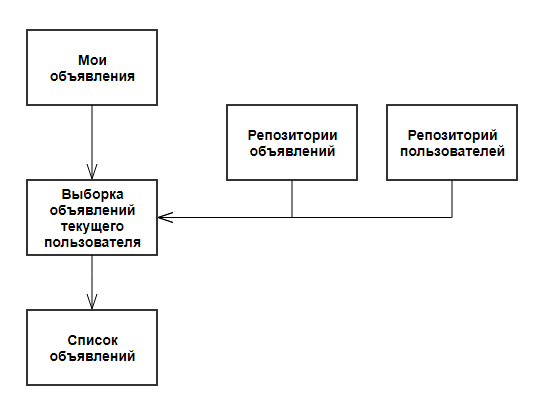


Рисунок 3.3 – «Мои объявления»

На рисунке 3.3 показана схема работы страницы «Мои объявления».

На этой странице ведётся редактирование, удаление, просмотр только тех объявлений, которые размещал вошедший пользователь.

Есть переключатель, который позволяет переключатся между объявлениями, которые ещё не прошли проверку и теми, которые уже находятся в разделе «Актуальные».

Редактирование происходит в отдельном окне, которое появляется при нажатии на соответствующую кнопку. В этом окне есть ограничения и коррекция недопустимых значений. На название отводится от 5 до 50 символов, а на описание от 10 до 1000 символов. Так же нельзя установить отрицательную цену т.к. она, при некорректно введённых значениях, будет обращаться в нуль или в то значение, которое было изначально.

При просмотре своего объявления, открывается окно, в котором выводится вся информация о данном объявлении.

Если происходит изменение объявления из раздела «Актуальные», то после изменения, оно автоматически попадает в раздел «Непроверенные» и становится невидимым для других пользователей.

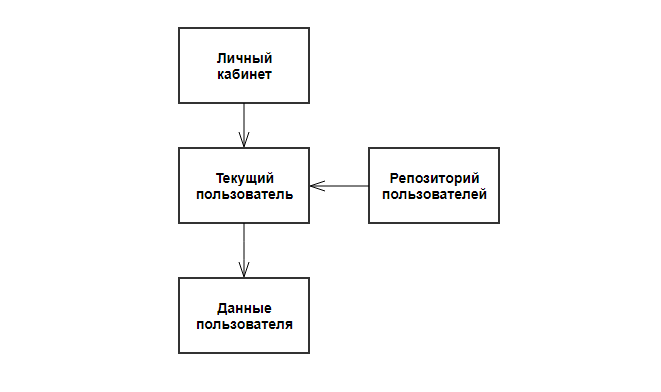


Рисунок 3.4 – «Личный кабинет»

На рисунке 3.4 показана схема страницы «Личный кабинет».

На этой странице можно редактировать свои данные, такие как:

* Имя
* Фамилия
* Mail
* Телефон
* Описание (о себе)
* Картинка профиля

Под картинкой профиля находится кнопка «Загрузить», нажатие на которую приводит к открытию окна с выбором файла картинки профиля, которую вы хотите установить в качестве основой.

На этой странице есть 2 кнопки: «Сохранить изменения», «Удалить свой профиль».

При нажатии на 1-ю кнопку происходит обновление информации о текущем пользователе в базе данных.

При на нажатии на 2-ю кнопку – удаление всех объявлений текущего пользователя, последующее удаление самого профиля и перенаправление на окно авторизации.

Так же на эту страницу можно попасть, нажав на элемент, находящийся в верху самого окна чуть правее середины, и содержащий имя и фамилию текущего пользователя, а также инициалы.

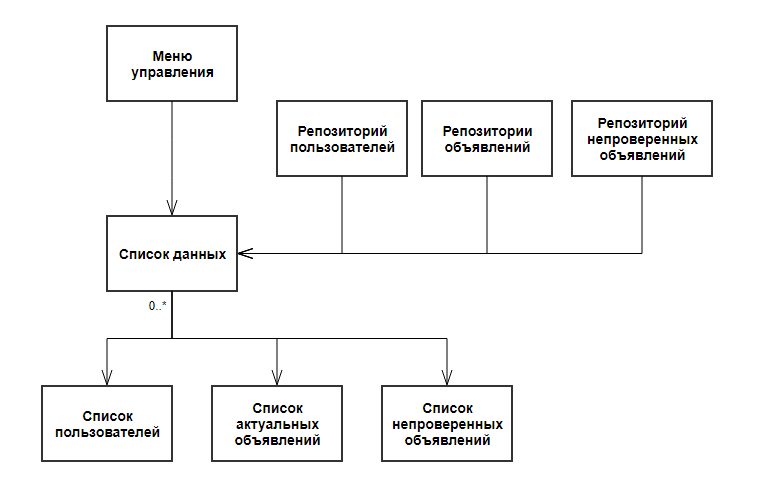


Рисунок 3.5 – «Меню управления»

В этом окне у нас производится администрирование нашего приложения.

Всего есть 3 типа привилегий:

* Пользователь
* Модератор
* Администратор

Пользователь не имеет доступ к этой странице. При попытке зайти в это меню, у него будет выведено сообщение о недостатке прав для доступа к этой странице.

Модератор имеет доступ к этой странице и имеет такие возможности:

* Проверка всех объявлений
* Удаление объявлений как из разряда «Актуальные», так и из разряда «Непроверенные»
* Просмотр информации о всех пользователях
* Одобрение и перенос объявления в «Актуальные»

Администратор имеет все те же права что и модератор, а также возможность назначать и изменять привилегии пользователей и удалять пользователей из приложения.

Структура проекта представлена на рисунке 3.7

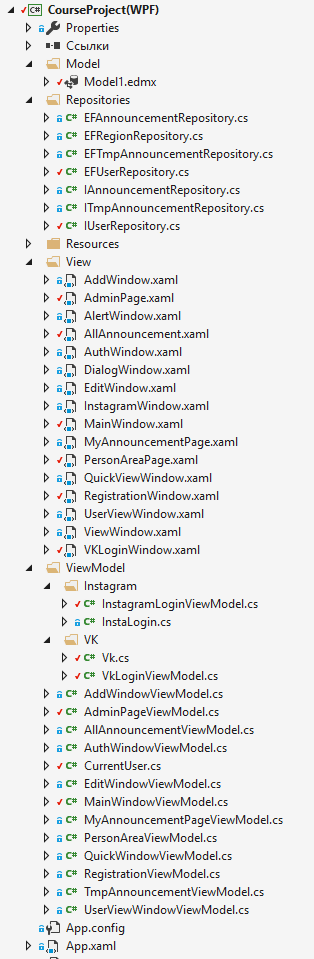


Рисунок 3.7 – «Структура проекта»

В таблице 3.8 описана структура проекта по пакетам классов.

В этой таблице приведены основные логические составляющие пакеты, которые используются в работе приложения.

### Таблица 3.8 – Описание структурных пакетов проекта

|  |  |
| --- | --- |
| Имя пакета | Описание |
| Model | Здесь описаны модели, с которыми происходит вся работа в приложении:   * Пользователь * Актуальное объявление * Непроверенное объявление |
| Repositories | Содержит классы и интерфейсы для работы с данными БД. Они являются связующим элементом между данными в БД и теми данными, с которыми работает приложение |
| Resources | Здесь находятся файлы, в которых описаны стили для элементов управления |
| View | Содержит все представления, которые позволяют пользователю работать с приложением. Они описывают графическую составляющую приложения, через которую связываются все манипуляции с данными |
| ViewModel | Содержит логику, которая позволяет получить данные при помощи View, обработать их, использую при этом Model, после чего передать в базу данных. |
| App.config | Файл конфигурации приложения |
| App.xaml | Основная задача данного файла состоит в определении ресурсов, общих для приложения |

С помощью этой таблицы, понять общую структуру проекта не составляет никакого труда.

Диаграмма в UML — это графическое представление набора элементов, изображаемое чаще всего в виде связанного графа с вершинами (сущностями) и ребрами (отношениями). Диаграммы рисуют для визуализации. Основная цель диаграмм — визуализация разрабатываемой системы с разных точек зрения.

Диаграмма классов для этого приложения представлена на рисунке 3.9

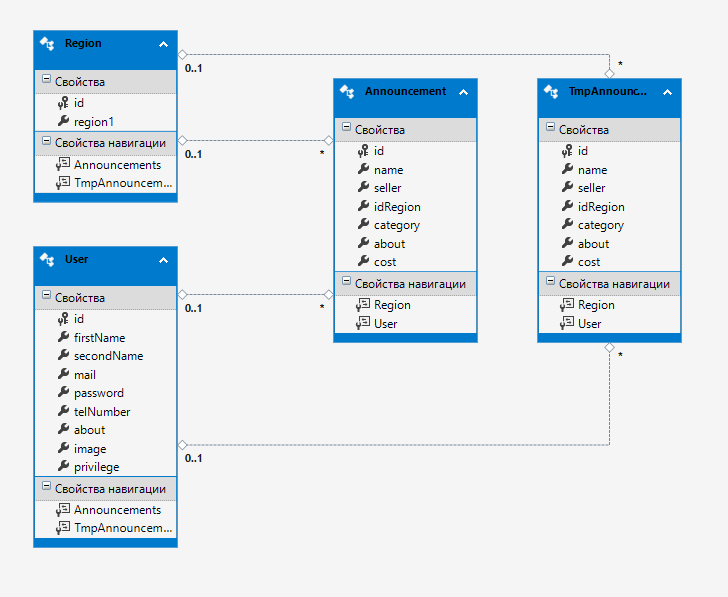


Рисунок 3.9 – «Диаграмма классов»

Как показано на рисунке 3.9, в вершине иерархии стоят классы Announcement и TmpAnnouncement, который характеризует актуальное и непроверенное объявление соответственно. Класс Region характеризует все регионы, в которых может быть размещено объявление. Такая сущность как User, характеризует пользователя с любым уровнем доступа. Announcement и TmpAnnouncement связаны с User по полю seller и id соответственно. Announcement и TmpAnnouncement связаны с Region по полю idRegion и id соответственно. TmpAnnouncement и Announcement между собой никак не связаны. Хоть у них и одинаковые поля, разнесены они по разным таблицам для того, чтобы не сливать воедино объявления всех типов.