МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра программной инженерии

Специальность 6-05-0612-01 Программная инженерия

Направление специальности 1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет приложений)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТ:**

по дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Тема Программное средство «Конфигуратор ПК»

Исполнитель

студент (ка) 2 курса группы 9 Бондарик Никита Дмитриевич

(Ф.И.О.)

Руководитель работы ассистент Ромыш А.С.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Смелов В.В.

(подпись)

Минск 2025

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра программной инженерии

Утверждаю

Заведующий кафедрой ПИ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В.Смелов

подпись инициалы и фамилия

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г.

**ЗАДАНИЕ**

**к курсовому проектированию**

**по дисциплине** "Объектно-ориентированные технологии программирования и стандарты проектирования"

|  |  |
| --- | --- |
| Специальность: 6-05-0612-01 Программная инженерия  Студент: Бондарик Никита Дмитриевич | Группа:\_\_9\_\_ |
| **Тема: Программное средство «Конфигуратор ПК»** | |

**1. Срок сдачи студентом законченной работы: "21 мая 2025 г."**

**2. Исходные данные к проекту:**

2.1. Функционально ПС поддерживает:

* Функции администратора:
  + Добавление/удаление/редактирование комплектующих.
  + Модерация объявлений.
  + Просмотр активных заказов.
  + Просмотр истории заказов.
  + Блокировка и разблокировка аккаунтов.
  + Генерация отчетов.
* Функции клиента:
  + Регистрация и авторизация.
  + Просмотр списка доступных комплектующих.
  + Поиск и фильтрация комплектующих.
  + Просмотр активной заказов, истории своих заказов.
  + Оформление заказа.
  + Возможность сохранения сборок, добавления в избранные.
  + Возможность оставить отзыв и оценку.
  + Личный кабинет.

2.2. При выполнении курсового проекта необходимо использовать принципы проектирования ООП. Приложение разрабатывается под ОС Windows и представляет собой настольное приложение (desktop). Отображение, бизнес логика должны быть максимально независимы друг от друга для возможности расширения. Диаграммы вариантов использования, классов реализации задачи, взаимодействия разработать на основе UML. Язык разработки проекта – C#. Управление программой должно быть интуитивно понятным и удобным. При разработке использовать несколько наиболее подходящих шаблонов проектирования ПО.

**3. Содержание расчетно-пояснительной записки**

* Введение
* Постановка задачи и обзор литературы
* Проектирование архитектуры проекта
* Разработка функциональной модели и модели данных ПС
* Тестирование
* Заключение
* Список используемых источников
* Приложения

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

* + Теоретическая часть курсового проекта должны быть представлены в формате docx. Оформление записки должно быть согласно выданным правилам.
  + Листинги программы представляются в приложении.
  + Пояснительную записку, листинги, проект (инсталляцию проекта) необходимо загрузить диск, указанный преподавателем.

#### Календарный план

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта | Примечание |
| 1 | Введение | 12.03.2025 |  |
| 2 | Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования | 19.03.2025 |  |
| 3 | Анализ и проектирование архитектуры приложения (построение диаграмм, проектирование бизнес-слоя, представления и данных) | 31.03.2025 |  |
| 4 | Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса | 09.04.2025 |  |
| 5 | Кодирование программного средства | 23.04.2025 |  |
| 6 | Тестирования и отладка программного средства | 07.05.2025 |  |
| 7 | Оформление пояснительной записки | 14.05.2025 |  |
| 8 | Защита проекта | 21.05.2025 |  |

**5. Дата выдачи задания \_\_\_\_**05.03.2025**\_\_\_\_**

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *асс. Ромыш А.С.*

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Бондарик Н.Д.*

(дата и подпись студента)

Оглавление

[ВВЕДЕНИЕ 6](#_Toc198873346)

[1. Аналитический обзор прототипов и литературных источников 7](#_Toc198873347)

[1 .1Анализ прототипов 7](#_Toc198873348)

[1.1.1. Интернет-ресурс «kst.by» 7](#_Toc198873349)

[1.1.2. Интернет-ресурс «multimart.by» 8](#_Toc198873350)

[1.1.3. Интернет-ресурс «leopanda.by» 9](#_Toc198873351)

[1.1.4. Интернет-ресурс «x-core.by» 10](#_Toc198873352)

[1.2. Постановка задачи 11](#_Toc198873353)

[2. Анализ требований к программному средству и разработка функциональных требований 13](#_Toc198873354)

[2.1 Определение требований к программному средству 13](#_Toc198873355)

[2.2 Описание средств разработки 13](#_Toc198873356)

[2.3 Диаграмма вариантов использования 14](#_Toc198873357)

[3 Проектирование программного средства 15](#_Toc198873358)

[3.1 Общая структура приложения 15](#_Toc198873359)

[3.2 Взаимоотношение между классами 16](#_Toc198873360)

[3.3 Проектирование логической структуры базы данных 16](#_Toc198873361)

[3.4 Проектирование архитектуры приложения 19](#_Toc198873362)

[3.5 Проектирование архитектуры окон 21](#_Toc198873363)

[3.6 Схема работы приложения 24](#_Toc198873364)

[4 Реализация программного средства 27](#_Toc198873365)

[4.1 Основные классы программного средства 27](#_Toc198873366)

[4.2 Регистрация и авторизация 27](#_Toc198873367)

[4.3 Управление профилем пользователя 28](#_Toc198873368)

[4.4 Просмотр и управление услугами 28](#_Toc198873369)

[4.5 Блокировка и разблокировка аккаунтов 28](#_Toc198873370)

[4.6 Работа с отзывами 28](#_Toc198873371)

[4.7 Вспомогательные классы и сервисы 28](#_Toc198873372)

[4.8 Разделение на слои MVVM 29](#_Toc198873373)

[5. Тестирование, проверка работоспособности и анализ 30](#_Toc198873374)

[5.1 Тестирование, проверка работоспособности 30](#_Toc198873375)

[5.2 Вывод по главе 31](#_Toc198873376)

[6. Руководство по установке и использованию 32](#_Toc198873377)

[6.1 Использование программного средства 32](#_Toc198873378)

[Список использованной литературы 35](#_Toc198873379)

[Приложение Б 37](#_Toc198873380)

[Приложение В 39](#_Toc198873381)

# **ВВЕДЕНИЕ**

В современном мире персональные компьютеры играют ключевую роль как в профессиональной деятельности, так и в повседневной жизни. Сборка ПК позволяет пользователям подобрать оптимальные компоненты под свои задачи, будь то работа с графикой, программирование, игровая индустрия или офисные приложения.

Анализ конкурентов помогает нам понять, что у многих приложений есть свои преимущества, а также есть и свои недостатки. В условиях растущей конкуренции и высоких ожиданий клиентов возникает необходимость в новых более удобных решениях, которые позволят пользователям решать свои задачи.

При разработке приложения учитываются несколько основных принципов, которые помогут достичь намеченных целей. Во-первых, важно, чтобы интерфейс приложения был простым и понятным, что позволит как опытным механикам, так и обычным пользователям быстро находить нужные автозапчасти. Во-вторых, приложение должно обеспечивать быстрый и удобный процесс оформления заказов и оплаты, чтобы пользователи могли совершать покупки с минимальными усилиями. В-третьих, необходимо создать эффективные инструменты для общения между клиентами и администраторами, чтобы они могли легко обмениваться информацией и оставаться на связи.

Дополнительно, приложение должно включать функционал для отслеживания статуса заказа, а также систему отзывов и оценок, позволяющую клиентам делиться своим опытом и помогать другим пользователям принимать обоснованные решения.

Данный курсовой проект направлен на разработку программного обеспечения — конфигуратора ПК, который предоставит пользователям удобный инструмент для подбора и виртуальной сборки компьютера. Пользователи смогут:

– выбирать компоненты из базы данных (процессор, видеокарта, материнская плата и т. д.);

– просматривать итоговую стоимость конфигурации;

– сохранять и редактировать свои сборки.

После разработки и запуска приложения с указанными выше принципами и требованиями, пользователи получат возможность легко и быстро собирать ПК и оформлять заказы. Это, в свою очередь, будет способствовать росту удовлетворенности клиентов и улучшению качества обслуживания.

# **Аналитический обзор прототипов и литературных источников**

## **Анализ прототипов**

Курсовой проект представляет из себя разработку приложения конфигуратора компьютера. В этой сфере существует множество аналогичных решений, которые предоставляют пользователям возможность конфигурации компьютеров. В ходе выполнения курсового проекта были проанализированы цели и задачи, а также рассмотрены аналогичные примеры их решений. На основании анализа всех достоинств и недостатков данных альтернативных решений были сформулированы требования к данному программному средству.

# **1.1.1. Интернет-ресурс «kst.by»**

KST представляет собой сайт для конфигурации и покупки компьютеров. Он предоставляет разнообразный функционал, который помогает пользователям ознакомиться с комплектующими, а также заказать их. Интерфейс сайта представлен на рисунке 1.1.

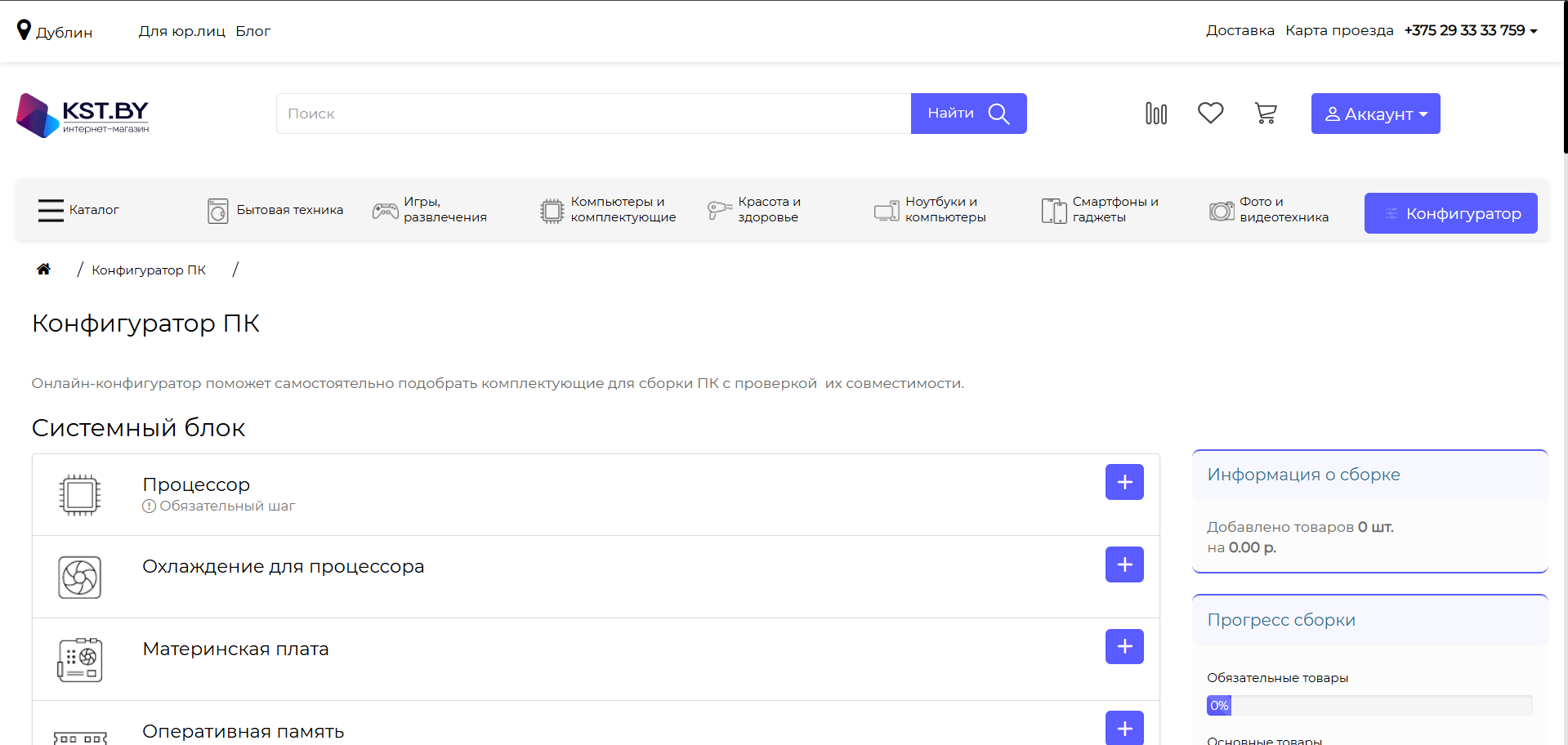


Рисунок 1.1 – Интерфейс сайта kst.by

На этом сайте пользователь может просмотреть контактную информацию о магазине и ознакомиться с графиком работы. Также на сайте представлены и другие товары электроники. Помимо этого пользователь может узнать основные события, связанные с данной магазином на главной странице.

Главной целью этого сайта является ознакомление с общей информацией, а также возможность заказа и покупки. Товары можно купить на месте, или заказать доставку. Сам конфигуратор представлен довольно удобно. Кроме комплектующих можно также выбрать и стороннюю компьютерную периферию, такую как клавиатуры, мониторы и так далее.

Одним из плюсов данного сайта является возможность просмотра отзывов и оставления отзыва, что складывает определённое мнение о магазине.

Также на сайте приведена информация о местоположении оффлайн магазина. А если у клиента возникнут вопросы и он захочет связаться со с поддержкой, на сайте в разделе контакты приведены номер телефона, электронная почта и ссылки на социальные сети для связи.

Также на сайте есть возможность регистрации и сохранение корзины в личном кабинете.

Сайт неплохой, но имеет свои недостатки. На сайте плоха настроена оптимизация при уменьшении страницы. Также отсутствует бургер-меню.

# **1.1.2. Интернет-ресурс «multimart.by»**

Сайт multimart представляет собой магазин электро-техники в Минске. Сайт содержит широкий функционал, с помощью которого пользователь может ознакомиться со всех необходимой информацией о товарах, а также о предоставляемых им услугах. Интерфейс сайта представлена на рисунке 1.2.

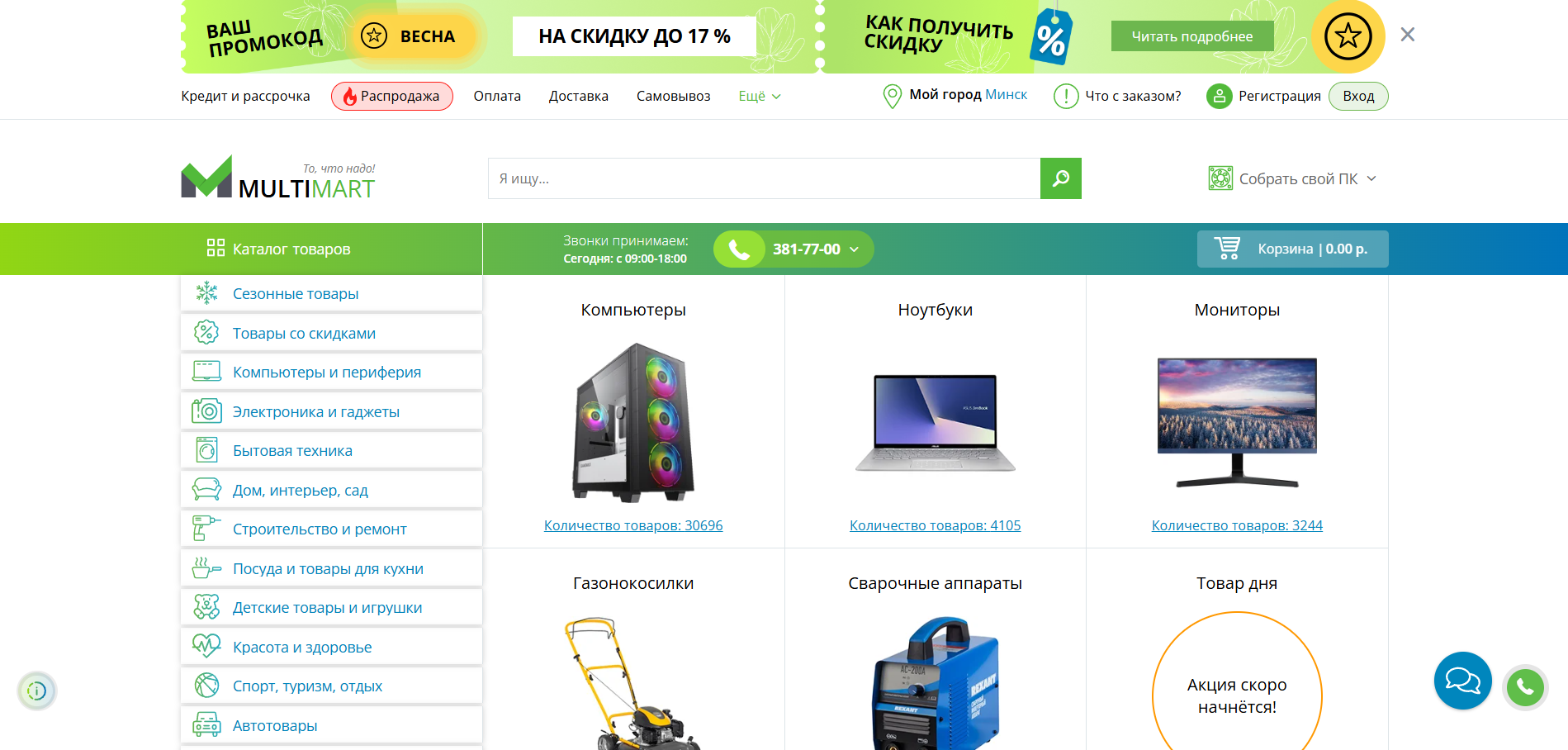


Рисунок 1.2 – Интерфейс сайта multimart.by

Также как и сайт kst.by, данный сайт имеет большой выбор товаров, а также конфигуратор для сборки компьютеров с понятным для новичков интерфейсом.

Помимо этого, данный сайт содержит небольшие подсказки о том, что нужно выбрать для полной сборки компьютера и на сколько процентов завершена конфигурация. Также сайт содержит информацию о акциях. Присутствует форма, после заполнения которой, пользователю помогут подобрать подходящий для него компьютер.

На сайте нету вкладки с отзывами, но их можно увидеть в социальных сетях, которые можно обнаружить в футтере страницы.

Интернет магазин кроме комплектующих предоставляет возможность покупки и другой электро-техники.

Как и сайт kst.by данный сайт позволяет совершить покупку как онлайн, так и оффлайн. Есть контактные адреса и номера телефонов.

Также присутствуют ссылки на социальные сети.

Также есть личный кабинет с такими же возможностями как и сайт kst.by.

Сайт неплохой, однако имеются свои минусы. Также как и у kst.by на сайте плохая оптимизация при уменьшении страницы, а также окна наползают друг на друга.

# **1.1.3. Интернет-ресурс «leopanda.by»**

Сайт Leopanda.by представляет собой сайт компании, занимающейся продажей и сборкой компьютеров. Данный сайт предоставляет широкий и разнообразный функционал. Интерфейс сайта представлен на рисунке 1.3, на котором показана главная страница сайта, содержащая информацию об акционных предложениях и популярных товарах.

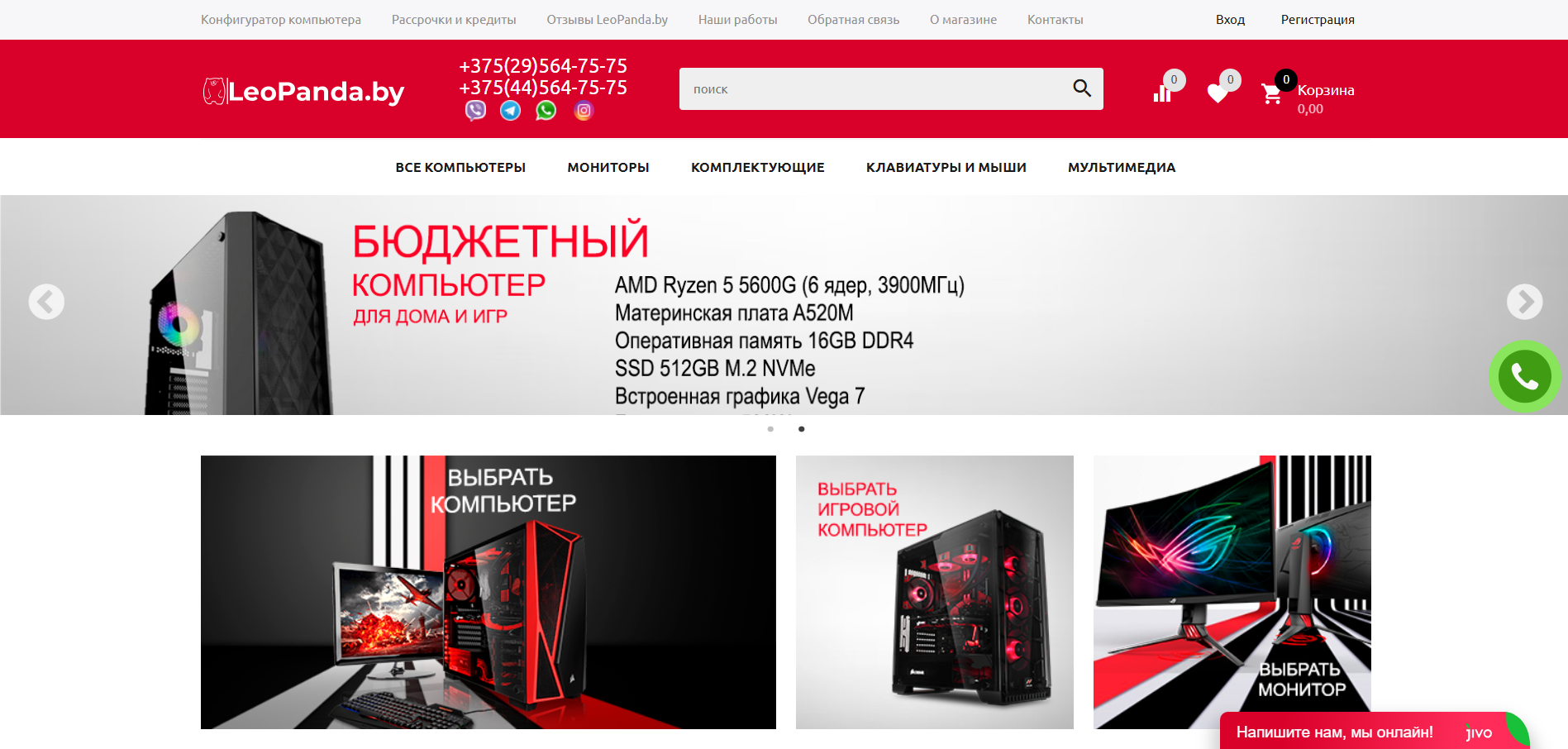


Рисунок 1.3 – Интерфейс сайта leopanda.by

Сайт имеет удобное меню, а также предоставляет информацию о компании. Весь сайт оснащен качественными изображениями и описаниями товаров, чтобы пользователь мог детально ознакомиться с предлагаемыми решениями.

На сайте доступна информация о каждом товаре: процессорах, видеокартах, готовых сборках и других комплектующих. Также представлены услуги по сборке и апгрейду компьютеров, чего не наблюдалось в предыдущих аналогах. Это, несомненно, большой плюс для компании.

Помимо покупки готовых решений, сайт предлагает возможность самостоятельно собрать ПК с помощью конфигуратора. Прилагается информация о совместимости компонентов и итоговой стоимости сборки.

Форма для заказа содержит поля для ввода имени, номера телефона, выбора способа доставки и оплаты. Также есть возможность оставить комментарий к заказу.

Сайт предоставляет информацию о местоположении магазинов, а также контакты для обратной связи. Указаны номера телефонов, электронная почта, а также ссылки на социальные сети: VK, Telegram, YouTube.

Несмотря на свои плюсы, такие как наличие конфигуратора и услуг по сборке ПК, сайт также имеет и минусы. Среди них, как и в предыдущих аналогах, отсутствие личного кабинета для отслеживания заказов. Также на сайте нет системы накопительных скидок, что является небольшим недостатком.

# **1.1.4. Интернет-ресурс «x-core.by»**

Сайт X-Core.by представляет собой сайт компании, специализирующейся на продаже компьютерных комплектующих и готовых сборок. Данный сайт предлагает разносторонний функционал для подбора и приобретения компьютерной техники. Интерфейс сайта представлен на рисунке 1.4, где отображена главная страница с акционными предложениями и основными категориями товаров.

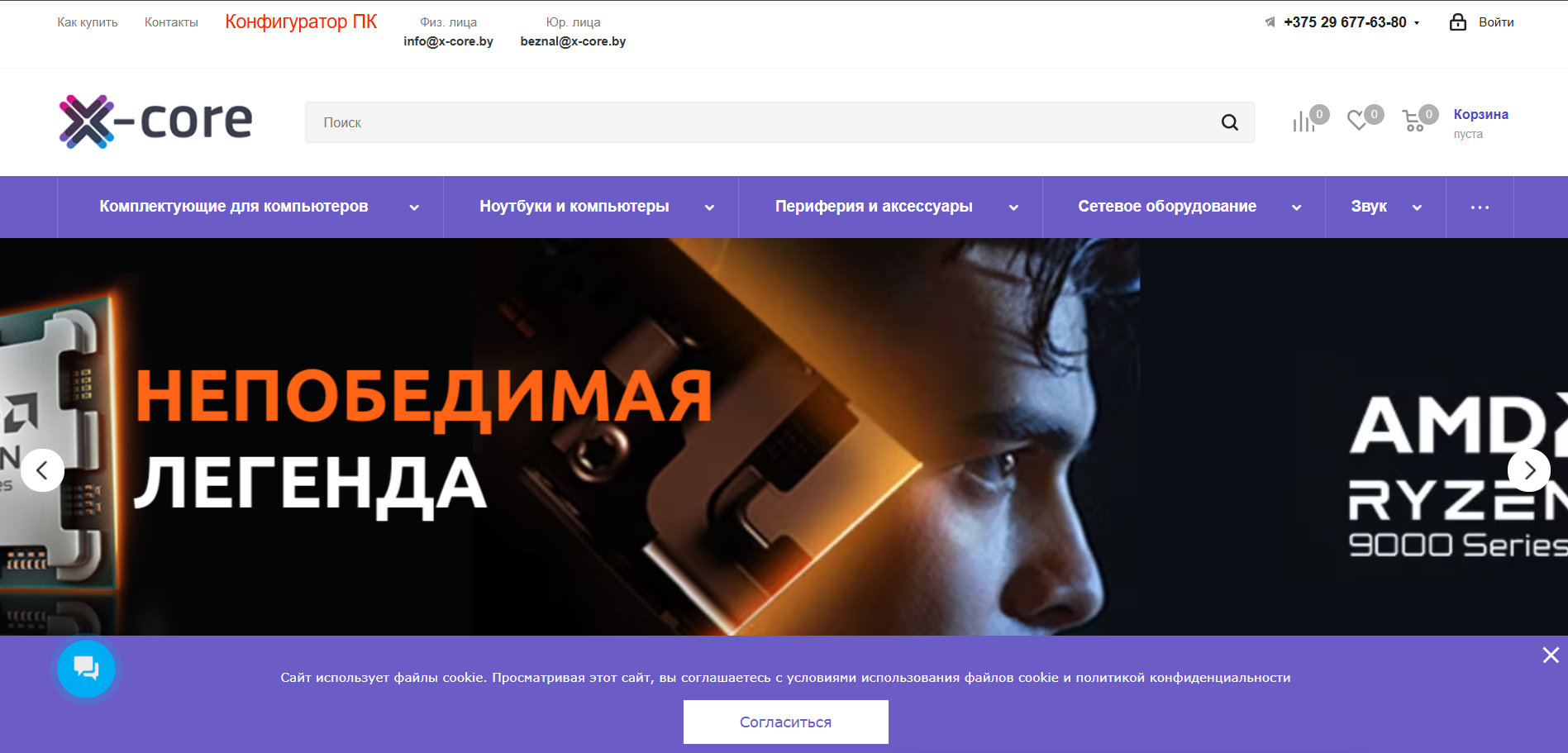


Рисунок 1.4 – Интерфейс сайта x-core.by

Сайт обладает продуманным меню и содержит подробную информацию о компании. Все разделы сопровождаются качественными фотографиями товаров и детальными техническими характеристиками, что позволяет покупателю получить полное представление о продукции.

На сайте представлен полный ассортимент комплектующих: процессоры, видеокарты, материнские платы, оперативная память и другие компоненты. Особого внимания заслуживает наличие услуг по профессиональной сборке и тестированию компьютеров, что выгодно отличает этот сайт от аналогов.

Помимо стандартного каталога, сайт предоставляет возможность самостоятельной сборки ПК через удобный конфигуратор. Система автоматически проверяет совместимость выбранных компонентов и рассчитывает итоговую стоимость конфигурации.

Форма оформления заказа включает стандартные поля: имя, контактный телефон, способ доставки и оплаты. Дополнительно предусмотрено поле для особых пожеланий к заказу.

Сайт содержит полную контактную информацию: адреса магазинов, телефоны для связи, электронную почту. Также указаны ссылки на официальные страницы в социальных сетях: VK, Telegram, YouTube.

Несмотря на очевидные преимущества, такие как удобный конфигуратор и услуги сборки, сайт имеет некоторые недостатки. В частности, отсутствует возможность создания личного кабинета для отслеживания заказов, а также не реализована система накопительных скидок для постоянных клиентов.

# **1.2. Постановка задачи**

Анализ существующих аналогов позволяет выявить сильные и слабые стороны альтернативных решений, а также сформировать четкие требования к разрабатываемому программному обеспечению.

**Конфигуратор ПК должен обеспечивать следующий функционал:**

#### **Функции администратора:**

– Регистрация и авторизация в системе

– Управление каталогом комплектующих (добавление, редактирование, удаление)

– Настройка правил совместимости компонентов

– Управление акциями и специальными предложениями

– Просмотр и обработка заказов пользователей

#### **Функции пользователя:**

– Регистрация и авторизация в системе

– Просмотр и оставление отзывов о товарах

– Выбор способа оплаты и доставки

– Просмотр характеристик комплектующих

– Использование конфигуратора для сборки ПК:

– Подбор компонентов по категориям (процессор, видеокарта и т. д.)

– Автоматическая проверка совместимости выбранных компонентов

– Просмотр итоговой стоимости сборки

– Возможность сохранения конфигурации

– Оформление заказа через заполнение формы

Данный функционал позволит пользователям удобно и быстро собирать персональные компьютеры под свои нужды, а администраторам — эффективно управлять ассортиментом и заказами.

# **2.** **Анализ требований к программному средству и разработка функциональных требований**

Анализ требований — это процесс сбора требований к программному обеспечению, их систематизации, документирования, анализа, выявления противоречий, неполноты, разрешения конфликтов в процессе разработки программного обеспечения.

Основная цель анализа требований в проектах заключается в получении максимально полной информации о клиенте, его задачах и особенностях проекта. Анализ требований помогает определить границы проекта, выявить возможные риски и ключевые требования, как методологические, так и технологические. На этом этапе формулируются цели и задачи проекта, а также определяются критические факторы успеха, которые будут использоваться для оценки результатов внедрения. Определение и описание требований является важным этапом, так как эти требования оказывают влияние на все последующие этапы проекта и определяют его успех.

# **2.1 Определение требований к программному средству**

Функционально ПС должно выполнять следующие задачи:

– регистрация и авторизация пользователей;

– сохранение информации в базе данных;

– для администратора добавление/изменение/удаление пользователей, комплектующих ПК;

– для администратора блокировка пользователей;

– для пользователя изменение личного кабинета;

– для пользователя просмотр информации о комплектующих;

# **2.2 Описание средств разработки**

При разработке приложения были использованы:

– интегрированная среда разработки Microsoft Visual Studio 2022;

– программная платформа .NET Framework;

– язык программирования C#;

– расширяемый язык разметки XAML;

– технология WPF;

– технология Entity Framework;

– SQLite.

В качестве интерфейса прикладного программирования был выбран обширный API-интерфейс – Windows Presentation Foundation (WPF), предназначенный для создания настольных программ с графически насыщенным пользовательским интерфейсом. В основе графической технологии WPF лежит мощная инфраструктура, основанная на DirectX. Это является одним из основных отличий WPF от более ранней технологии создания пользовательских интерфейсов – Windows Forms.

Для работы с WPF использовался объектно-ориентированный язык программирования с С-подобным синтаксисом – С#, разработанный для создания приложений на платформе Microsoft .NET Framework.

Есть различные механизмы работы с БД в С#. Т.к. моя база данных небольшая, мой выбор пал на Entity Framework. Entity Framework предоставляет простой способ доступа к данным в .Net Framework.

# **2.3 Диаграмма вариантов использования**

Диаграмма вариантов использования (Use Case Diagram) является графическим средством моделирования, которое позволяет представить функциональность системы или программного продукта с точки зрения ее пользователей. Она помогает идентифицировать и описать различные сценарии использования системы, включая взаимодействие между актерами (пользователями) и самой системой.

Диаграмма вариантов использования состоит из актеров, вариантов использования (Use Case) и связей между ними. Актеры представляют различных пользователей или внешние системы, которые взаимодействуют с программой. Варианты использования представляют собой функциональные возможности системы, которые могут быть выполнены пользователями или другими актерами.

Преимущества использования диаграммы вариантов использования включают:

Визуализацию функциональности системы: Диаграмма вариантов использования помогает визуализировать все возможные сценарии использования системы, что помогает лучше понять ее функциональность и взаимодействие с пользователями.

Определение требований: Диаграмма вариантов использования помогает идентифицировать и описать функциональные требования к системе на основе сценариев использования.

Улучшение коммуникации: Диаграмма вариантов использования служит средством коммуникации между разработчиками, заказчиками и другими заинтересованными сторонами, позволяя им лучше понять ожидания от данного приложения.

UML-диаграмма вариантов использования для программного средства «HireVich» представлена в приложении А.

# **3 Проектирование программного средства**

# **3.1 Общая структура приложения**

Общая структура проекта Windows Presentation Foundation (WPF) – это организация файлов и папок в рамках проекта WPF. Структура разрабатываемого проекта «PCConfigurator» представлена на рисунке 3.1.

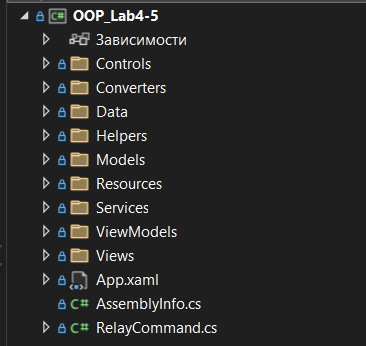


Рисунок 3.1 — Структура проекта

Описание структуры основных папок и файлов проекта и библиотеки классов представлено в таблице 3.1.

Таблица 3.1 — Описание структуры папок и файлов проекта

|  |  |
| --- | --- |
| Имя файла/папки | Содержание |
| App.xaml | Определение глобальных ресурсов |
| Папка Images | Директория, содержащая подпапки с всем изображениями: иконки, логотипы, изображения профилей пользователей. |
| Папка Control | Директория, содержащая пользовательские элементы управления. |
| Папка Converters | Конвертеры для преобразования значений,например,видимости,текста кнопок и тем. |

Продолжение таблицы 3.1

|  |  |
| --- | --- |
| Имя файла/папки | Содержание |
| Папка Data | Директория с классами для работы с базой данных, включая контекст PCComponentsContext и сущности, такие как User, Build, ComponentEntity. |
| Папка Helpers | Директория с вспомогательными классами, например, анимации изменения Margin и проверки вводимых данных. |
| Папка Resources | Директория с файлами локализации (Strings.en-US.xaml, Strings.ru-RU.xaml) и стилями (Styles.xaml, LightStyles.xaml). |
| Папка Services | Содержит классы для работы с сервисами, например AuthService для управления авторизацией пользователей. |
| Папка Views | Директория Views содержит несколько окон (Window), папку Pages, в которой находятся страницы (Page) для навигации внутри приложения и папку Controls, в которой находятся ЭУ созданные мной. |
| AssemblyInfo.cs | Файл в котором находится информация компиляции. |
| Папка Data | Директория, содержащая классы для создания базы данных и всех необходимых таблиц и зависимостей. |
| Папка ViewModels | Директория, содержащая некоторые класса взаимодействия со страницами. |

# **3.2 Взаимоотношение между классами**

Для визуализации взаимосвязей между классами используется диаграмма UML — графическое представление набора элементов, изображаемое чаще всего в виде связанного графа с вершинами (сущностями) и рёбрами (отношениями).

Для представления внутренней структуры программы в виде классов и связей между ними используется диаграмма классов. Приложение спроектировано таким образом, что каждый класс выполняет свои функции и практически не зависит от других. Диаграмма классов представлена в приложении Б.

# **3.3 Проектирование логической структуры базы данных**

Для создания базы данных описываемого приложения использовалась система управления реляционными базами данных SQL Server Management Studio 20.

База данных — это совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимо от прикладных программ.

Система управления базами данных — совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами.

База данных данного курсового проекта состоит из 5 таблиц. Схема базы данных PCConfiguratorDB изображена на рисунке 3.3.

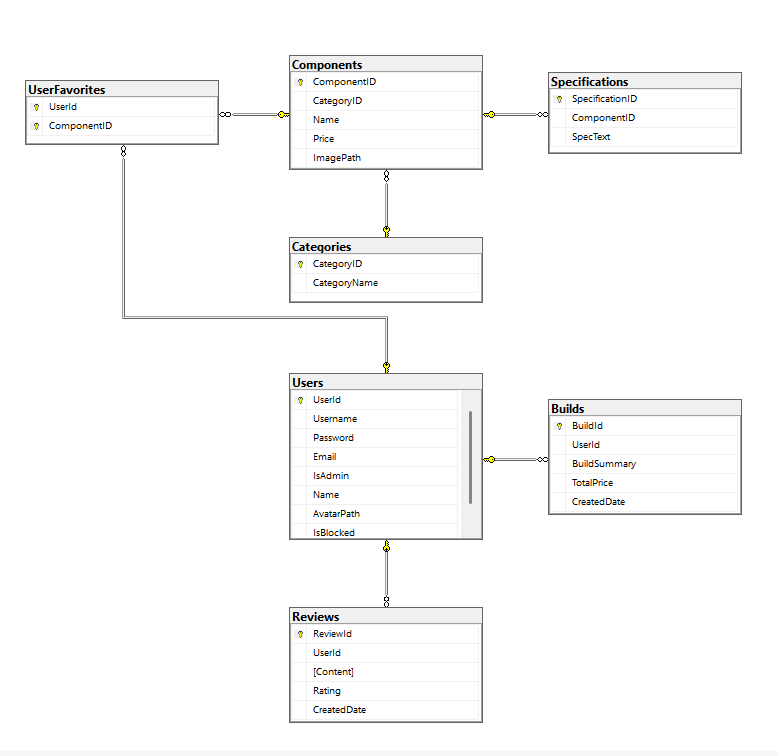


Рисунок 3.3 — Схема базы данных приложения

Остановимся более подробно на некоторых основных таблицах и рассмотрим, какие поля они содержат.

В таблице 3.3 представлена структура таблицы «Users», которая содержит информацию о пользователях системы. Таблица включает в себя персональные данные, а также тип аккаунта.

Таблица 3.3 — Таблица «Users»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| UserId | INTEGER | Уникальный идентификатор пользователя |
| Username | TEXT | Логин пользователя |
| Password | TEXT | Пароль |
| Email | TEXT | Электронная почта |
| IsAdmin | BOOLEAN | Флаг административных прав (true - администратор) |
| Name | TEXT | Имя пользователя |
| AvatarPath | TEXT | Путь к изображению профиля |
| IsBlocked | BOOLEAN | Флаг блокировки пользователя |

Таблица 3.4 — Таблица «Builds»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| BuildId | INTEGER | Уникальный идентификатор сборки |
| UserId | INTEGER | Идентификатор пользователя, который создал сборку |
| BuildSummary | TEXT | Краткое описание сборки |
| TotalPrice | DECIMAL | Общая стоимость сборки |
| CreatedDate | DATETIME | Дата создания сборки |

Таблица 3.5 — Таблица «Components»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| ComponentID | INTEGER | Уникальный идентификатор компонента |
| CategoryID | INTEGER | Идентификатор категории компонента |

Продолжение таблицы 3.5:

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| Name | TEXT | Название компонента |
| Price | DECIMAL | Цена компонента |
| ImagePath | TEXT | Путь к изображению компонента |

Таблица 3.6 — Таблица «Categories»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| CategoryID | INTEGER | Уникальный идентификатор категории |
| CategoryName | TEXT | Название категории |

Таблица 3.7 — Таблица «Specifications»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| SpecificationID | INTEGER | Уникальный идентификатор спецификации |
| ComponentID | INTEGER | Идентификатор компонента, к которому относится спецификация |
| SpecText | TEXT | Текст спецификации |

# **3.4 Проектирование архитектуры приложения**

Архитектура программного обеспечения представляет собой совокупность ключевых решений, касающихся организации программной системы. Она включает выбор структурных элементов и их интерфейсов, определение их взаимодействия, а также объединение этих элементов в более крупные системы. Однако, центральную роль играет архитектурный стиль, который определяет всю организацию системы, включая элементы, их интерфейсы, их сотрудничество и способы их соединения.

Для удовлетворения требований проектируемой системы с точки зрения различных атрибутов качества, применяются различные архитектурные шаблоны, так называемые паттерны. В разрабатываемых приложениях часто используется архитектурный шаблон Model-View-ViewModel (MVVM). Этот шаблон состоит из трех компонентов: модели (Model), модели представления (ViewModel) и представления (View).

На рисунке 3.4 представлена диаграмма, которая показывает общую структуру приложения в рамках шаблона MVVM.

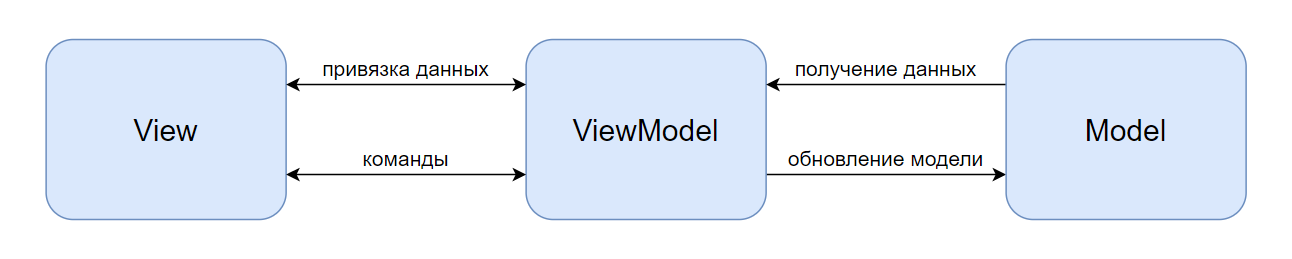


Рисунок 3.4 — Структура шаблона MVVM

Model (модель) описывает используемые в приложении данные. Модели могут содержать логику, непосредственно связанную с этими данными, например, логику валидации свойств модели. В то же время модель не должна содержать никакой логики, связанной с отображением данных и взаимодействием с визуальными элементами управления. Часто модель реализует интерфейс INotifyPropertyChanged, который позволяет системе автоматически обнаруживать изменения свойств модели и облегчает привязку к представлению, хотя сама модель не взаимодействует напрямую с представлением.

View (представление) определяет визуальный интерфейс, через который пользователь взаимодействует с приложением. Применительно к WPF представление — это код XAML, который определяет интерфейс в виде кнопок, текстовых полей и прочих визуальных элементов. Представление не обрабатывает события, а в основном выполняет действия посредством команд.

ViewModel (модель представления) связывает модель и представление через механизм привязки данных. Если в модели изменяются значения свойств, при реализации моделью интерфейса INotifyPropertyChanged автоматически идет изменение отображаемых данных в представлении, хотя напрямую модель и представление не связаны. Поскольку элементы представления, то есть визуальные компоненты типа кнопок, не используют события, то представление взаимодействует с ViewModel посредством команд.

Для своего проекта я выбрал в основу архитектуру Code-Behind, но также использовал архитектуру MVVM для некоторых задач. Code-Behind — это подход в разработке WPF-приложений, при котором логика работы интерфейса пишется в файлах .xaml.cs, связанных с XAML-разметкой. В отличие от MVVM, где используется разделение на Model-View-ViewModel, Code-Behind позволяет напрямую управлять элементами интерфейса и обрабатывать события в коде.

Преимущества Code-Behind:

– Быстрое прототипирование — не нужно создавать ViewModel и Binding-команды.

– Понятность — вся логика сосредоточена в одном файле.

– Можно динамически изменять свойства элементов (видимость, текст, стили) без сложных привязок.

Удобно для анимаций и кастомных взаимодействий.

# **3.5 Проектирование архитектуры окон**

Приложение включает в себя 10 окон и 7 страниц.

При запуске приложения пользователь видит окно авторизации. Пользователь может ввести данные и получить доступ к стандартному функционалу приложения (при входе под ролью обычного пользователя) или же к расширенному функционалу (при входе под ролью администратора), либо перейти к окну «Регистрация», после которого он может ввести данные в форму входа и также получить доступ к приложению.

Внешний вид стартовой страницы «Авторизациии» приложения «PCConfigurator» представлен на рисунке 3.5.

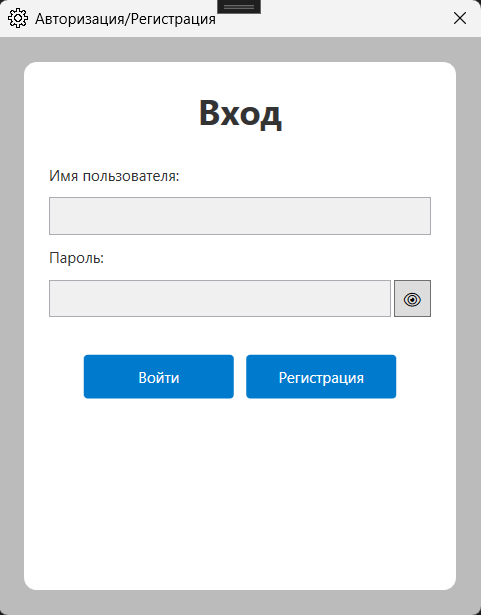


Рисунок 3.5 — Стартовая страницы «PCConfigurator»

В этом окне требуется ввести только логин и пароль для входа в аккант, если аккаунта нет, то следует перейти по кнопке регистрации.

Внешний вид страницы регистрации представлен на рисунке 3.6.

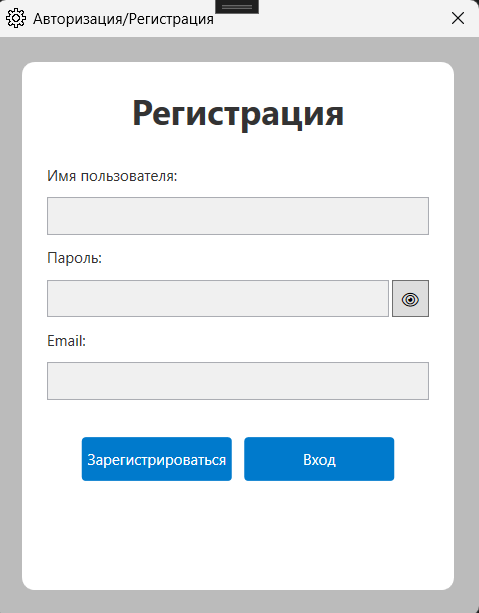


Рисунок 3.6 — Страница регистрации

После входа в приложение, в зависимости от роли, пользователь переходит на страницу администратора, или обычного пользователя.

Боковое меню состоит из набора кнопок, которые относятся к определённому типу пользователя согласно его уровню доступа. Эти кнопки позволяют перейти на страницы с функционалом. Далее рассмотрены варианты окон.

Страница пользователя даёт доступ к стандартному функционалу приложения для выбора комплектующих и сборки конфигурации. Окно пользователя изображено на рисунке 3.7.

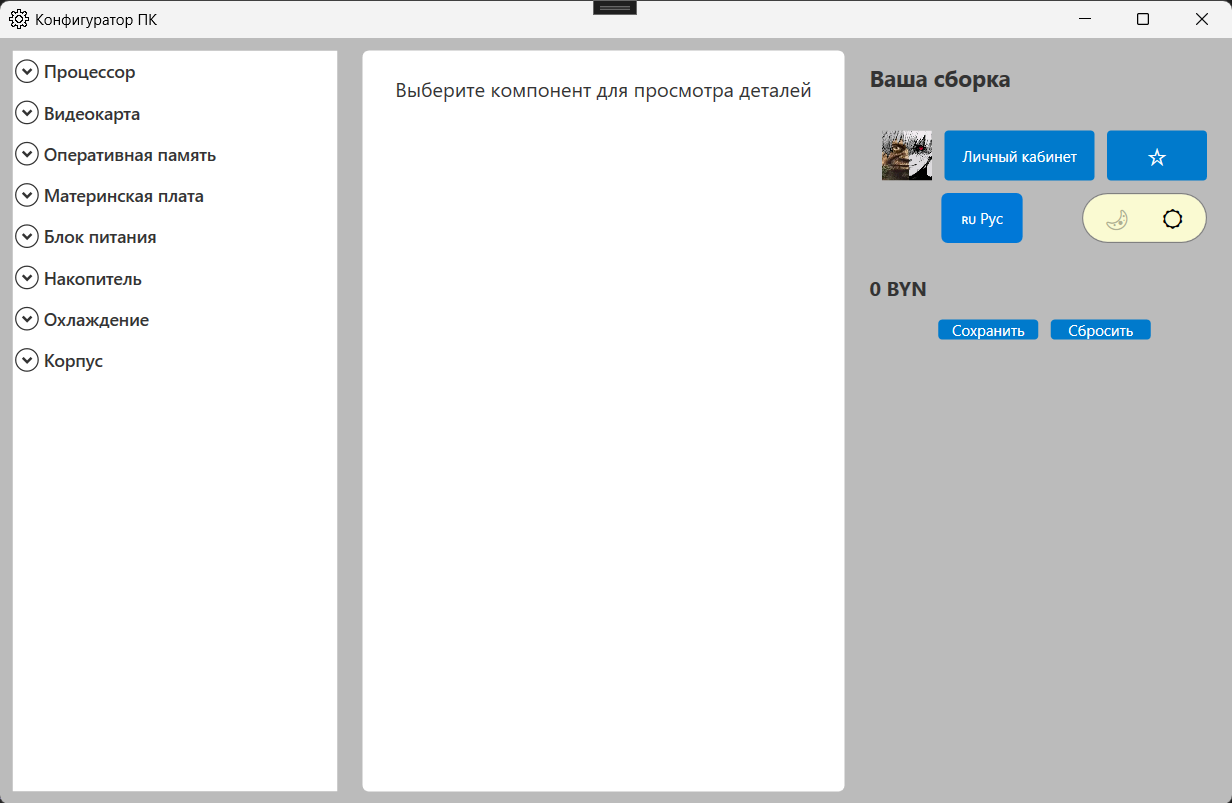


Рисунок 3.7 — Окно для пользователя

При авторизации под логином и паролем администратора, приложение открывает тоже окно, но с дополнительными кнопками, пункты меню изображены на рисунке 3.9.

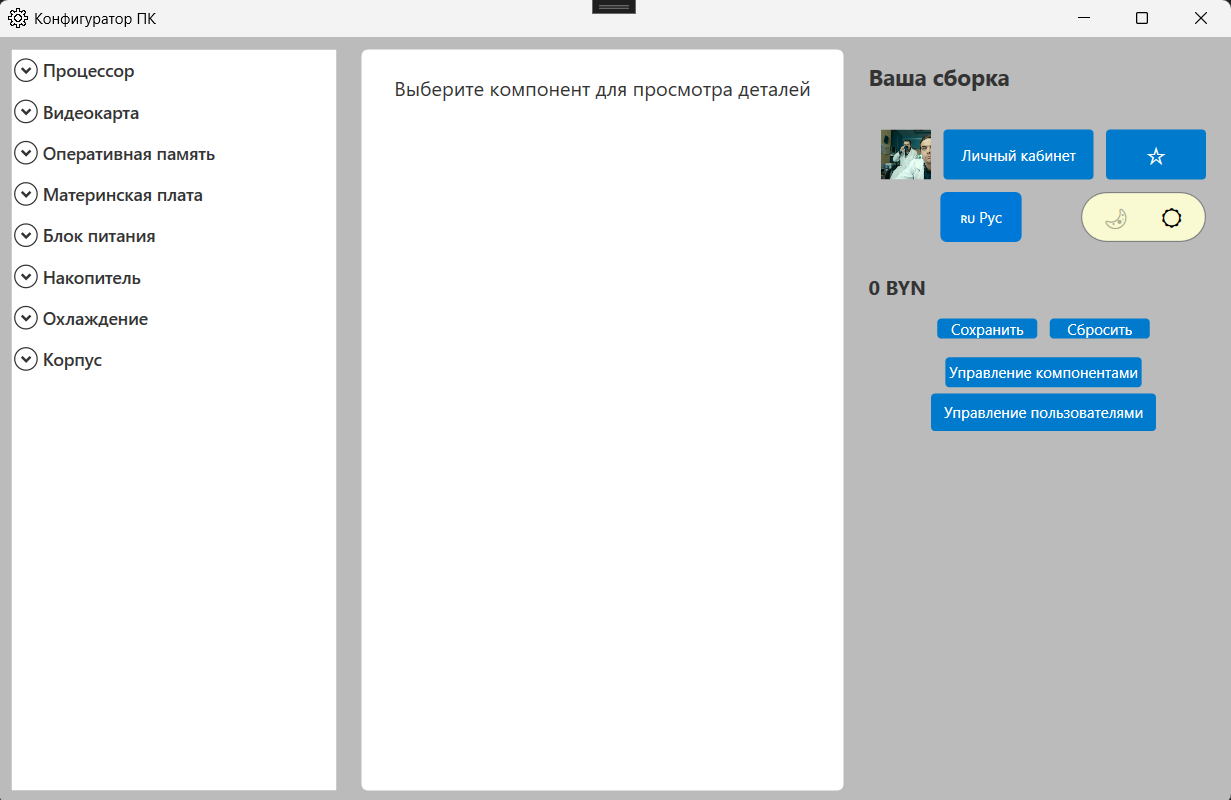


Рисунок 3.9 — Окно администратора, пункты меню администратора

Администратор имеет две дополнительные кнопки, а именно “Управление компонентами” и “Управление пользователями”. Они позволяют изменить список товаров и заблокировать пользователей. Изменения в окнах администратора также редактируют таблицы БД.

# **3.6 Схема работы приложения**

Для описания маршрутов работы программного продукта была создана диаграмма деятельности, которая показывает пути перехода между страницами приложения, представлена на рисунке 3.10.

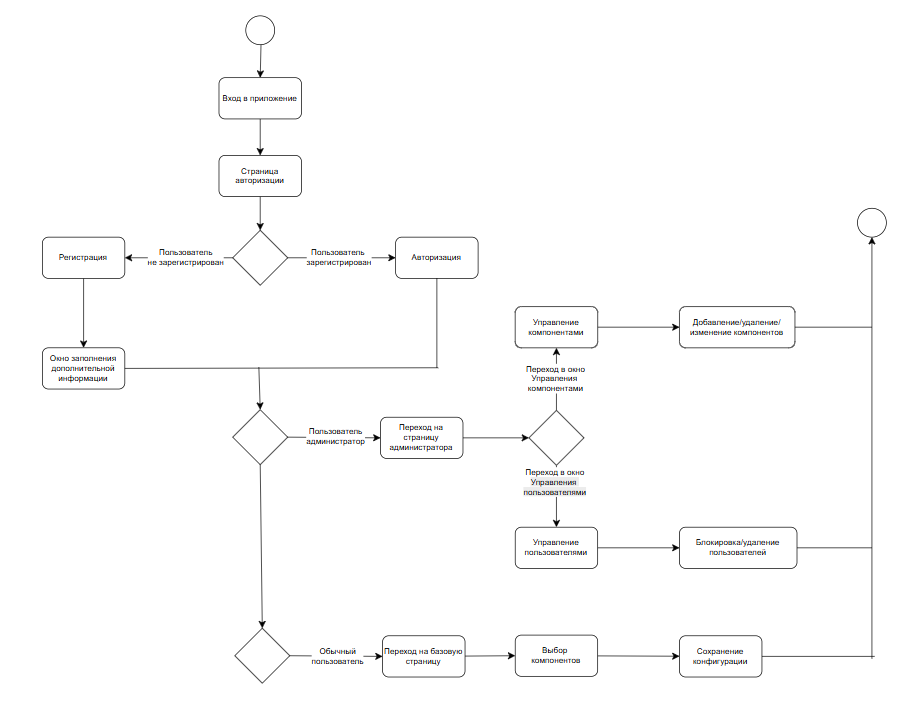


Рисунок 3.10 — Диаграмма деятельности

Для наглядного отображения взаимодействия между объектами системы в различные моменты времени в рамках одного сценария использования применяется диаграмма последовательностей в UML. Эта диаграмма позволяет иллюстрировать, как объекты системы взаимодействуют друг с другом для выполнения определенной функции, а также отображает порядок их взаимодействия в рамках конкретного случая использования.

Пример диаграммы последовательности представлен в приложении на рисунке 3.11.

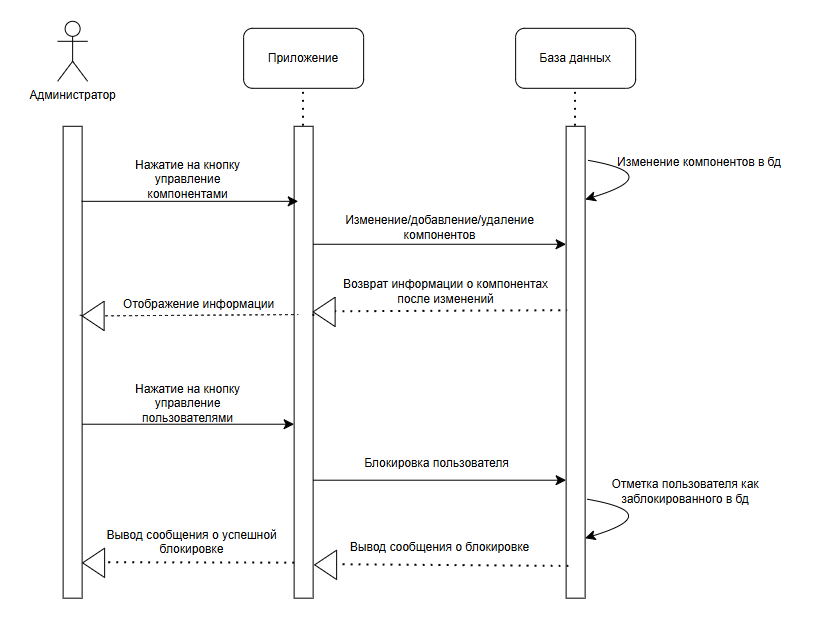


Рисунок 3.11 - диаграмма последовательности

Для отображения времени на диаграмме используется линия жизни объекта, которая представляет период его существования в системе и изображается вертикальной штриховой линией, направленной вниз. Объекты представлены прямоугольниками, а сообщения, которыми они обмениваются, обозначаются линиями с стрелками.

# **4 Реализация программного средства**

Следующим этапом разработки приложения является непосредственная реализация программного решения в соответствии с уже сформированными требованиями и шаблонами.

# **4.1 Основные классы программного средства**

Для выполнения технических задач программного средства «Конфигуратор ПК» должны быть реализованы следующие функции и соответствующие им классы и методы:

– Добавление/удаление/редактирование комплектующих.

– Модерация объявлений.

– Просмотр активных заказов.

– Просмотр истории заказов.

– Блокировка и разблокировка аккаунтов.

– Генерация отчетов.

– Функции клиента:

– Регистрация и авторизация.

– Просмотр списка доступных комплектующих.

– Поиск и фильтрация комплектующих.

– Просмотр активной заказов, истории своих заказов.

– Оформление заказа.

– Возможность сохранения сборок, добавления в избранные.

– Возможность оставить отзыв и оценку.

– Личный кабинет.

Далее подробно рассмотрены каждые из необходимых для выполнения технических задач функции, а также созданные для их выполнения классы и методы и их функционал и реализация.

# **4.2 Регистрация и авторизация**

Доступ в систему является ключевым звеном в обеспечении безопасности и персонализации пользовательского опыта. Для этого создано окно авторизации (AuthWindow) с тесной связью с ViewModel (AuthViewModel), что позволяет обрабатывать ввод пользователя мгновенно. При регистрации система предъявляет обязательные поля – имя пользователя, пароль, подтверждение пароля и электронный адрес. Каждое из этих полей проходит строгую проверку на соответствие формату и обязательности заполнения, что позволяет избежать ошибок и повысить надежность регистрации.

После успешного ввода данных, специальный сервис AuthService, выступающий посредником между пользовательским интерфейсом и базой данных через контекст PCComponentsContext, сохраняет информацию о новом пользователе. В режиме авторизации предъявляются только необходимые поля (имя пользователя и пароль), что экономит время пользователей и способствует быстрому входу в приложение.

# **4.3 Управление профилем пользователя**

Для обеспечения удобного управления личными данными пользователя разработано отдельное окно (ProfileWindow) и соответствующая ViewModel (ProfileViewModel). Пользователь имеет возможность не просто просматривать свои текущие данные, но и изменять их, загружать аватар, а также отправлять сборки на указанный электронный адрес.

Особое внимание уделено истории заказов и бронирований, которая доступна в личном кабинете. Это помогает пользователю в анализе своих покупок, а также в принятии решений о повторном заказе.

Механизм обновления данных, реализованный посредством методов сохранения в базе через PCComponentsContext, гарантирует, что информация всегда актуальна и синхронизирована с сервером.

# **4.4 Просмотр и управление услугами**

Для администрирования перечня услуг (комплектующих) реализовано окно ComponentsWindow с соответствующим ComponentsViewModel.

Администратор получает возможность добавлять, редактировать и удалять записи, которые хранятся в базе данных. Взаимодействие с данными осуществляется через набор моделей (например, ComponentEntity, Specification, Category), а изменения сохраняются через EF Core.

# **4.5 Блокировка и разблокировка аккаунтов**

Функциональность блокировки и разблокировки аккаунтов реализована через окно управления пользователями (UserManagementWindow) и связанный с ним UserManagementViewModel. Такой инструмент позволяет администратору оперативно реагировать на инциденты, просматривая список всех зарегистрированных пользователей и переключая их статус с помощью команд интерфейса.

Изменения статуса блокировки мгновенно отражаются на соответствующих объектах User через обновляемое свойство IsBlocked, что в свою очередь моментально синхронизируется с базой данных и обновляет пользовательский интерфейс. Это обеспечивает высокий уровень безопасности и контроля над системой.

# **4.6 Работа с отзывами**

Обратная связь – один из ключевых элементов совершенствования сервиса, поэтому предусмотрена возможность оставлять отзывы непосредственно о предоставляемых услугах. Через окно создания отзывов (ReviewWindow) пользователь может оставить комментарий, выставить рейтинг и зафиксировать дату отправки.

Класс Review аккумулирует всю информацию, связанную с отзывом, и связывает её с данными пользователя посредством внешнего ключа. Просмотр оставленных ранее отзывов осуществляется через специально разработанные ViewModel, что позволяет пользователям оценивать качество оказанных услуг и помогает администрации своевременно вносить коррективы в систему.

# **4.7 Вспомогательные классы и сервисы**

Для обеспечения удобного доступа к данным и поддержания бизнес-логики используются:

PCComponentsContext – контекст базы данных, реализованный на базе Entity Framework Core, обеспечивает CRUD‑операции для всех моделей.

AuthService – класс, инкапсулирующий логику авторизации и регистрации, позволяющий отделить логику работы с БД от представлений.

RelayCommand – универсальная реализация интерфейса ICommand, позволяющая связывать действия с командами в ViewModel.

Дополнительно реализованы вспомогательные классы и конвертеры (например, InverseBooleanToVisibilityConverter, BlockStatusConverter), позволяющие динамически изменять отображение элементов интерфейса в зависимости от состояния приложения.

# **4.8 Разделение на слои MVVM**

Каждое окно (View) имеет свой набор данных и логику, реализованную в соответствующем ViewModel (AuthViewModel, MainViewModel, ProfileViewModel, ComponentsViewModel, UserManagementViewModel). Использование MVVM позволяет полностью изолировать визуальное представление от логики приложения, что значительно упрощает тестирование, отладку и дальнейшее расширение функционала. Такой подход гарантирует гибкость разработки, позволяя адаптироваться под изменяющиеся требования и обеспечивать стабильную работу продукта в условиях постоянного роста нагрузки и функциональности.

# **5. Тестирование, проверка работоспособности и анализ**

# **5.1 Тестирование, проверка работоспособности**

Для обеспечения корректной работы программного средства «Конфигуратор ПК» было проведено тестирование, направленное на выявление и обработку ошибок, возникающих при некорректном вводе данных пользователем, а также проверку валидации данных, взаимодействующих с базой данных. Особое внимание уделялось сценариям, связанным с регистрацией, авторизацией, конфигурацией ПК, так как ошибки в этих процессах могут привести к неработоспособности приложения.

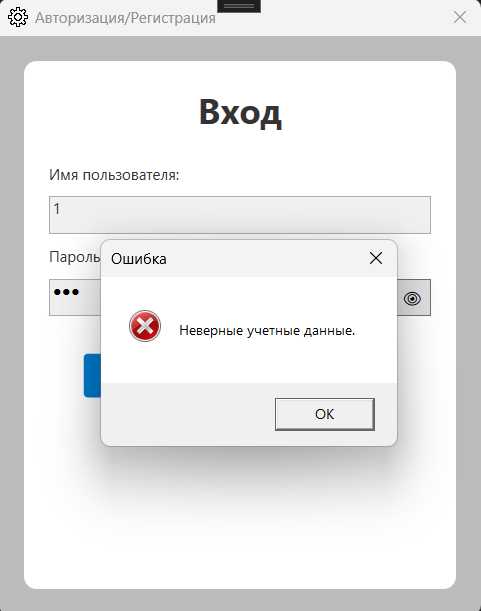


Рисунок 5.1 – Обработка исключений в RegisterWindow.xaml

В окне AuthWindow.xaml была протестирована улучшенная валидация полей ввода. Логин должен содержать 4–50 символов (буквы, цифры, подчёркивания), пароль — минимум 4 символа, полное имя — только буквы и пробелы (включая кириллицу), а email — соответствовать корректному формату (например, example@domain.com). При некорректном вводе, например, логине короче 4 символов или email без символа @, система отображает сообщения об ошибках под соответствующими полями с использованием NullToVisibilityConverter. На рисунке 5.1 показано сообщение: «Неверные учётные данные» .

В окне LoginWindow.xaml тестировалась обработка ошибки при вводе неверного логина или пароля. Если пользователь указывает несуществующие учётные данные, приложение отображает сообщение об ошибке.

# **5.2 Вывод по главе**

В данной главе был проведён анализ тестирования программного средства «Благотворительный фонд», направленный на проверку обработки ошибок и валидации данных в ключевых функциональных сценариях. Тестирование подтвердило, что приложение эффективно обрабатывает некорректный ввод, предотвращая сбои благодаря валидации в код-бихайнд и использованию конвертеров, таких как NullToVisibilityConverter. Сообщения об ошибках, отображаемые динамически, обеспечивают чёткую обратную связь, позволяя пользователю исправить данные и продолжить работу. Взаимодействие с базой данных организовано надёжно, с учётом ограничений, таких как уникальность логина. Результаты тестирования демонстрируют соответствие приложения требованиям, обеспечивая стабильность, удобство использования и устойчивость к ошибкам для пользователей и администраторов.

В результате проведенного тестирования спроектированного приложения были выявлены ключевые аспекты его работоспособности и надежности. Тесты на страницах регистрации и авторизации подтвердили, что валидация данных выполняется корректно, обеспечивая информирование пользователей о допущенных ошибках. Аналогично, функциональность добавления и удаления товаров администратором также прошла успешные проверки, что свидетельствует о надежности системы в обработке некорректных данных.

# **6. Руководство по установке и использованию**

# **6.1 Использование программного средства**

Программное средство «Конфигуратор ПК» работает с локальной базой данных, сервер которой установлен на компьютере пользователя, что устраняет необходимость дополнительной установки базы данных вместе с приложением. Это обеспечивает простоту развертывания и удобство использования для пользователей с минимальными техническими навыками. Локальная база данных гарантирует быстрый доступ к данным и их надёжное хранение без зависимости от внешних серверов. Все операции, такие как регистрация пользователей, добавление проектов и обработка пожертвований, выполняются локально, что повышает безопасность и минимизирует задержки. Благодаря интуитивному интерфейсу и автоматической обработке ошибок приложение доступно даже для пользователей, не имеющих опыта работы с подобными системами.

При запуске приложения открывается окно авторизации, представленное на рисунке 6.1.

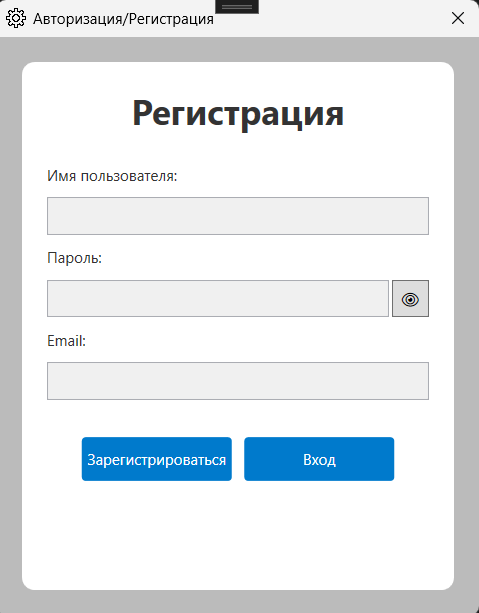


Рисунок 6.1 – Окно авторизации

Окно авторизации (AuthWindow.xaml) требует ввода логина и пароля для начала работы с приложением. Если учётная запись отсутствует, пользователь может нажать кнопку «Зарегистрироваться», после чего он будет перенаправлен в окно регистрации, представленное на рисунке 6.2. В форме регистрации необходимо заполнить поля логина, пароля, полного имени и email, которые проходят валидацию на соответствие формату (например, логин — 4–50 символов, email — корректный формат).

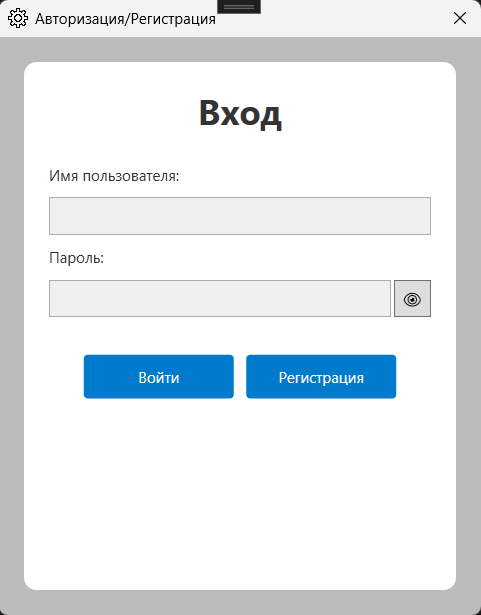


Рисунок 6.2 – Окно регистрации

После успешной регистрации и последующей авторизации открывается главное окно приложения, представленное на рисунке 6.3. Если пользователь вводит неверные учётные данные, отображается сообщение об ошибке, например, «Неверный логин или пароль». Основные ошибки авторизации и регистрации были рассмотрены в разделе 5.

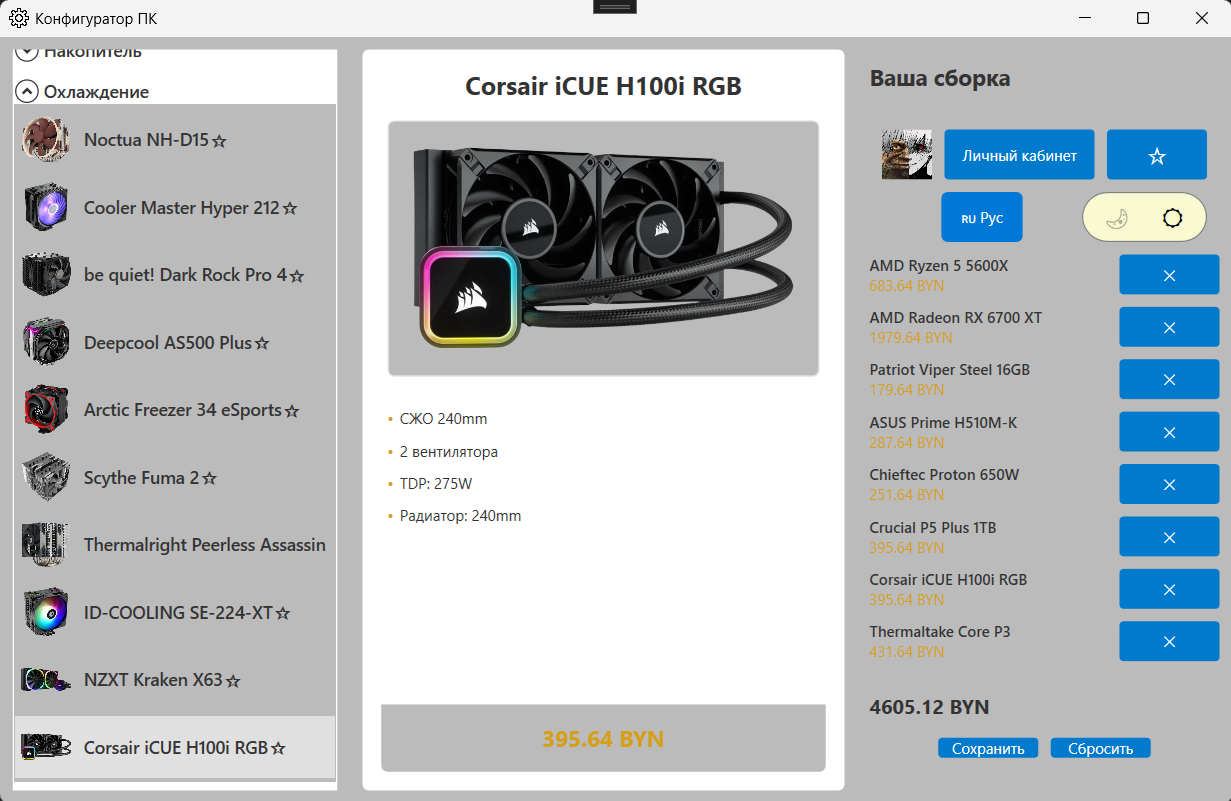


Рисунок 6.3 – Главное окно приложения

Главное окно (MainWindow.xaml) предоставляет доступ к основным функциям приложения. Можно выбирать комплектующие, которые автоматически добавляются в сборку при их открытии, а также убрать при необходимости. Есть кнопки смены темы, языка, сохранения и очистки сборки, а также профиля.

# **Список использованной литературы**

1. Хабр [Электронный ресурс] / Режим доступа: <https://habr.com> – Дата доступа: 11.04.2025
2. METANIT.COM Сайт о программировании [Электронный ресурс] / Режим доступа: <https://metanit.com> – Дата доступа: 12.04.2025
3. Пацей, Н.В. Технология разработки программного обеспечения / Н.В. Пацей. – Минск: БГТУ, 2016. – 129 с. [3]. MSDN сеть разработчиков в Microsoft [Электронный ресурс] / Режим доступа: <http://msdn.microsoft.com/library/> – Дата доступа: 12.03.2025
4. StackOverflow [Электронный ресурс] / Режим доступа: <https://stackoverflow.com>. – Дата доступа: 11.05.2025.
5. ProfessorWeb .NET & Web Programming [Электронный ресурс] / Режим доступа: <https://professorweb.ru> – Дата доступа: 23.04.2025
6. Сайт “Bloomerang”[Электронный ресурс] – Режим доступа: <https://bloomerang.co/features/mobile-app/> - Дата доступа: 05.03.2025
7. Сайт “DonorPerfect”[Элекронный ресурс] – Режим доступа: <https://www.donorperfect.com/fundraising-software/donorperfect-mobile-app/> - Дата доступа: 05.03.2025
8. Сайт “Charity Advantage”[Электронный ресурс] – Режим доступа: <https://www.charityadvantage.com/> - Дата доступа: 05.03.2025

**Приложение А**

|  |
| --- |
|  |

Рисунок 1 – UML-диаграмма

# **Приложение Б**

|  |
| --- |
|  |

Рисунок 1 – UseCase-диаграмма для пользователя

|  |
| --- |
|  |

Рисунок 2 – UseCase-диаграмма для администратора

# **Приложение В**

|  |
| --- |
|  |