МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-47 01 02 «Дизайн электронных и веб-изданий»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированное программирование»

Тема «Программное средство "Собери устройство"»

Исполнитель

студентка 2 курса 9 группы Белавская Анна Вадимовна

(Ф.И.О.)

Руководитель ассистент Бакиев А.Р.

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

Курсовой проект защищен с оценкой

Руководитель Пацей Н.В.

(подпись)

## Минск 2019

## **СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
|  |  |
| Введение 4 |  |
| 1. Аналитическая часть разработки 5 |  |
| * 1. Аналитический обзор прототипов 5 |  |
| * 1. Анализ требований к программному средству и   разработка функциональных требований 8   * 1. Информационные системы и технологии 8 |  |
| 1. Проектирование программного средства 9    1. Авторизация и регистрация 9    2. Создание конфигурации персонального компьютера 9    3. Работа с базами данных 11 |  |
| 1. Тестирование и анализ полученных результатов 14 |  |
| 1. Руководство по установке и использованию 18 |  |
| Заключение 24 |  |
| Список использованных источников 25 |  |
| Приложение А 26 |  |
| Приложение Б 27  Приложение В 28 |  |

## **ВВЕДЕНИЕ**

Тема курсового проекта — «Собери устройство».

Актуальность темы курсового проекта вызвана тем, что на данный момент для пользователя существует множество возможностей приобрести какую-либо технику в готовом варианте или собрать ее самостоятельно из доступных в продаже компонентов. Созданная в ходе проекта программа позволит упростить процесс поиска, предлагая на выбор различные компоненты для комплектации компьютера, а также готовые варианты уже собранных комплектов. Для удобства использования программы реализован графический интерфейс, позволяющий увидеть результат в процессе сборки.

Основной задачей курсового проекта является создание возможности для пользователей комфортно управлять сборкой компьютера желаемыми компонентами.

Основная цель курсового проекта — разработка приложения для комплектации компьютеров.

**1. Аналитическая часть разработки**

**1.1. Аналитический обзор прототипов**

Аналитический обзор аналогов и прототипов программного средства показал, что подобные конструкторы используются в основном на сайтах компаний, занимающихся сборкой и продажей сборных либо готовых моделей компьютеров и их периферией. В своих конструкторах они предлагают выбрать необходимую величину параметров при их наличии, а также предлагают на выбор несколько доступных моделей компонентов. В конструкторе присутствуют не только необходимые компоненты системного блока, поля с которыми могут быть отмечены специальным знаком, но и элементы компьютерной периферии, позволяющие собрать не просто системный блок, а полноценный компьютер. Цены на компоненты суммируются с ценой сборки, которую выполняет компания, и выводится в качестве итоговой цены комплекта.

Варианты готовых сборок делятся на типы компьютеров по их основному назначению. Так, например, на сайте hyperpc.ru предлагается следующий способ деления компьютеров на типы:

* игровой компьютер. Компьютер, который будет сочетать в себе высокую производительность, оптимальную стоимость и современные технологии.
* эксклюзивный компьютер. Уникальный компьютер ручной работы с эксклюзивным дизайном, водяной системой охлаждения и экстремальным разгоном;
* бесшумный компьютер. Решение для пользователей, предъявляющих повышенные требования к уровню шума и производительности системного блока;
* рабочая станция. Профессиональный компьютер для работы с системами CAD, работы с 2D и 3D графикой, видеомонтажа и ресурсоемких вычислений;
* компьютер в столе. Компьютерный стол, созданный из материалов и компонентов премиум качества, воплощенный в жизнь благодаря стараниям лучших специалистов.

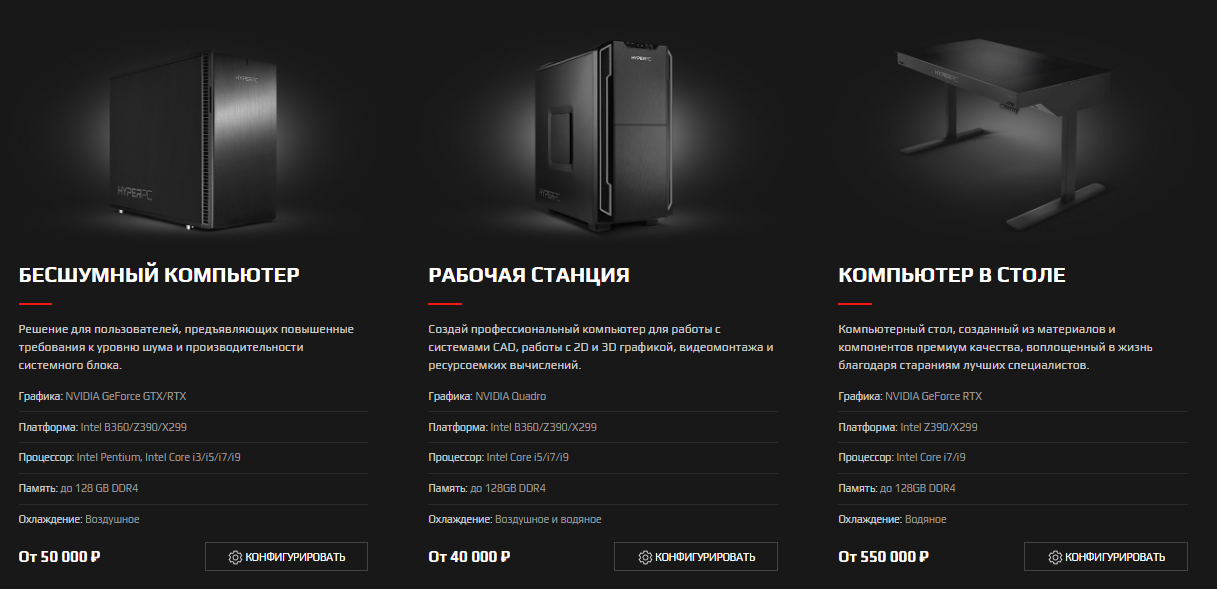


Рисунок 1.1 – Вариант деления сборок на категории

Данное деление на категории не является истинно верным, но оно позволяет как опытному пользователю, так и новичку, желающему приобрести компьютер, правильно сориентироваться в его выборе по небольшому описанию возможностей предлагаемой сборки.

Важными составляющими интерфейса являются возможность выбора конкретных моделей компонентов, а также выбор его основных параметров. Некоторые компоненты должны быть отмечены как обязательные составляющие сборки и иметь соответствующую отметку для привлечения внимания пользователя.

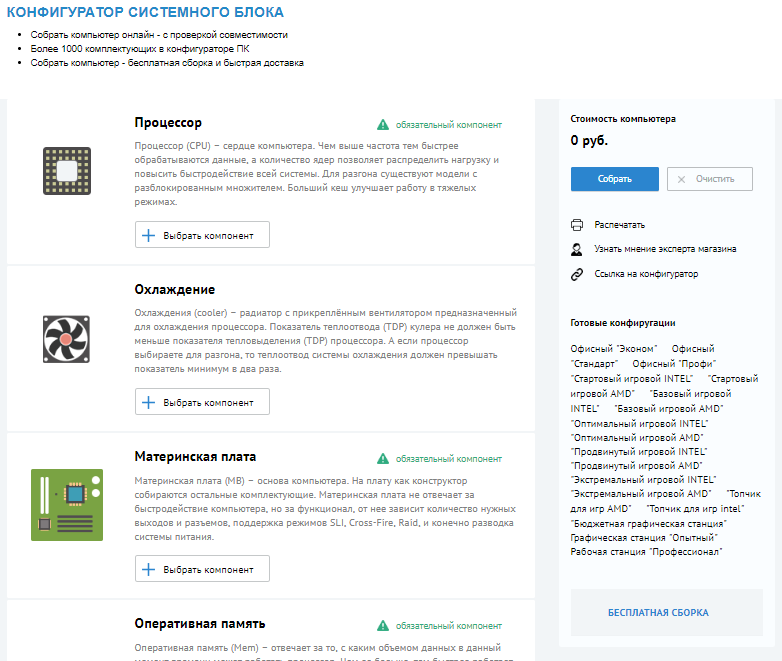


Рисунок 1.2 – Пример конструктора с отметкой обязательных компонентов на сайте ironbook.ru

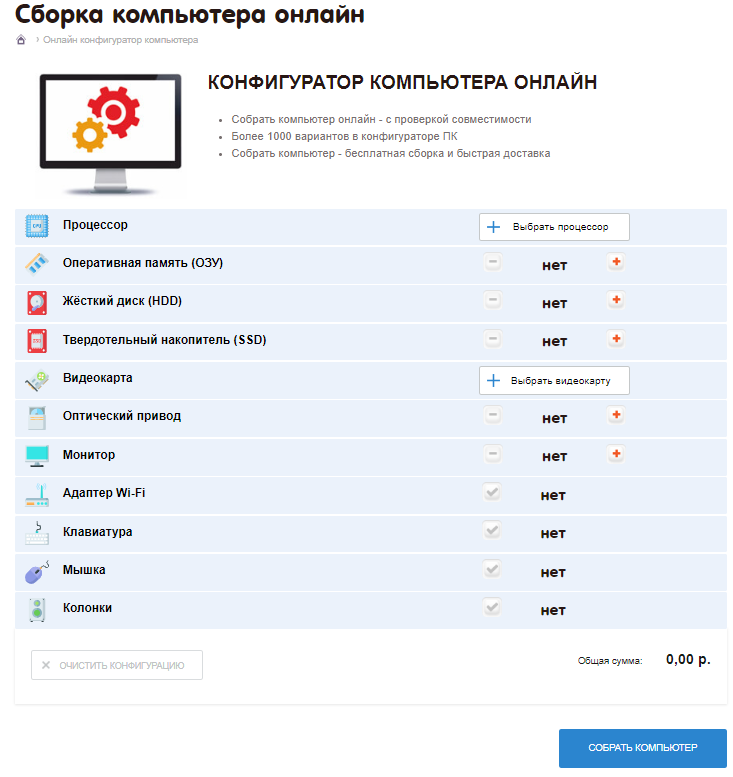


Рисунок 1.3 – Аналог конструктора на сайте pcmarket.by

Важной составляющей графического интерфейса программы является цветовое решение. Его основная задача — привлечение внимания пользователя и комфорт использования программного средства. Цвета выбранной темы должны подчеркивать достоинства модели компьютера, но вместе с тем не должны мешать правильному отображению элементов системного блока и периферии. Для персонализации программы могут быть предложены альтернативные варианты цветовых решений. Цветовая тема также может изменяться для лучшего отображения внешнего вида компьютера.

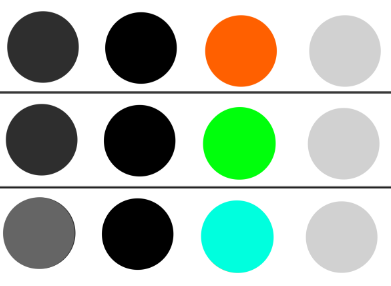


Рисунок 1.4 – Варианты цветового решения программы

При выборе компонентов пользователю должны быть предложены альтернативные варианты с указанными достоинствами и недостатками моделей для помощи в выборе и наглядности различий между ними.

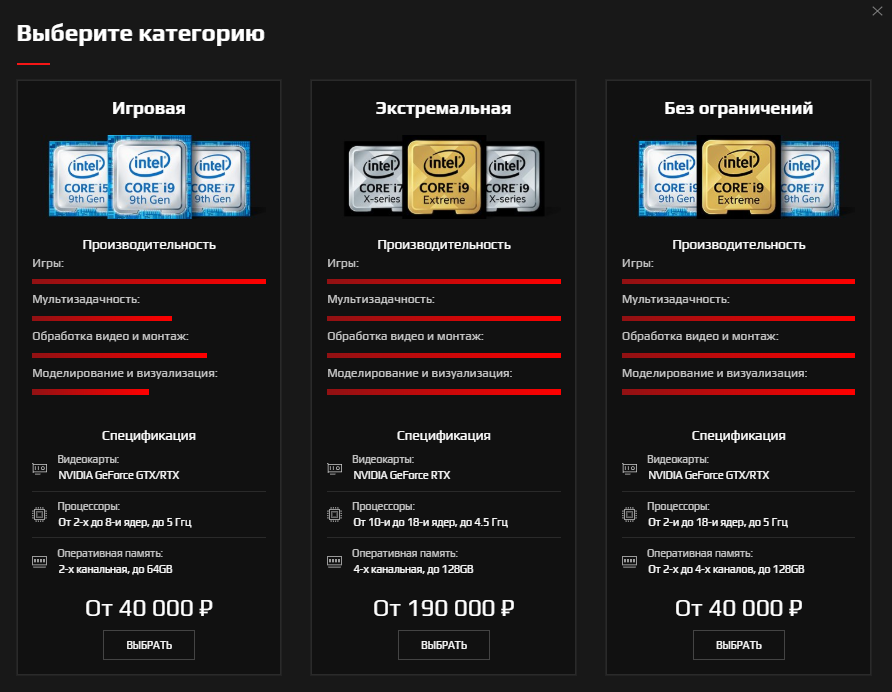


Рисунок 1.5 – Варианты компонентов

**1.2. Анализ требований к программному средству и разработка функциональных требований**

Основываясь на анализе прототипов можно сделать вывод о требованиях к данной программе. Конструктор компьютера должен включать в себя несколько режимов, а конкретно режим пользователя и режим администратора. В режиме пользователя будут доступны такие возможности как:

* добавление и удаление компонентов в системном блоке;
* сортировка предлагаемых компонентов по цене и названию;
* сохранение и загрузка шаблонов сборки.

В режиме администратора помимо основного функционала программы будет доступна функция добавления шаблонов сборок в базу данных.

Графическая часть интерфейса должна быть выполнена в едином стиле в соответствии с нормами сочетания цветов и графических объектов.

**1.3. Информационные системы и технологии**

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2019;
* язык программирования C#;
* расширяемый язык разметки XAML;
* SQL Server Management Studio 17;
* технология WPF.

C# — объектно-ориентированный язык программирования. Разработан для платформы Microsoft.

Технология WPF (Windows Presentation Foundation) собой систему для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .NET Framework, использующая язык XAML.

SQL символизирует собой Структурированный Язык Запросов. Язык, который дает возможность создавать и работать в реляционных базах данных, которые являются наборами связанной информации сохраняемой в таблицах.

XAML (Extensible Application Markup Language — расширяемый язык разметки приложений) представляет собой язык разметки, используемый для создания экземпляров объектов .NET. Хотя язык XAML — это технология, которая может быть применима ко многим различным предметным областям, его главное назначение — конструирование пользовательских интерфейсов WPF. Другими словами, документы XAML определяют расположение панелей, кнопок и прочих элементов управления, составляющих окна в приложении WPF.

**2. Проектирование программного средства**

**2.1. Авторизация и регистрация**

При запуске приложения появляется окно авторизации, в котором пользователи или администраторы с помощью textbox и passwordbox а также выбором типа user или admin могут авторизоваться в системе. С помощью кнопки Register можно перейти на окно регистрации, где также с помощью textbox и passwordbox можно зарегистрироваться в системе. Процессы регистрации и авторизации проходят следующим образом:

* пользователь вводит свои логин и пароль в соответствующие поля;
* при регистрации также необходимо ввести повтор пароля;
* данные обрабатываются и переносятся в базу данных;
* при авторизации из базы данных возвращается значение, если пользователь найден и может войти в систему.

Число попыток ввода пароля не ограничено. Логин и пароль пользователя и администратора могут содержать только буквы латинского алфавита и цифры.

**2.2. Создание конфигурации персонального компьютера**

После успешной авторизации в качестве пользователя, ему становятся доступным окно с кнопками перехода в конструктор конфигурации и в библиотеку готовых сборок (шаблонов).

При переходе по первой кнопке пользователь попадает в окно конструктора, где с помощью таблиц с данными о каждом компоненте может собрать желаемую конфигурацию персонального компьютера. Выбор компонента осуществляется с помощью нажатия на него и выбора в дальнейшем необходимой модели из представленных в таблице. При успешном выборе компонента его изображение на кнопке меняется с черно-белого на цветное. После выбора необходимого количества компонентов пользователь может сохранить свою конфигурацию в текстовом файле, где будет находиться вся информация о компонентах сборки. Сохранение осуществляется с помощью потокового вывода в файл с последующим его открытием с целью сохранения пользователем вручную данного файла с измененным названием и расположением.

При переходе по второй кнопке пользователь может выбрать тип шаблонов для просмотра:

* домашний (выбор персонального компьютера для домашнего использования);
* рабочий (выбор персонального компьютера для офиса);
* игровой (выбор персонального компьютера для использования в игровых целях).

После выбора одного из предложенных появится таблица с доступными конфигурациями, в которой описан каждый компонент а также добавлена примерная цена. При выборе одной из конфигураций ее данные будут сохранены в текстовый файл и выведены в программе Блокнот с целью пользовательского переименования и изменения расположения файла.

После успешной авторизации в качестве администратора станет доступна только функция добавления шаблона в используемые. Внешне она выглядит как конфигуратор сборки для пользователей с различием в том, что кроме компонентов необходимо выбрать тип полученной конфигурации. Все данные о сборке вносятся в базу данных с помощью кнопки сохранения.

На протяжении всего использования программы как пользователю так и администратору будут доступны кнопки перехода на предыдущую страницу, на страницу авторизации и кнопка завершения программы, расположенные в правом верхнем углу окна.

Функция авторизации подробно изображена на следующем рисунке:

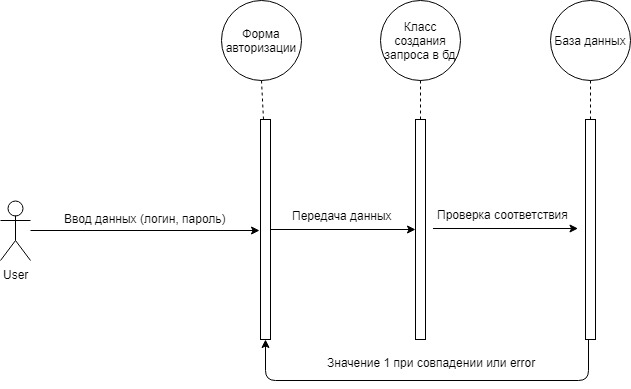


Рисунок 2.1 – Диаграмма последовательности авторизации

Решение CourseWork представлено одним проектом CourseWork, имеющим структуру, представленную на рисунке:

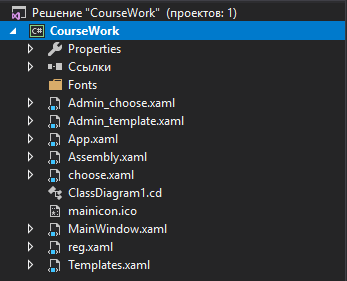


Рисунок 2.2 – Структура проекта

Таблица 2.1 – Описание структуры файлов проекта.

|  |  |
| --- | --- |
| Имя файла | Содержание |
| Admin\_choose.xaml | Окно выбора функции администратора |
| Admin\_template.xaml | Окно создания шаблона |
| Assembly.xaml | Окно создания сброки |
| Choose.xaml | Окно выбора функции пользователя |
| ClassDiagram1.cd | Диаграмма классов |
| Mainicon.ico | Иконка приложения |
| MainWindow.xaml | Окно авторизации |
| Reg.xaml | Окно регистрации |
| Templates.xaml | Окно выбора шаблона |

На рисунке 2.3 представлена блок-схема действий пользователя при работе с приложением.

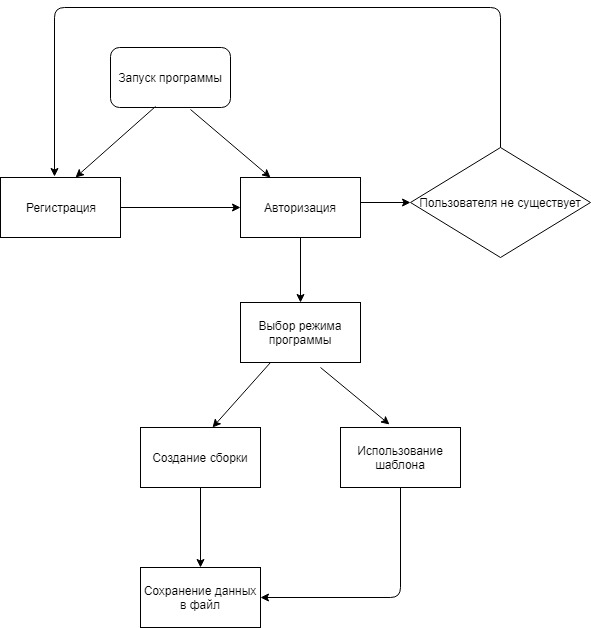


Рисунок 2.3 – Блок-схема работы с приложением

**2.3. Работа с базами данных**

Для работы была создана и использована одна база данных, которая содержит 12 таблиц: таблица данных пользователей, таблицы данных администраторов, таблицы с данными о шаблонах сборок и таблицы с данными о компонентах для конфигураций.



Рисунок 2.4 – Таблица для авторизации пользователей

Таблица users содержит поля:

* + login – логин пользователя;
  + password – пароль пользователя.

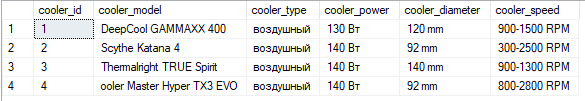


Рисунок 2.5 – Пример таблицы моделей для компонентов

Таблицы с моделями компонентов имеют схожее содержание, поэтому их можно описать на примере одной таблицы Coolers:

* cooler\_id содержит порядковый номер модели в таблице;
* cooler\_model – модель компонента;
* cooler\_type – тип кулера;
* cooler\_power – мощность;
* cooler\_diameter – диаметр лопастной части;
* cooler\_speed – скорость вращения.



Рисунок 2.6 – Пример таблицы шаблона конфигурации

Как и таблицы компонентов, таблицы шаблонов по типам идентичны по своему строению, поэтому их можно описать на примере одной таблицы Home:

* id\_temp содержит порядковый номер шаблона;
* cooler – модель кулера;
* cpu – модель процессора;
* video – модель видеокарты;
* oper – модель оперативной памяти;
* power – модель блока питания;
* hard – модель жесткого диска;
* mother – модель материнской платы;
* price – цена полученной сборки.

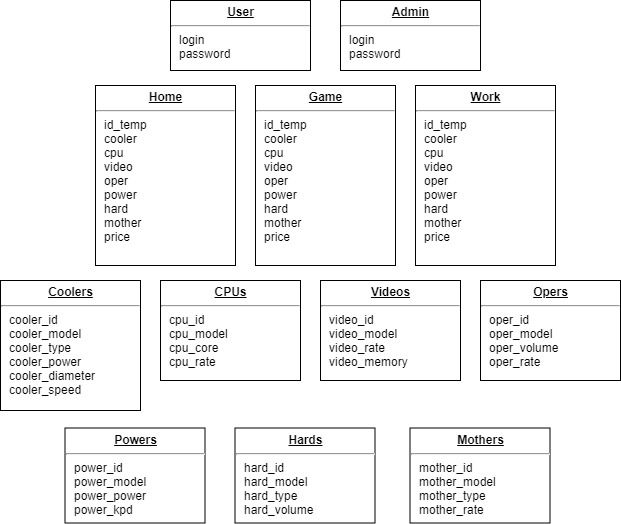


Рисунок 2.7 – Структура базы данных

Подключение базы данных осуществлялось с помощью технологии Ado. Net. ADO.NET предоставляет собой технологию работы с данными, которая основана на платформе .NET Framework. Эта технология представляет нам набор классов для отправки запросов к базам данных, установки подключения и получения ответа от базы данных. С помощью объекта Connection происходит установка подключения к источнику данных. Объект Command позволяет выполнять операции с данными из БД. Объект DataSet предназначен для хранения данных из БД и позволяет работать с ними независимо от БД. И объект DataAdapter является посредником между DataSet и источником данных. Главным образом, через эти объекты и будет идти работа с базой данных.

**3. Тестирование и анализ полученных результатов**

Тестирование – один из важнейших этапов проверки качества разработанного программного обеспечения. В связи с постоянным увеличением сложности разрабатываемых программных систем роль тестирования растет.

Разработка любой программы предполагает наличие ошибок в исходном тексте и борьбу с ними. Ошибки обычно подразделяют на три группы:

* синтаксические ошибки;
* ошибки времени выполнения программы;
* смысловые (логические) ошибки.

Интерфейс программы является понятным для пользователя. Работа самой программы не требует больших затрат ресурсов. В части ввода данных при авторизации и регистрации предусмотрена валидация введенных данных.

В курсовом проекте задействована обработка исключительных ситуаций таким образом, что пользователь будет уведомлен о вводе некорректной информации. В телах методов присутствуют конструкции if/else, которые будут выводить ошибку на экран. Также данные конструкции задействованы в процедурах базы данных.

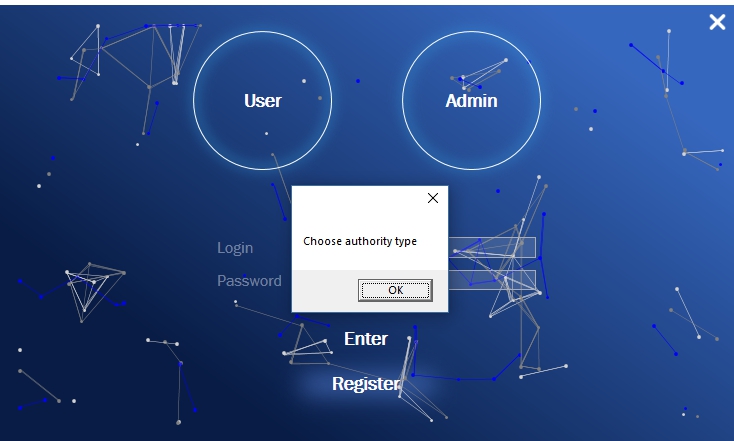


Рисунок 3.1 – Ошибка типа авторизации

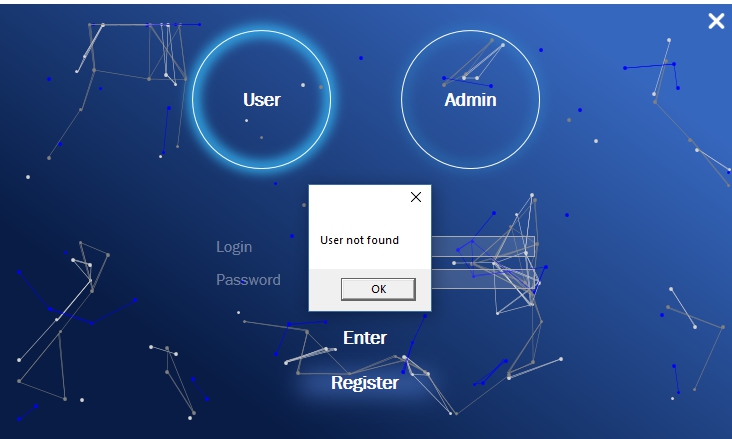


Рисунок 3.2 – Пользователь не найден

На рисунках 3.1 и 3.2 представлены возможные ошибки при авторизации пользователя или администратора. На рисунке 3.3 показана ошибка, появляющаяся при незаполненном поле логина.

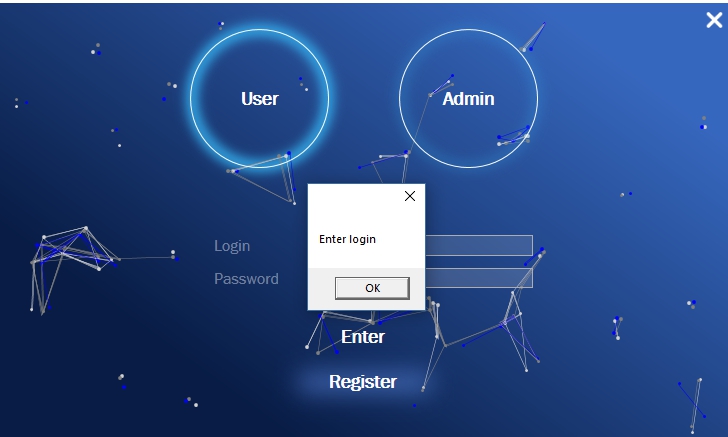


Рисунок 3.3 – Генерация ошибки пустого поля login

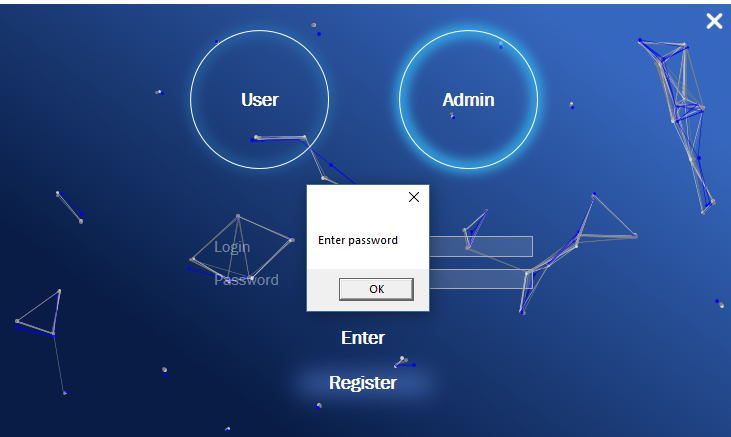


Рисунок 3.4 – Генерация ошибки пустого поля password

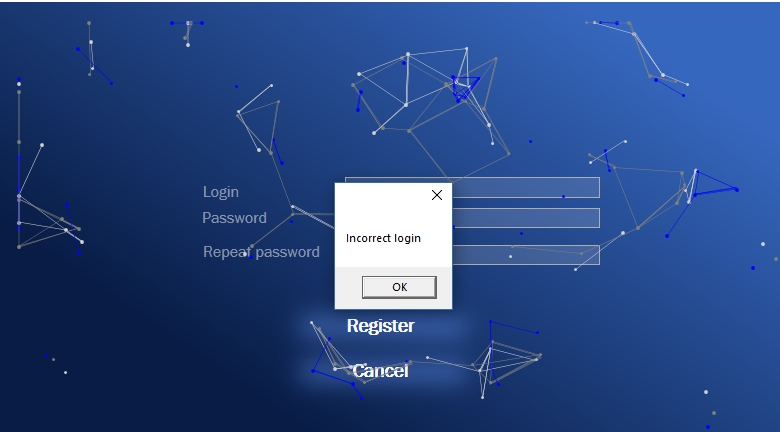


Рисунок 3.5 – Ошибка валидации логина

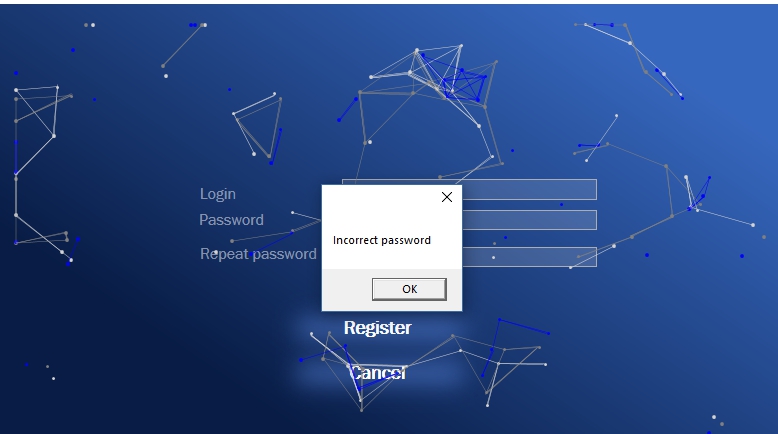


Рисунок 3.6 – Ошибка валидации пароля

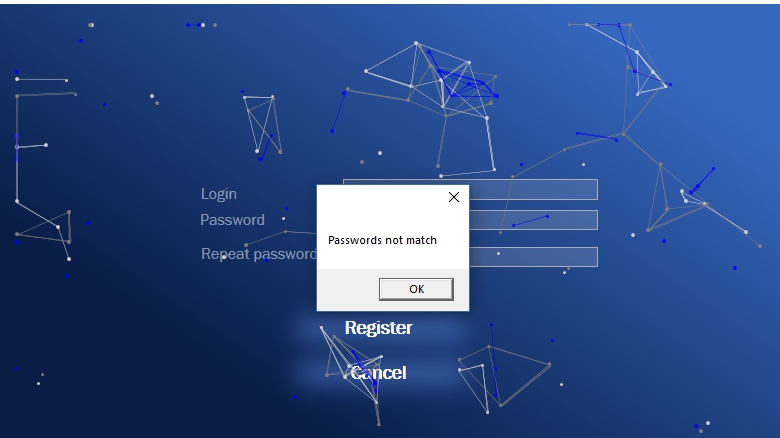


Рисунок 3.7 – Несовпадение паролей при регистрации

**4. Руководство по использованию**

При запуске программы пользователь выбирает тип авторизации, вводит логин и пароль или регистрируется.

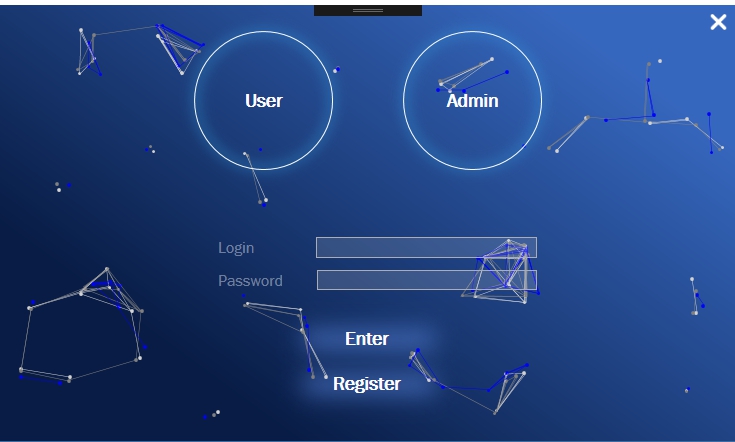


Рисунок 4.1 – Начальное окно приложения

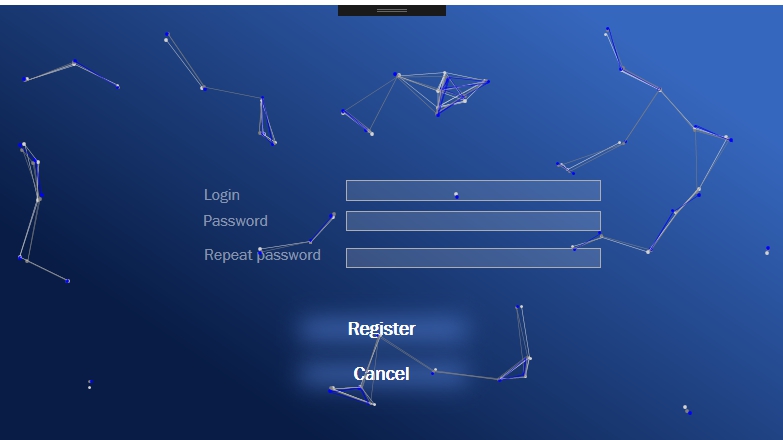


Рисунок 4.2 – Окно регистрации пользователя

После успешной авторизации пользователь выбирает нужную ему функцию.

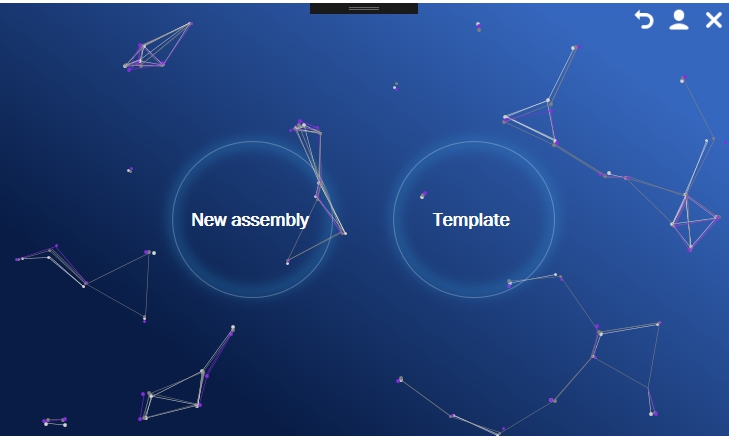


Рисунок 4.3 – Меню функций приложения

При выборе окна конфигурации пользователь видит схематичное изображение системного блока с изображениями компонентов, кликнув по которым можно выбрать нужную модель. После выбора изображение компонента станет цветным.



Рисунок 4.4 – Начальный вид окна конфигуратора

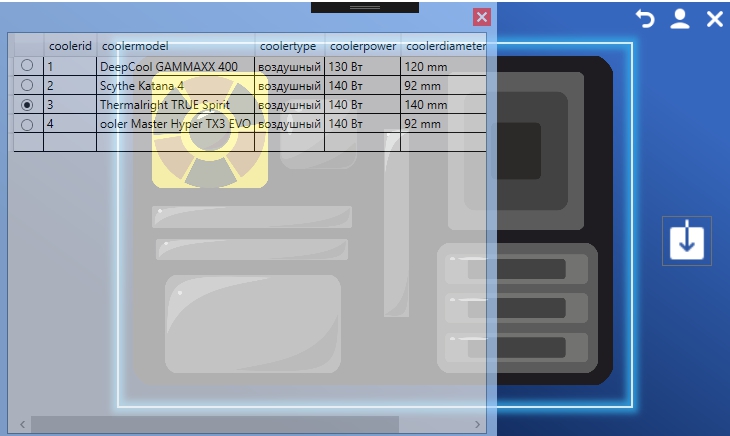


Рисунок 4.5 – Меню выбора модели

Доступные компоненты для сборок следующие: кулер, процессор, блок питания, материнская плана, видеокарта, оперативная память и жесткий диск. Для каждого компонента доступны по 4 модели с различными характеристиками: мощные, слабые и оптимальные.



Рисунок 4.6 – Вид конфигуратора при выборе всех компонентов

При выборе окна шаблонов пользователь выбирает нужный ему тип персонального компьютера и по типу уже может выбрать доступные ему сборки.

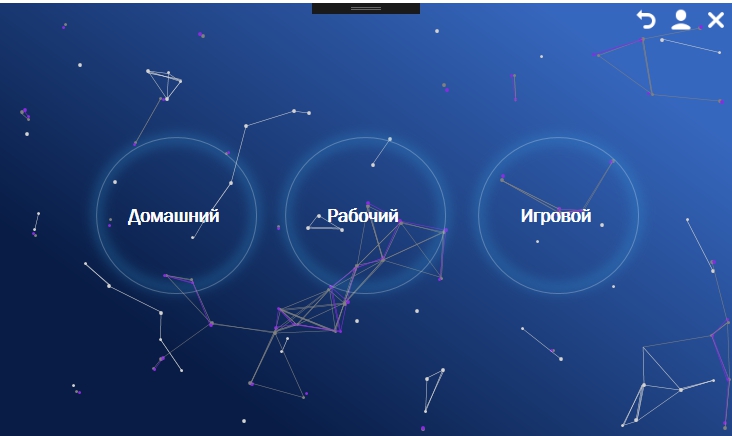


Рисунок 4.7 – Выбор типа сборки

Для каждого типа персонального компьютера изначально доступны 3 вида шаблонов: низкобюджетный, средний и высокобюджетный.

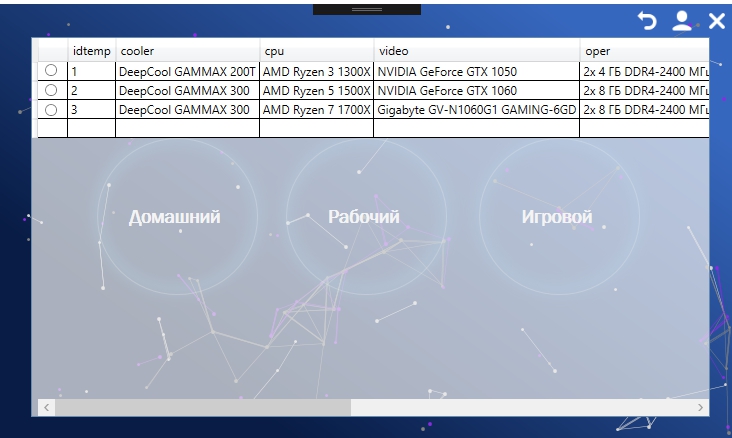


Рисунок 4.8 – Примеры шаблонов игровой конфигурации

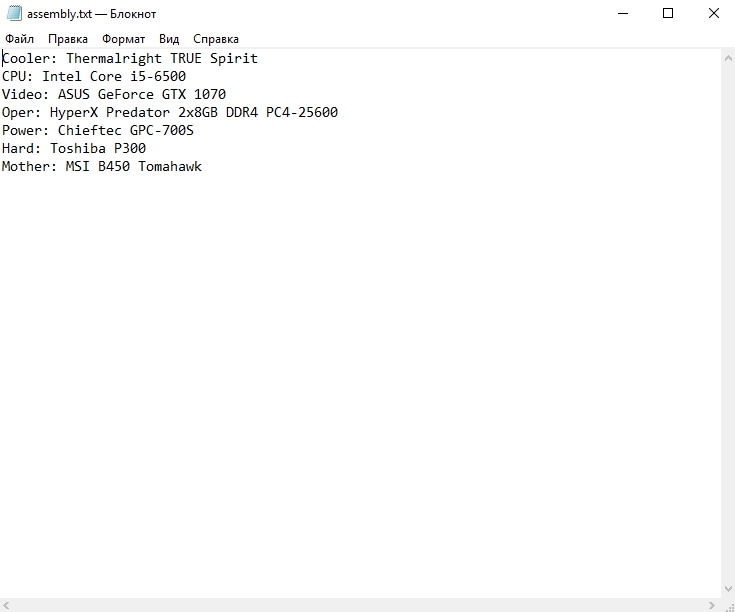


Рисунок 4.9 – Пример файла вывода

После успешной авторизации в качестве администратора появляется функция добавления шаблона с указанием не только компонентов, но и типа сборки.

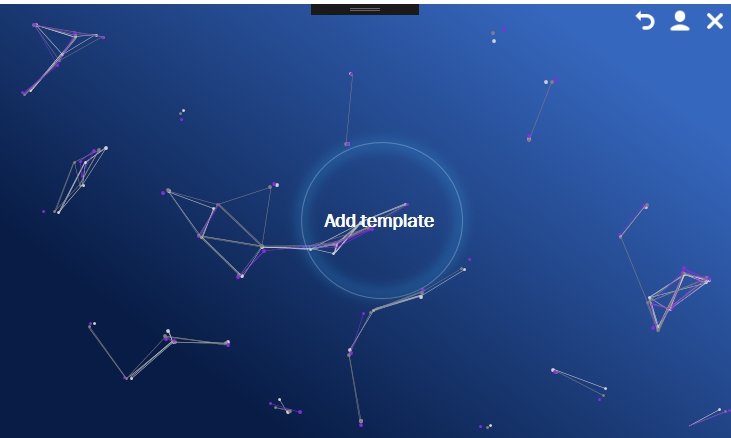


Рисунок 4.10 – Функция добавления шаблона для администратора



Рисунок 4.11 – Вид конфигуратора шаблонов



Рисунок 4.12 – Выбор типа добавляемого шаблона

Кнопки быстрого перехода в верхнем правом углу окна включают в себя кнопку перехода на предыдущее окно, кнопку перехода на экран авторизации и кнопку закрытия приложения. Кнопка закрытия приложения при открытом списке шаблонов закрывает сначала список, а при повторном нажатии — приложение.



Рисунок 4.13 – Кнопки быстрого перехода

**ЗАКЛЮЧЕНИЕ**

При разработке программного средства были изучены способы графического представления сборки персонального компьютера с делением на типы по назначению и выбором отдельно каждого компонента. Программа имеет функционал сохранения полученных и готовых сборок в текстовый файл для дальнейшего использования пользователем в личных целях. Все логины и пароли, а также данные о моделях компонентов, хранятся в базе данных в соответствующих таблицах.

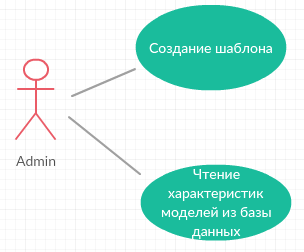
**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Мак-Дональд, М. WPF: Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов. / М. Мак-Дональд. – Москва : Вильямс, 2013. – 1024 с.
2. РАБОТА С БАЗАМИ ДАННЫХ НА ЯЗЫКЕ C#. ТЕХНОЛОГИЯ АDO .NET: учебное пособие / сост. О. Н. Евсеева, А. Б. Шамшев. – Ульяновск: УлГТУ, 2009. – 170 с. – Дата доступа: 04.05.2018
3. Уотсон К., Нейгел К.Visual C# 2010. Полный курс/ Уотсон К., Нейгел К. — СПб.: Питер, 201. — 653 с. – Дата доступа: 13.05.2018
4. Введение в WPF [Электронный ресурс]. – Режим доступа : https://msdn.microsoft.com/ru-ru/library/mt149842.aspx .
5. Кнут, Д.Э. Искусство программирования. В 4 т. / Д.Э. Кнут. – Москва: Вильямс, 2007. – Т. 3. – 832 с.

**ПРИЛОЖЕНИЕ А**

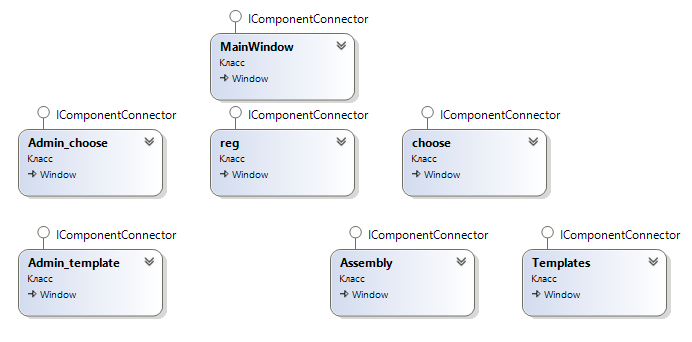
Диаграмма использования





**ПРИЛОЖЕНИЕ Б**

Диаграмма классов



**ПРИЛОЖЕНИЕ В**

Листинг файла Assembly.xaml.cs

Данный файл является главным и отвечает за взаимодействие пользователя с функционалом программного средства и несет основной функционал.

