

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

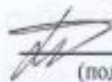
Профиль подготовки: «Вычислительные машины, комплексы, системы и сети»

БАКАЛАВРСКАЯ РАБОТА

на тему

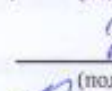
**Информационная система для учета квартир
застройщика**

Студент

 20.06.19
(подпись, дата)

Федосеев Евгений Алексеевич
(ФИО полностью)

Руководитель

 20.06.19
(подпись, дата)

Карамышева Н. С.
(фамилия, инициалы)

Нормоконтролёр

 22.06.19
(подпись, дата)

А.В. Кучин
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 31.05.2019 г. № 38-09-14)

Заведующий кафедрой


(подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой хорошо (протокол заседания ГЭК от 24.06.2019 г. № 16)

Секретарь ГЭК


(подпись)

Е.И. Гурин
(фамилия, инициалы)

Пенза, 2019

Федеральное государственное бюджетное образовательное учреждение
высшего образования

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»

Заведующий кафедрой

 М. А. Митрохин

«__» _____ 2019г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Федосеев Евгений Алексеевич гр. 15BB1 факультета вычислительной техники.

2. Тема работы: «Информационная система для учета квартир застройщика». Тема утверждена приказом ПГУ № 554/о от "29"апреля 2019 г.

3. Руководитель работы к.т.н., доцент каф. ВТ ПГУ Карамышева Н. С.

4. Задание на работу (назначение разработки, исходные данные и т.п.).

Информационная система предназначена для ведения учета свободных, забронированных и купленных квартир застройщика.

5. Перечень подлежащих разработке вопросов.

Требуется произвести анализ предметной области, выбрать системы управления базой данных, выбрать среду разработки, определить интерфейс приложения, спроектировать приложение, реализовать приложение, отладить и провести тестирование разработанного приложения.

6. Календарный график выполнения работы.

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10	01.05.19	
2.	Определение требований к системе	10	05.05.19	
3.	Проектирование приложения	20	12.05.19.	
4.	Написание программы	20	13.05.19	
5.	Отладка и тестирование приложения	10	24.05.19	
6.	Оформление выпускной работы	30	31.05.19	

Дата выдачи задания "29" апреля 2019 г.

Руководитель бакалаврской работы  Карамышева Н. С.
(подпись, дата)

Задание к исполнению принял студент  20.06.19 Федосеев Е. А.
(подпись, дата)

Работу к защите допустить

Декан факультета  Финонова Л. Р.
(подпись, дата)

РЕФЕРАТ

Пояснительная записка содержит 65 страниц формата А4, 34 рисунка, 3 таблицы, 11 источников, 3 приложения.

Объект разработки - информационная система для учета квартир застройщика.

FRAMEWORK 4.6, БАЗА ДАННЫХ, ИНФОРМАЦИОННАЯ СИСТЕМА, SQL-ЗАПРОСЫ, ЛОКАЛЬНОЕ ПРИЛОЖЕНИЕ, WPF.

Целью дипломной работы является разработка информационной системы для учета квартир застройщика, предназначенной для быстрого отслеживания свободных, купленных и забронированных квартир. Это достигается путем заполнения и сбора данных в MS SQL, введенных самим пользователем и их визуальное отображение в виде обычных числовых данных, таблиц.

В результате проведенной работы создан проект, протестированы программные средства, удовлетворяющие поставленной задаче.

В пояснительной записке приведен анализ предметной области, обзор технологии разработки и отладки программы.

Графическая часть содержит ER-диаграмму БД и UML-диаграммы программы.

					ПГУ 1.090301.27.001 ПЗ			
Изм	Лист	№ докум.	Подпись	Дата	Информационная система для учета квартир застройщика Пояснительная записка	Литер.	Лист	Листов
Разработ.		Федосеев Е. А.				У	4	65
Проверил		Карамышева Н.С				ФВТ, каф. ВТ Гр. 15ВВ1		
Н. контр.		Кучин А. В.						
Утвердил		Митрохин М.А.						

Содержание

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ	6
ВВЕДЕНИЕ.....	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.1 Описание предметной области	9
1.2 Обзор существующих решений.....	10
1.3 Выбор средств разработки	10
1.4 Архитектура БД.....	11
1.5 Используемая технология для создания ИС	14
2 ОПИСАНИЕ ТРЕБОВАНИЙ К СИСТЕМЕ.....	17
2.1 Требования к надежности	17
2.2 Требования к пользователю ИС	17
2.3 Требования к информационной и программной совместимости	17
2.4 Требования к программной документации	18
3 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ	19
4 РЕАЛИЗАЦИЯ ПРОЕКТА.....	21
4.1 Процесс разработки	21
4.1.1 Реализация БД	21
4.1.2 Реализация интерфейса приложения	23
4.2 Отладка и тестирование	28
5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	31
ЗАКЛЮЧЕНИЕ	36
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	37
ПРИЛОЖЕНИЕ А Листинг программы	38
Приложение А.1 Файл Login.xaml.cs	39
Приложение А.2 Файл MainWindow.xaml.cs	40
Приложение А.3 Файл Magnifier.xaml.cs	42
Приложение А.4 Файл ContractConclusion.xaml.cs	47
Приложение А.5 Файл App.config.....	52
ПРИЛОЖЕНИЕ Б Рисунки и диаграммы.....	53
ПРИЛОЖЕНИЕ В Презентация	56

					ПГУ 1.090301.27.001 ПЗ			
Изм	Лист	№ докум.	Подпись	Дата				
Разработ.	Федосеев Е. А.				Информационная система для учета квартир застройщика Пояснительная записка	Литер.	Лист	Листов
Проверил	Карамышева Н.С					У	5	65
						ФВТ, каф. ВТ Гр. 15ВВ1		
Н. контр.	Кучин А. В.							
Утвердил	Митрохин М.А.							

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

ВКР – Выпускная квалификационная работа.

ИС – Информационная система.

ПО – Программное обеспечение.

СУБД – Система управления базами данных.

БД – База данных.

WPF - Windows Presentation Foundation.

ПКМ – Правая кнопка мыши.

					ПГУ 1.090301.27.001 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

ВВЕДЕНИЕ

В настоящее время жизнь человека полностью зависит от различного рода информации, для обработки и управления которой требуются создания огромного количества баз данных различного назначения.

Базы данных — это совокупность структур, предназначенных для хранения больших объемов информации и программных модулей, осуществляющих управление данными, их выборку, сортировку и другие подобные действия.

Информация базы данных хранится в одной или нескольких таблицах. Любая таблица с данными состоит из набора однотипных записей, расположенных друг за другом. Они представляют собой строки таблицы, которые можно добавлять, удалять или изменять.

Создав одну таблицу, вы уже получаете полноценную базу данных. Однако в реальной жизни структуры баз данных, а соответственно и способы их создания, намного сложнее.

При создании большого количества таблиц и заполнением большим объемом информации возникает избыточность информации, в которой иногда трудно сориентироваться и выбрать нужные сведения.

Для решения подобных проблем применяются информационные системы различных направлений. Они стали неотъемлемой частью начиная от отрасли и до отдельного предприятия.

К настоящему времени накоплен значительный опыт проектирования БД, предназначенных для управления производством, это позволяет сделать процесс создания БД более эффективным.

Целью данной выпускной квалификационной работы является разработка информационной системы для учета квартир застройщика, который сможет отслеживать свободные, забронированные и купленные квартиры, а также отслеживания договоров с клиентами.

					ПГУ 1.090301.27.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

Для достижения поставленной цели были поставлены следующие задачи:

- провести анализ предметной области и постановка задачи на разработку программного средства для поиска, подбора, бронирования и покупки квартир застройщика работая с БД;
- разработка схемы БД;
- реализация разработанной схемы в СУБД (MS SQL Server);
- создание форм для работы с БД;
- автоматизация работы с созданной БД;
- тестирование ИС.

					ПГУ 1.090301.27.001 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Описание предметной области

Разрабатываемая информационная система предназначена для функционирования в организации застройщика ООО «Термодом» по жилому комплексу «Лазурный».

Строительный холдинг «Термодом» является одним из лидеров жилищного многоэтажного строительства в Пензенской области. Развитие на рынке начиналось с индивидуального и малоэтажного строительства и дошло до проекта комплексного освоения территории – микрорайона Город Спутник.

Программное обеспечение данной предметной области это процесс работы с базой данных через приложение. Анализ, управление и добавление информации напрямую через СУБД не является самым удобным решением. В результате возник вопрос о создании системы сбора информации о свободных, забронированных и купленных квартир застройщика.

Необходимо разработать структуру таблиц, которые будут хранить информацию о тех или иных домах и квартирах. По результатам поиска система должна показать квартиры удовлетворяющих запросу, а также возможность добавить клиента в БД, забронировать, отменить бронирование и покупку квартиры.

Актуальность разработки информационной системы заключается в ускорении и упрощении работы по учету квартир, клиентов, добавлении новых клиентов, хранении информации о забронированных квартирах, договорах на бронирование, отмены бронировании или покупку квартир.

Выбор оптимального решения требует проведения анализа предметной области и определения некоторых ключевых моментов, на которых в дальнейшем будет основываться проектирование и разработка приложений.

					ПГУ 1.090301.27.001 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

1.2 Обзор существующих решений

На сегодняшний момент существует несколько приложений, позволяющих пользователям просматривать и приобретать квартиры от застройщиков. В данном разделе будут рассмотрены некоторые существующие решения, наиболее пользующиеся спросом.

Bpm real estate. Программное обеспечение, которое основано на управлении всем комплексом задач застройщиков и агентств недвижимости. Имеет хороший дизайн, гибкие настройки, широкий выбор инструментов и большая база данных для совершения сделок. Одним из основных минусов приложения является то, что оно имеет платную подписку.

Также, стоит отметить то, что программное обеспечение ориентировано на работу с листингами и управление счетами и платежами.

Циан. Программный продукт, позволяющий просматривать недвижимость не только от застройщика, но и агентств и риелторов. На данный момент — это крупнейший сервис для покупателей и арендаторов жилья, с самой большой базой объявлений.

Следует отметить, что «Циан» имеет программный продукт на разные платформы, такие как Windows, Android и iOS.

Самым большим минусом таких сервисов является не актуальная информация, она обновляется в течении нескольких дней или даже нескольких месяцев.

1.3 Выбор средств разработки

В ходе анализа предметной области были проанализированы различные средства, технологии и языки программирования.

Таким образом был выбран ряд программных средств, использованных в данной работе:

- локальный сервер MS SQL Server;
- СУБД MS SQL;
- Visual Studio – как среда разработки.

					ПГУ 1.090301.27.001 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

Использованные языки программирования:

- SQL для работы с базой данных;
- C#;
- XAML язык разметки, для создания интерфейса приложения.

Выбор средств разработки исходил из функциональных возможностей, высокой производительности и бесплатного использования.

1.4 Архитектура БД

Прежде чем начать разработку ПО необходимо изучить архитектуру создания базы данных.

Проектирование БД осуществляется на основе трехуровневой архитектуры. Наглядно это представлено на рисунке 1.

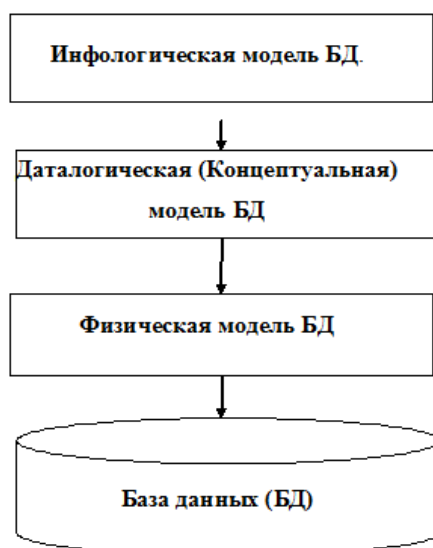


Рисунок 1 - Трехуровневая архитектура БД

1. **Инфологическая модель БД.** Модель описываемая данные предметной области на естественном языке. В ней определена логика отношений, и она является человекоориентированной.

Цель инфологической модели является обеспечить наиболее естественный для человека способ сбора и предоставления той или иной информации для предметной области, которая будет храниться в создаваемой базе данных. Поэтому инфологическую модель в большинстве случаев строят

по аналогии с естественным языком. Основой данной модели являются сущности, связи между ними и их свойства. Поэтому необходимо четко поставить цель данной БД, а также установить, какую информацию получает пользователь в результате работы с ИС.

2. **Даталогическая (концептуальная) модель БД.** Данная модель является логической организацией данных и их взаимосвязь. Структура данных выполняется в соответствии выбранной СУБД, которая строится на основе выбранной модели представления данных: иерархическая, сетевая или реляционная. На данном этапе создается структура данных и организовывается связь между объектами.

На этом этапе необходимо правильно выделить сущности и описать их атрибуты. Ошибка может обернуться значительными потерями времени и сил в дальнейшем. Данная модель описывает логику организации данных. Даталогическое проектирование заключается в проектировании логической структуры БД. Таким образом, главным отличием даталогической модели от инфологической состоит в том, что инфологическая модель хранит в себе всю информацию о предметной области, необходимую и достаточную для проектирования БД, но она не привязана к определенной СУБД.

Даталогическое проектирование состоит из следующих этапов:

- 1) определение таблиц;
- 2) определение полей таблиц;
- 3) определение типов данных в соответствии с выбранной СУБД;
- 4) определение длины каждого поля таблиц;
- 5) определение обязанности каждого поля;
- 6) определение индексации каждого поля.

3. **Физическая модель БД.** Состоит из структуры, реализованной в среде, выбранной СУБД.

Также при создании БД важно учитывать следующие параметры:

					ПГУ 1.090301.27.001 ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

- целостность БД- правильность данных в любой момент времени;
- избыточность БД- дублирование данных в нескольких таблицах (избыточность влечет за собой противоречивость данных, ее стараются исключить введением кодовых полей).

Цель проектирования БД состоит в сокращение избыточных данных. Эта цель достигается с помощью теории нормализации баз данных.

Нормализация – это разбиение таблиц на две или более, обладающими лучшими свойствами при внесении или изменении данных. В результате получается БД, в которой каждый факт появляется лишь в одном месте, что исключает избыточность информации.

В качестве СУБД было выбран программный продукт – MS SQL Server. Языком создания и управления БД является SQL. Это универсальный компьютерный язык, который предназначен для создания, модификации и управления таблицами и данными в реляционных базах данных. Прежде всего, SQL это информационно – логический язык, применяемый для изменения, описания и извлечения данных хранимых в БД.

Рассмотрим MS SQL Server. Эта система является одной из наиболее популярных систем управления БД (СУБД) в мире. Данная СУБД подходит для реализации как небольших проектов, так и больших высоконагруженных проектов.

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.

Центральным аспектом в MS SQL Server, как и в любой СУБД, является база данных. База данных представляет хранилище данных, организованных определенным способом. Для хранения и администрирования баз данных

					ПГУ 1.090301.27.001 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

применяются системы управления базами данных (database management system) или СУБД (DBMS). И как раз MS SQL Server является одной из такой СУБД.

Написанное программистом приложение не напрямую связывается с БД, а через сервер. На рисунке 2 показана архитектура взаимодействия клиента (созданное приложение) и сервера. Особенностью данной архитектуры является обработка запроса на стороне сервера. В данной ситуации предъявляются дополнительные требования к его производительности, но и в тоже время данная архитектура является очень надежной.

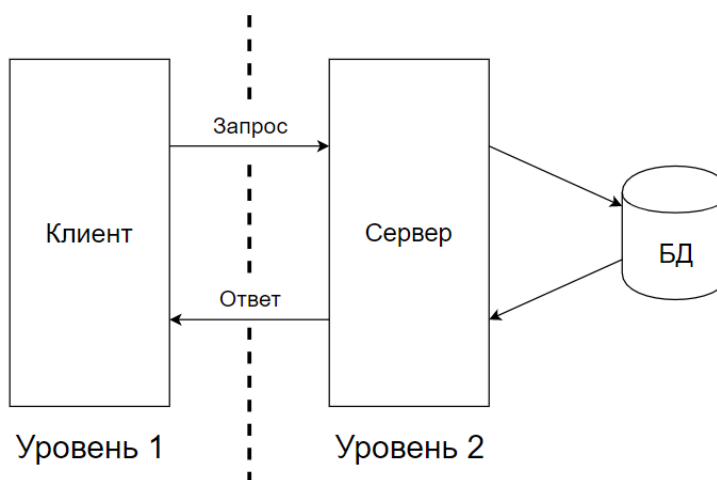


Рисунок 2 – Двухзвенная архитектура

1.5 Используемая технология для создания ИС

Возможности программирования на C# настольных Windows – приложений безграничны. При выборе средств разработки между Windows Forms и Windows Presentation Foundation (WPF). Был выбран WPF в качестве графической презентационной подсистемы в виде автономного настольного Windows – приложения.

Разработчик такого приложения получает возможности мощного графического ядра и программного интерфейса для отрисовки различных графических элементов. В отличие от WinForms, в которой за отрисовку элементов управления и графики отвечают User32 и GDI+, то приложения WPF основаны на DirectX.

Одна из важных особенностей — это язык разметки интерфейса XAML, с помощью которого можно создавать насыщенный графический интерфейс, используя объявление интерфейса или код под управлением языков C# и VB.NET.

Схема архитектуры WPF состоит из двух уровней, представлено на рисунке 3.

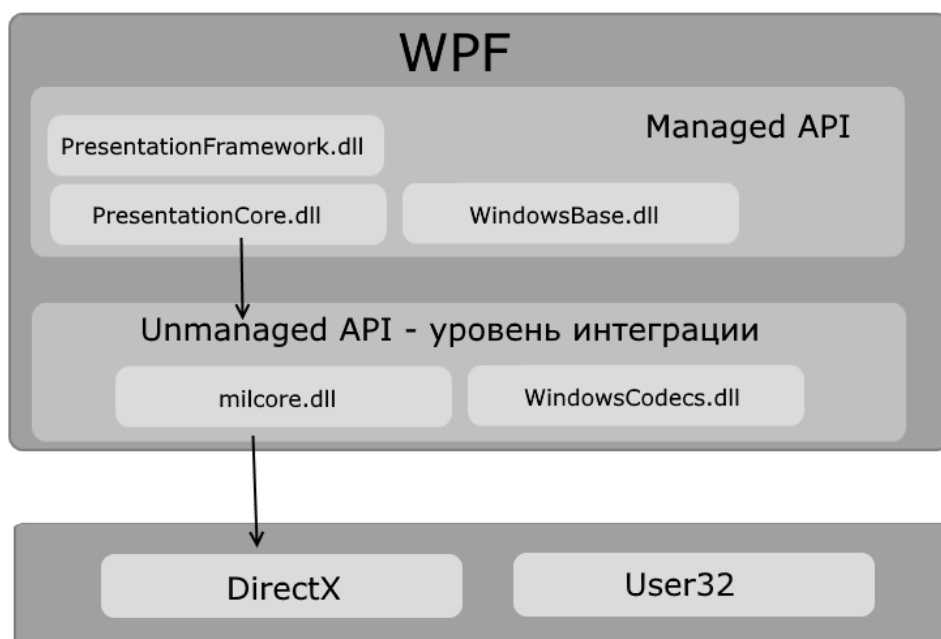


Рисунок 3 - Схема архитектуры WPF

В первый уровень входят: managed API и unmanaged API (уровень интеграции с DirectX).

Managed API (управляемый API-интерфейс) содержит код, исполняемый под управлением общезыковой среды выполнения .NET - Common Language Runtime. Этот API описывает основной функционал платформы WPF и состоит из следующих компонентов:

- **PresentationFramework.dll.** Содержит все основные реализации компонентов и элементов управления, которые можно использовать при построении графического интерфейса;
- **PresentationCore.dll.** Содержит все базовые типы для большинства классов из PresentationFramework.dll;

- **WindowsBase.dll**. Содержит ряд вспомогательных классов, которые применяются в WPF, но могут также использоваться и вне данной платформы.

Интерфейс Unmanaged API используется для интеграции вышележащего уровня с DirectX:

- **milcore.dll**. Собственно обеспечивает интеграцию компонентов WPF с DirectX. Данный компонент написан на неуправляемом коде (C/C++) для взаимодействия с DirectX;
- **WindowsCodecs.dll**. Библиотека, которая предоставляет низкоуровневую поддержку для изображений в WPF.

На втором уровне находятся компоненты операционной системы и DirectX, которые производят визуализацию компонентов приложения, либо выполняют прочую низкоуровневую обработку. С помощью низкоуровневого интерфейса Direct3D, который входит в состав DirectX, происходит трансляция.

Также на этом уровне находится библиотека user32.dll. И хотя выше говорилось, что WPF не использует эту библиотеку для рендеринга и визуализации, однако для ряда вычислительных задач (не включающих визуализацию) данная библиотека продолжает использоваться.

					ПГУ 1.090301.27.001 ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

2 ОПИСАНИЕ ТРЕБОВАНИЙ К СИСТЕМЕ

2.1 Требования к надежности

Основными требованиями к надёжности системы являются: бесперебойное функционирование системы и защита от неправильного ввода данных в систему.

В системе должен быть реализован контроль над корректностью вводимой информации. Ввод данных в информационную систему должен осуществляться только после исправления пользователем ошибок ввода. При вводе новой информации в случае ошибки во входных параметрах, должны выдаваться соответствующие сообщения и отменяться операция ввода с целью сохранения целостности базы данных. Система не должна допускать дублирования ввода уже имеющейся информации и фиксировать информацию, пока она не введена по всем полям в базе данных.

2.2 Требования к пользователю ИС

Данной информационной системой может пользоваться даже не уверенный пользователь ПК, так как у системы простой и понятный интерфейс, простота в вводе данных и их использовании. Система предназначена как для риелторов застройщика, так и для руководства с целью отслеживания реализации жилья.

2.3 Требования к информационной и программной совместимости

Информационная система для учета квартир застройщика должна быть в среде разработки Visual Studio на языке программирования C#.

На рабочих станциях сотрудников должно быть установлено:

- локальный сервер СУБД Microsoft SQL Server 2016;
- среда Microsoft .NET Framework 4.6;
- операционная система не ниже Windows 10;
- исполняемый файл информационной системы о квартирах застройщика.

					ПГУ 1.090301.27.001 ПЗ	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

2.4 Требования к программной документации

В состав пояснительной записки дипломного проекта должны быть включены следующие разделы и документы:

- техническое задание;
- описание программы;
- программа и методика испытаний;
- описание применения.

					ПГУ 1.090301.27.001 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

3 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

В ходе реализации Window desktop приложения, изначально была выбрана архитектура, на основе которой будет строиться приложение. Грамотно выбранная архитектура позволит в дальнейшем «безболезненно» вносить изменения, добавляя или улучшая функционал.

В данной работе будет использоваться архитектура MVC (Model-View-Controller). Его концепция заключается в отделении логики представления данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

- Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.
- Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.
- Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

На рисунке 4 представлена примерная схема MVC архитектуры приложения.

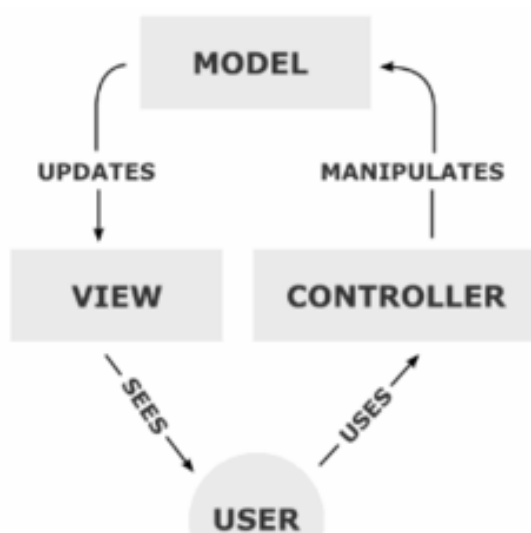


Рисунок 4 - Схема MVC

Что же это дает нам в конечном итоге?

- Гибкость разработки;
- Тестирование;
- Разграничение логики.

Приложение должно обладать интуитивно понятным интерфейсом и позволять производить различные действия над данными базы. Запросы к базе данных должны конструироваться на основании того или иного действия пользователя и представлять из себя команды на языке SQL. Затем сформированные запросы следует отправлять на сервер для дальнейшей их передачи в базу. Возможные варианты взаимодействия пользователя с базой данных представлены на UML – диаграмме вариантов использования, показано в приложении Б рисунок 32.

Для реализации этих возможностей необходимо будет разместить в окне приложения кнопки, меню или предусмотреть иные интуитивно понятные способы осуществления указанных действий.

Действия пользователя должны вызывать соответствующие обработчики. Поскольку выбранная СУБД поддерживает возможность взаимодействия с базой данных посредством выполнения запросов на языке SQL, в обработчиках должны конструироваться команды, в которые подставляются названия таблиц и/или столбцов в соответствии с синтаксисом той или иной конкретной команды. Схема алгоритма обработки запросов, представлено в приложении Б на рисунке 33. Отправляемые приложением запросы и обработка результата можно изобразить с помощью UML – диаграммы состояний, представлено в приложении Б на рисунке 34.

					ПГУ 1.090301.27.001 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

4 РЕАЛИЗАЦИЯ ПРОЕКТА

4.1 Процесс разработки

4.1.1 Реализация БД

Необходимо спроектировать базу данных для хранения информации о квартирах, клиентах и сделок с ними. Анализируя данную тему БД можно составить описание бизнес – процессов информационной системы, представлено в таблице 1.

Таблица 1 - Описание бизнес-процессов информационной системы

п/п	Бизнес-процесс	Исполнитель	Входные данные		Выходные данные		Ограничения
			Поставщик	Содержание	Потребитель	Содержание	
1	Бронирование квартиры	Риэлтор	Контактное лицо владельца комплекса	Информация о квартире, контактном лице и о риэлторе	Клиент	Информация о квартире, контактном лице, риэлторе и о клиенте	Забронированную квартиру нельзя забронировать повторно. Купить квартиру может только клиент, забронировавший эту квартиру.
			Клиент	Данные клиента	Контактное лицо владельца комплекса	Информацию о за бронировании и квартиры	
2	Отмена бронирования	Риэлтор	Контактное лицо владельца	Информация о квартире	Клиент	Информация, что квартира больше не забронирована	Невозможно повторно отменить.
3	Покупка квартиры	Риэлтор	Контактное лицо владельца комплекса	Информация о квартире, контактном лице и о риэлторе	Клиент	Информация о квартире, контактном лице, риэлторе и о клиенте	Купленную квартиру повторно купить нельзя.
			Клиент	Данные клиента	Контактное лицо владельца комплекса	Информацию о за бронировании и квартиры	

В таблице 2 представлено описание типов сущностей.

В таблице 3 показывает связи между типами сущностей.

Таблица 2 – Описание типов сущностей

№	Наименование	Описание
1	Квартиры	id, номер этажа, номер квартиры, номер подъезда, номер дома, кол-во комнат, общ. площадь, жил. площадь, цена за квм, статус, стоимость, описание
2	Дома	id, код комплекса, тип постройки, адрес, кол-во этажей, студий, однокомнатных, двухкомнатных, трехкомнатных, четырехкомнатных, описание
3	Комплекс	id, название, адрес, код владельца, описание
4	Владелец	id, название строительной компании, контактное лицо, телефон, описание
5	Сделки	id, дата, код покупателя, код комплекса, описание
6	Пользователь	id, имейл, хэш, ФИО, логин, телефон, описание
7	Покупатель	id, ФИО паспорт, адрес, телефон, описание

Таблица 3 – Описание типов связей между типами сущностей предметной области

№	Тип сущности 1	Тип сущности 2	Описание	относительность	Обязательность	Ограничения
1	Квартиры	Дома	Квартира входит в состав дома	1:M	Обязательно	В одном доме не могут повторяться номера квартир
2	Дома	Комплекс	Дом входит в состав комплекса	1:M	Обязательно	
3	Покупатели	Пользователи	Покупатель совершает сделку с пользователем (сотрудником)	1:1	Обязательно	Сделка заключается между одним покупателем и одним сотрудником
4	Владельцы	Комплекс	Владелец является контактным лицом строительной организации комплекса	1:1	Обязательно	

Диаграмма базы данных показана в приложении Б – рисунок 31.

4.1.2 Реализация интерфейса приложения

Основным элементом использующийся в интерфейсе для отображения данные БД, является таблица.

Для работы с приложением были реализованы следующие экраны:

- Экран авторизации сотрудников;
- Экран регистрации сотрудников
- Интерфейс главного экрана, в котором будут происходить взаимодействия с БД;

Экрана авторизации пользователей.

Существует две ситуации исходя из которых пользователь может попасть на этот экран. Первая – это когда пользователь заходит в приложение впервые: для работы в данном программном продукте ему необходимо авторизоваться. И вторая ситуация – когда во время работы необходимо сменить пользователя.

Стоит отметить, что при вводе пользователем пароля, символы закрашиваются точками, что необходимо для безопасности, как показано на рисунке 5.

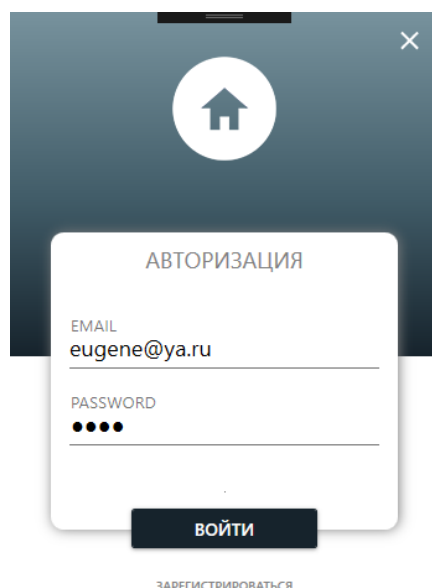


Рисунок 5 – Экран авторизации пользователя

В программе реализована навигация по страницам с помощью выплывающего меню слева, показано на рисунке 6.

					ПГУ 1.090301.27.001 ПЗ	Лист
						23
Изм.	Лист	№ докум.	Подпись	Дата		

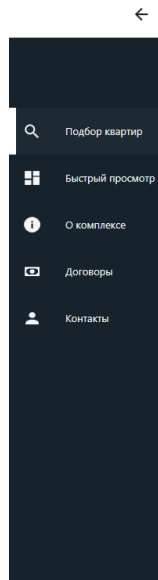


Рисунок 6 – Меню программы

Экран с выбранной вкладкой «Подбор квартир».

На данной странице осуществляется поиск квартир по необходимым параметрам. В таблице отображается вся необходимая информация, представлено на рисунке 7.

Каждую из квартир можно добавить в сравнение и отобразить только их. Также квартиры можно забронировать на того или иного покупателя, отменить бронь или продать. Все сведения будут видны на странице «Договоры», представлено на рисунке 8.

Адрес	№ этажа	№ квартиры	Кол-во комнат	Общ площадь	Жил площадь	Цена за кв.м	Стоимость	Статус
Изумрудная, 1	1	1	1	40.7	18.1	38000	1546600	свободна
Изумрудная, 1	1	7	1	40.7	18.1	38000	1546600	свободна
Изумрудная, 1	2	8	1	40.9	18.1	38000	1554200	свободна
Изумрудная, 1	2	14	1	40.9	18.2	38000	1554200	свободна
Изумрудная, 1	3	15	1	40.9	18.2	38000	1554200	свободна
Изумрудная, 1	3	21	1	40.9	18.2	38000	1554200	свободна
Изумрудная, 1	4	22	1	40.9	18.2	38000	1554200	свободна
Изумрудная, 1	4	28	1	40.9	18.1	38000	1554200	свободна
Изумрудная, 1	5	29	1	40.8	18.1	38000	1550400	свободна
Изумрудная, 1	5	35	1	40.8	18.1	38000	1550400	свободна
Изумрудная, 1	6	36	1	40.8	18.1	39000	1591200	свободна
Изумрудная, 1	6	42	1	40.9	18.1	39000	1595100	свободна

Рисунок 7 – Страница "Подбор квартир"

Экран с выбранной вкладкой «Договоры».

Данная страница показывает информацию о клиенте, квартире, статусе договора и номер сотрудника, который заключил тот или иной договор (бронирование, отмена брони или приобретение квартиры), как показано на рисунке 8.

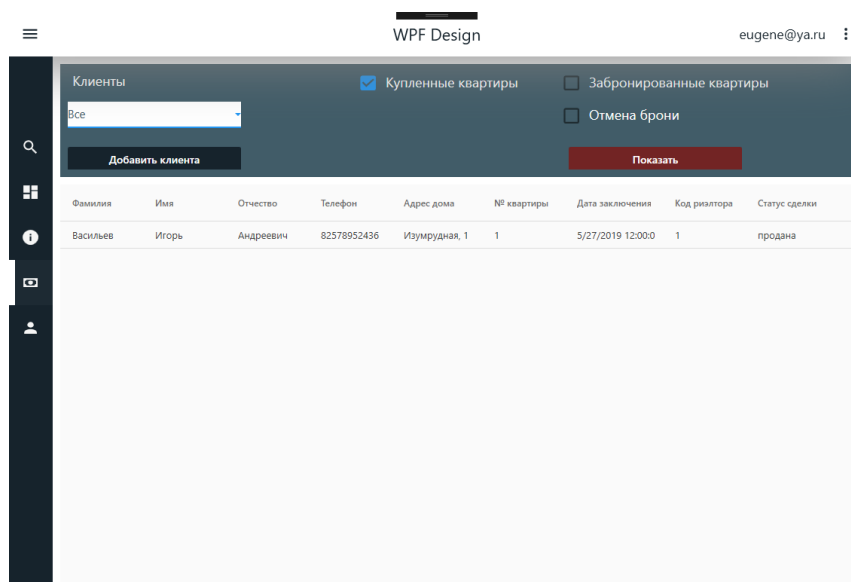


Рисунок 8 – Страница "Договоры"

Экран с выбранной вкладкой «Быстрый просмотр».

Представляет собой отображение кнопок для перехода на новые страницы, связанные с тем или иным домом, показано на рисунке 9. После нажатия на одну из кнопок, осуществляется переход на новую страницу. На ней отображается информация о выбранном доме, как показано на рисунке 10, если перейти на вкладку «Планировка», откроется страница где можно выбрать подъезд и этаж для просмотра планировки и информации (при наведении компьютерной мыши на ту или иную квартиру), показано на рисунке 11.

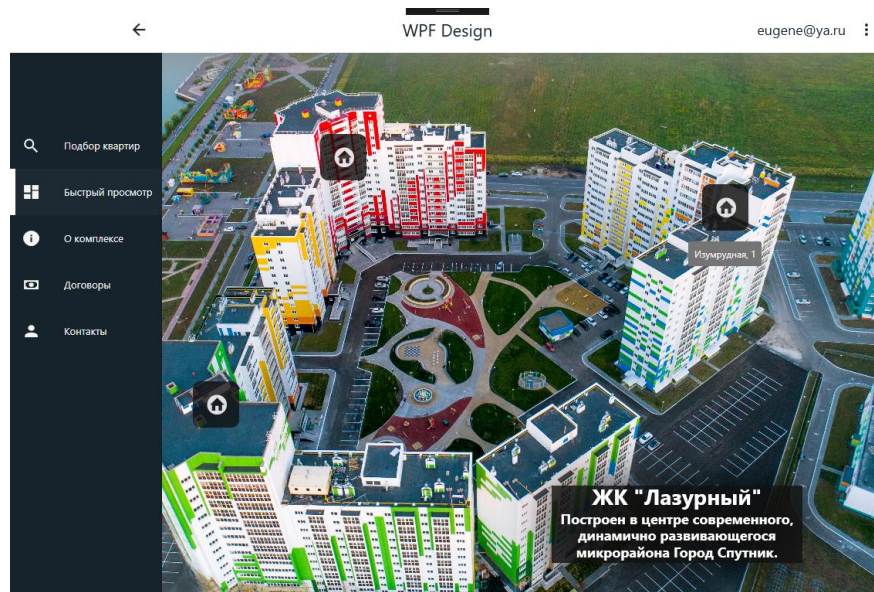


Рисунок 9 – Интерфейс страницы «Быстрый просмотр»

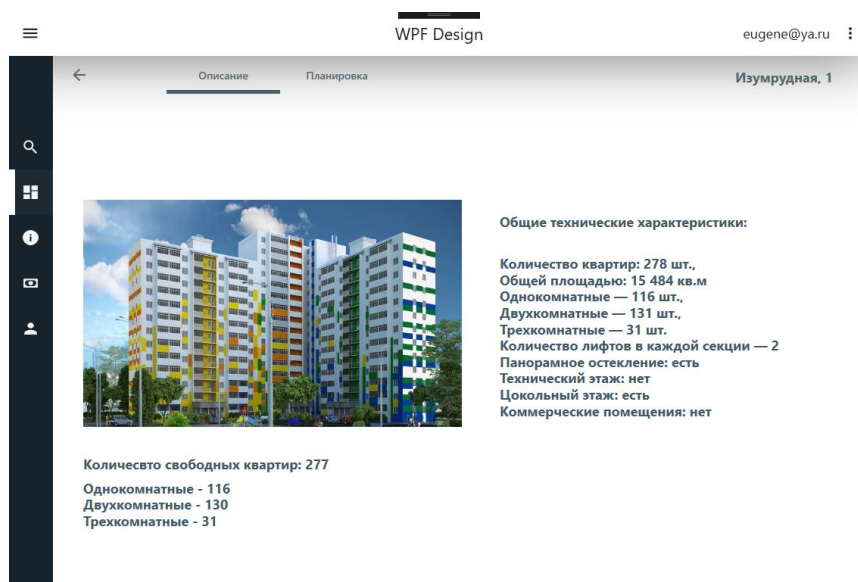


Рисунок 10 – Описание выбранного дома



Рисунок 11 – Планировка с выбранного подъезда и этажа

					ПГУ 1.090301.27.001 ПЗ	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		

Экран с выбранной вкладкой «О комплексе».

На данной странице, продемонстрировано на рисунке 12, показана информация и преимущества комплекса.

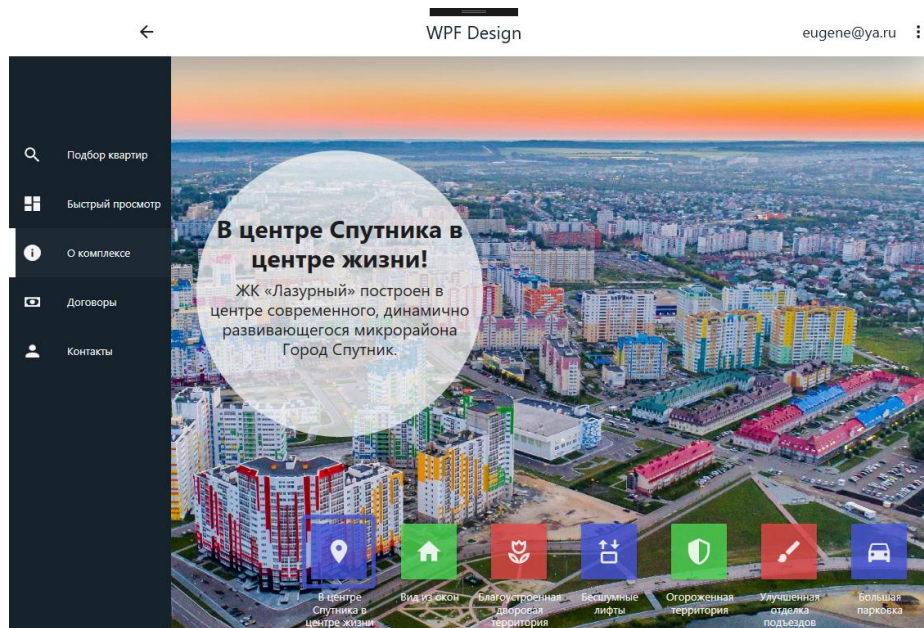


Рисунок 12 – Интерфейс страницы «О комплексе»

Экран с выбранной вкладкой «Контакты».

Данная страница состоит из двух таблиц. В первой отображается информация о контактных лицах данной строительной фирмы. Во второй информация о сотрудниках (риелторах), показано на рисунке 13.

Рисунок 13 – Страница «Контакты»

4.2 Отладка и тестирование

Программа написана и отлажена с использованием среды программирования Visual Studio 2017 и Microsoft SQL Server Management Studio 17.

Отладка проводилась с помощью таких инструментов, как точка останова, выполнение кода по шагам, анализа содержимого переменных.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы. В ходе тестирования было выявлено и исправлено множество проблем, связанных с синтаксисом конструируемых в программе SQL-запросов, взаимодействием приложения с базой данных, а также непосредственным доступом к базе данных.

Произведем тестирование страницы «Подбор квартир». Зададим параметры поиска: номер дома – Изумрудная 1; тип квартиры – двухкомнатная; свободные квартиры; цена от 1.5 до 2 миллионов рублей. И нажмем на кнопку «Показать», результат представлен на рисунке 14.

The screenshot shows a WPF application window titled 'WPF Design' with a user email 'eugene@ya.ru'. The interface features a search bar with filters: 'Номер дома' (House Number) set to 'Изумрудная, 1', 'Цена' (Price) from '1500000' to '2000000', 'Общ площадь' (Total Area) from '0' to '120', and 'Номер этажа' (Floor) from '1' to '20'. Under 'Тип квартиры' (Apartment Type), 'Двухкомнатные' (Two-bedroom) is selected, and 'Свободные квартиры' (Free apartments) is checked. Buttons for 'Показать сравнение' (Show comparison), 'Сбросить' (Reset), and 'Показать' (Show) are present. Below the filters is a table with the following data:

Адрес	№ этажа	№ квартиры	Кол-во комнат	Общ площадь	Жил площадь	Цена за кв.м	Стоимость	Статус
Изумрудная, 1	2	9	2	51.8	26.2	38500	1994300	свободна
Изумрудная, 1	3	16	2	51.7	26.4	38500	1990450	свободна
Изумрудная, 1	5	30	2	51.8	26.3	38500	1994300	свободна
Изумрудная, 1	1	190	2	51.4	26.1	38000	1953200	свободна
Изумрудная, 1	2	197	2	51.6	26.2	38000	1960800	свободна

Рисунок 14 – Тест по параметрам поиска

Далее добавим квартиры под номерами 16 и 30 в сравнение. Для этого по выбранной строке нажмем ПКМ и выберем нужный пункт из контекстного меню, представлено на рисунке 15. При добавлении в сравнение строка закрашивается в красный цвет, тем самым выделяет ее. Чтобы убедиться, находятся ли квартиры в сравнении, необходимо нажать на кнопку «Показать сравнение», показано на рисунке 16.

Адрес	№ этажа	№ квартиры	Кол-во комнат	Общ площадь	Жил площадь	Цена за кв.м	Стоимость	Статус
Изумрудная, 1	2	9	2	51.8	26.2	38500	1994300	свободна
Изумрудная, 1	3	16	2	51.7	26.4	38500	1990450	свободна
Изумрудная, 1	5	30		51.8	26.3	38500	1994300	свободна
Изумрудная, 1	1	190		51.4	26.1	38000	1953200	свободна
Изумрудная, 1	2	197		51.6	26.2	38000	1960800	свободна

Рисунок 15 – Контекстное меню

☰

WPF Design

eugene@ya.ru

🔍

🏠

📌

📄

👤

Номер дома

Изумрудная, 1

Цена

От 1500000 До 2000000

Общ площадь

От 0 До 120

Номер этажа

От 1 До 20

Тип квартиры

Двухкомнатные

☒ Свободные квартиры

Показать сравнение

Сбросить

Показать

Адрес	№ этажа	№ квартиры	Кол-во комнат	Общ площадь	Жил площадь	Цена за кв.м	Стоимость	Статус
Изумрудная, 1	3	16	2	51.7	26.4	38500	1990450	свободна
Изумрудная, 1	5	30	2	51.8	26.3	38500	1994300	свободна

Рисунок 16 – Тест сравнения квартир

Теперь проведем тест на бронирование, отмена бронирования и покупку квартиры. Для бронирования квартиры необходимо в контекстном меню выбрать соответствующий пункт «Забронировать». Откроется окно «Заключение договора». Данный владельца, сотрудника и информация о квартире заполняются автоматически, данные покупателя заносятся вручную (показано на рисунке 17), если клиент уже есть в БД его можно выбрать из соответствующего списка (внизу слева).

Рисунок 17 – Заключение договора

После заключения договора на странице «Заключение договора» квартира с номером 16 имеет статус «забронирована». Функции отмены бронирования и покупки происходят также, как и бронирование.

Все заключенные договоры можно просмотреть на странице «Договоры», показано на рисунке 18.

Рисунок 18 – Страница «Договоры»

Откроем базу данных в СУБД MS SQL Server и убедимся, что данные были внесены, показано на рисунке 19.

НомерДогово...	ДатаЗаключе...	КодПокупателя	КодКомплекса	НомерПольз...	СтатусСделки	АдресДома	НомерКварти...	Описание
1	2019-06-19	1	1	1	продана	Изумрудная, 1	1	NULL
2	2019-06-19	5	1	1	забронирована	Изумрудная, 1	16	NULL
3	2019-06-19	5	1	1	отмена брони	Изумрудная, 1	16	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 19 – Таблица «Сделки» в

Таким образом, можно сделать вывод о том, что функции бронирования, отмены бронирования и покупка квартиры работают корректно и заносятся в базу.

5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Работа приложения начинается с открытие окна авторизации, представлено на рисунке 20. Если в компанию застройщика приходит новый сотрудник, его данные для входа можно зарегистрировать через приложение, показано на рисунке 21, нажав на кнопку регистрации. После ввода данных если все введено корректно, новый пользователь будет создан и вся информация сохранится в базе данных. Вид окна с примером заполнения полей представлен на рисунке 22.

АВТОРИЗАЦИЯ

EMAIL
eugene@ya.ru

PASSWORD
••••

ВОЙТИ

ЗАРЕГИСТРИРОВАТЬСЯ

Рисунок 20 – Авторизация пользователя

REGISTRATION

EMAIL

FULL NAME

USERNAME

PHONE

PASSWORD

CONFIRM PASSWORD

REGISTRATION

Рисунок 21 – Регистрация нового пользователя

REGISTRATION

EMAIL
ilina@ya.ru

FULL NAME
Ильина Екатерина Николаевна

USERNAME
iLiNa

PHONE
89016015801

PASSWORD
••••••••

CONFIRM PASSWORD
••••••••

REGISTRATION

Рисунок 22 – Введенные данные при регистрации нового пользователя

					ПГУ 1.090301.27.001 ПЗ	Лист
						31
Изм.	Лист	№ докум.	Подпись	Дата		

Рассмотрим основные возможности, которые представляет приложение. В левой части окна расположено меню с вкладками:

- Подбор квартир. Данная страница необходима для поиска нужных квартир их бронировании и покупки.
- Быстрый просмотр. На этой стране можно просмотреть планировку и информацию каждой квартиры.
- О комплексе. Представлена информация о комплексе.
- Договоры. Страница содержит таблицу со всеми договорами.
- Контакты. Данная страница отображает информацию о контактных лицах фирмы застройщика и о риелторах (сотрудниках).

Основная работа с БД происходит на странице «Подбор квартир». Для поисков подходящих квартир предусмотрена сортировка – верхняя часть страницы, показано на рисунке 23. Чтобы показать квартиры по заданным параметрам необходимо нажать на кнопку «Показать».

The screenshot shows a search interface with the following elements:

- Номер дома** (House Number): Бульвар Прибрежный, 1
- Цена** (Price): От 3700000 До 5000000
- Общ площадь** (Total Area): От 0 До 120
- Номер этажа** (Floor Number): От 1 До 20
- Тип квартиры** (Apartment Type): Трехкомнатные
- ☒ Свободные квартиры
- Buttons:** Показать сравнение, Сбросить, Показать

Рисунок 23 – Пример сортировки квартир по параметрам

Для взаимодействия с квартирами необходимо ее выбрать и вызвать контекстное меню нажатием ПКМ, показано на рисунке 24.

№ квартиры	Кол-во комнат	Общ площадь
166	3	119.5
167	Сравнить	120
172	Отменить	119.9
178	Купить	119.4
179	Забронировать	120
184	Отмена брони	119

Рисунок 24 – Вызванное контекстное меню

После совершения той или иной сделки договор сохраняется БД и всю информацию можно просмотреть на стране «Договоры», показано на рисунке 27.

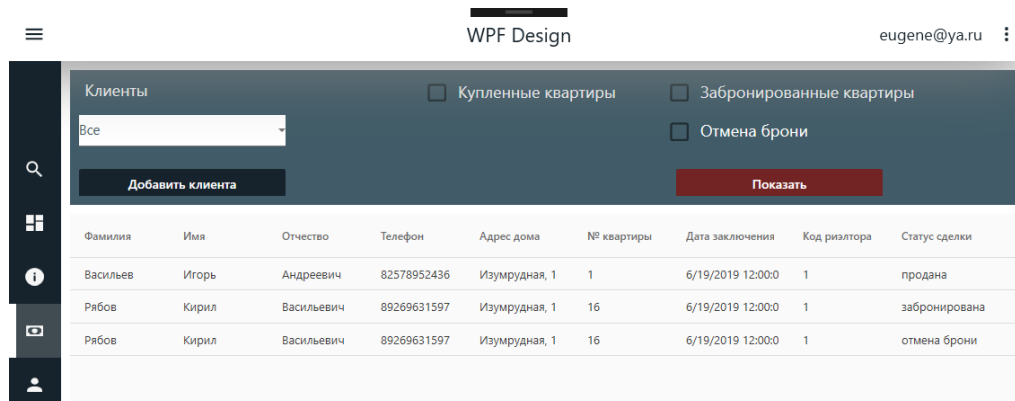


Рисунок 27 – Страница с договорами

На данной странице договоры можно отфильтровать по клиентам или по статусу договора.

Рассмотрим страницу «Быстрый просмотр». На ней присутствуют три кнопки при наведении мыши высвечивается адрес данного дома. После выбора одного из объектов открывается страница с описанием дома, представлено на рисунке 28. На ней содержится общая информацию по дому и о количестве свободных квартир. Также на данной странице в верхней части присутствует вкладка с планировками. При переходе открывается страница с выбором этажа и подъезда, представлено на рисунке 29.

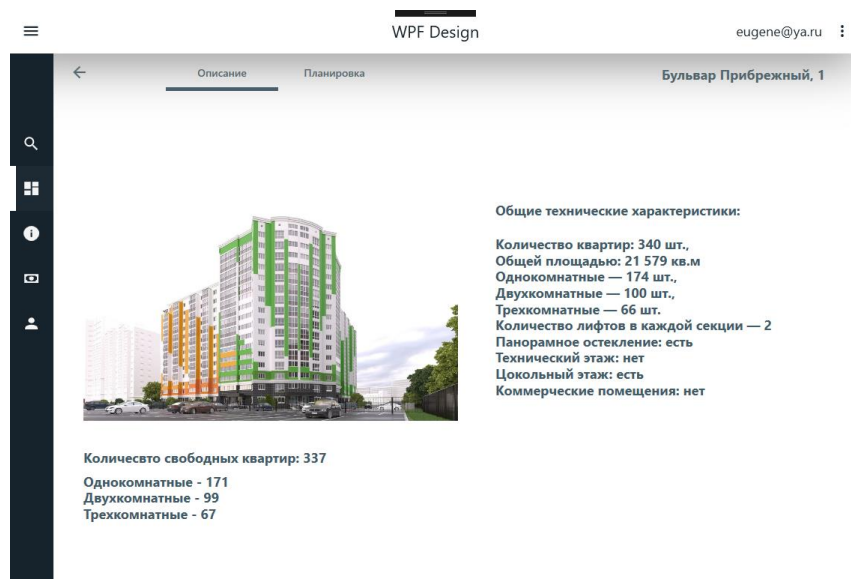


Рисунок 28 – Страница с описание объекта

					ПГУ 1.090301.27.001 ПЗ	Лист
						34
Изм.	Лист	№ докум.	Подпись	Дата		

При переходе на тот или иной подъезд доступны те или иные этажи. При переходе на какой-либо этаж открывается план этого этажа. Каждая квартира имеет индикатор статуса:

- Зеленый – квартира свободна;
- Желтый – квартира забронирована;
- Красный – квартира куплена.



Рисунок 29 – Страница с планировкой этажа

При наведении мыши на одну из квартир высвечивается информация о ней, показано на рисунке 30.



Рисунок 30 – Информация о выбранной квартире

После окончания работы с программы следует выйти из профиля сотрудника нажав на иконку сверху слева и далее в окне авторизации нажать на крестик.

					ПГУ 1.090301.27.001 ПЗ	Лист
						35
Изм.	Лист	№ докум.	Подпись	Дата		

ЗАКЛЮЧЕНИЕ

Результатом работы является клиентское приложение с интуитивно понятным пользовательским интерфейсом. Приложение позволяет работать с базой данных застройщика по комплексу, выбирать любой из домов, просматривать планировку этажей и квартир, сравнивать их, также можно осуществлять бронирование и покупку квартир, и занесение данных клиента для заключения договоров. Данный проект полностью удовлетворяет поставленной цели.

Данная информационная система позволяет пользователю беспрепятственно вести учет квартир застройщика, быстро отслеживать свободные, купленные и забронированные квартиры. Данные вводятся самим пользователем и визуально отображаются в виде обычных числовых данных в таблицах и на планах этажей.

Отладка и тестирование состоит из поиска квартир по нужным критериям и сравнения их, добавления клиентов в базу данных, бронировании и покупки квартир.

Работа над проектом позволила расширить знания и улучшить навыки создания клиентских приложения и работы с базами данных.

Программа может быть подвержена апгрейду в дальнейшем, в соответствии с потребностями пользователей, например, можно добавить больше комплексов от застройщика. Кроме того, можно добавить возможность работы приложения несколькими пользователями одновременно с разных рабочих мест в офисе застройщика. Что может быть достигнуто путем создания серверного системного блока и наличия локальной сети в офисе. Кроме того, информационная система может в будущем использоваться удаленно. Это может быть достигнуто путем подключения web-ресурса или создание облачной базы данных системы.

					ПГУ 1.090301.27.001 ПЗ	Лист
						36
Изм.	Лист	№ докум.	Подпись	Дата		

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. И.А. Казакова. Основы языка Transact SQL. Учебное пособие. - Пенза: Изд-во Пенз. гос. ун-та., 2010.
2. Хендерсон, Кен. Профессиональное руководство по SQL Server: структура и реализация. Пер. с англ. – М.: Издательский дом “Вильямс”, 2006.
3. О.В. Прокофьев, И.А. Казакова, С.В. Самуйлов, С.В. Шибанов. Работа с базами данных в Microsoft SQL Server 2008. – Пенза: Изд-во Пенз. гос. ун-та., 2012.
4. Пушников А. Ю. Введение в системы управления базами данных. [Электронный ресурс] - <http://www.citforum.ru/database/dblearn/index.shtml>
5. Дейт К. Дж. Введение в системы баз данных – М.: Издательский дом “Вильямс”, 2001.
6. Руководство по MS SQL Server 2017. [Электронный ресурс] - <https://metanit.com/sql/sqlserver/>
7. Руководство по WPF [Электронный ресурс] – <https://metanit.com/sharp/wpf/>
8. WPF - Windows Presentation Foundation [Электронный ресурс] – https://professorweb.ru/my/WPF/base_WPF/level1/info_WPF.php
9. Герберт Шилдг. Полный справочник по C#.:Пер. с англ. — М. : Издательский дом "Вильямс", 2004.
10. Шилдт Герберт- C# Учебный Курс. 2003.
11. Полное руководство по языку программирования C# 7.0 и платформе .NET 4.7 [Электронный ресурс] – <https://metanit.com/sharp/tutorial/>

					ПГУ 1.090301.27.001 ПЗ	Лист
						37
Изм.	Лист	№ докум.	Подпись	Дата		

ПРИЛОЖЕНИЕ А
ЛИСТИНГ ПРОГРАММЫ

					ПГУ 1.090301.27.001 ПЗ	<i>Лист</i>
						38
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

Приложение А.1 Файл Login.xaml.cs

```

using System.Windows;
using System.Data;

namespace Test02031950
{
    public partial class Login : Window
    {
        public MainWindow mainWindow; // ссылка на форму MainWindow

        public Login(MainWindow _mainWindow) // принимаем ссылку форму
        {
            InitializeComponent();
            loginBox.Text = Data.Value;
            mainWindow = _mainWindow; // запоминаем ссылку на форму
            loginBox.Text = "eugene@ya.ru";
            passwordBox.Password = "9798";
        }

        private void ButtonLogin_Click(object sender, RoutedEventArgs e)
        {
            Data.Value = loginBox.Text;

            if (loginBox.Text.Length > 0) // проверяем введен ли логин
            {
                if (passwordBox.Password.Length > 0) // проверяем введен ли пароль
                {
                    // ищем в базе данных пользователя с такими данными
                    DataTable dt_user = mainWindow.Select("SELECT * FROM
[dbo].[Пользователи] WHERE [ИмейлПользователя] = '" + loginBox.Text + "' AND
[ХэшПароляПользователя] = '" + passwordBox.Password + "'");
                    if (dt_user.Rows.Count > 0) // если такая запись существует
                    {
                        this.Visibility = Visibility.Collapsed;
                    } else
                    {
                        LabelDecline.Text = "Пользователя не найден";
                    }
                } else LabelDecline.Text = "Введите пароль"; // выводим ошибку
            } else LabelDecline.Text = "Введите логин"; // выводим ошибку
        }

        public string LoginBox
        {
            get { return loginBox.Text; }
        }

        public string PasswordBox
        {
            get { return passwordBox.Password; }
        }

        private void ButtonExitLogin_Click(object sender, RoutedEventArgs e)
        {
            Application.Current.Shutdown();
        }

        private void ButtonRegistration_Click(object sender, RoutedEventArgs e)
        {
            frame.Navigate(new Registration(mainWindow));
        }
    }
}

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						39
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А.2 Файл MainWindow.xaml.cs

```

using System.Windows;
using System.Windows.Controls;
using System.Data.SqlClient;
using System.Data;
using System.Configuration;

namespace Test02031950
{
    public partial class MainWindow : Window
    {
        Login loginWindow;
        // получаем строку подключения из app.config
        static string connectionString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
        string ConnectionString;

        public MainWindow()
        {
            loginWindow = new Login(this);
            InitializeComponent();
            loginWindow.ShowDialog();
            LoginName.Text = Data.Value; // запоминание логина
            ConnectionString = Data.ValueConnectionString;
        }
        // список доступных окон
        public enum pages
        {
            UserControlTab4
        }

        void OpenPage(pages pages)
        {
            if (pages == pages.UserControlTab4)
            {
                GridPrincipal.Children.Add(new UserControlTab4(this));
            }
        }

        public DataTable Select(string selectSQL) // функция подключения к базе
данных и обработка запросов
        {
            DataTable dataTable = new DataTable("dataBase"); // создаём таблицу в
приложении
            // подключаемся к базе данных
            SqlConnection sqlConnection = new SqlConnection(connectionString);
            sqlConnection.Open(); // открываем базу
данных
            SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём
команду
            sqlCommand.CommandText = selectSQL; // присваиваем команде
текст
            SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём
обработчик
            sqlDataAdapter.Fill(dataTable); // возвращаем таблицу с результатом
            return dataTable;
        }

        private void ButtonPopUpLogout_Click(object sender, RoutedEventArgs e) //
выход
        {
            Authorize();
            //OpenPage(pages.UserControlTab4);
        }
    }
}

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

```

// открытие меню
private void ButtonOpenMenu_Click(object sender, RoutedEventArgs e)
{
    ButtonOpenMenu.Visibility = Visibility.Collapsed;
    ButtonCloseMenu.Visibility = Visibility.Visible;
}

// закрытие меню
private void ButtonCloseMenu_Click(object sender, RoutedEventArgs e) //
{
    ButtonOpenMenu.Visibility = Visibility.Visible;
    ButtonCloseMenu.Visibility = Visibility.Collapsed;
}

private void Authorize() // повторный вход
{
    loginWindow = new Login(this);
    loginWindow.ShowDialog();
}

private void ListViewMenu_SelectionChanged(object sender,
SelectionChangedEventArgs e)
{
    int index = ListViewMenu.SelectedIndex;
    MoveCursorMenu(index);
    switch (index)
    {
        case 0:
            GridPrincipial.Children.Clear();
            GridPrincipial.Children.Add(new UserControlTab0_Magnifier());
            break;
        case 1:
            GridPrincipial.Children.Clear();
            GridPrincipial.Children.Add(new UserControlTab1());
            break;
        case 2:
            GridPrincipial.Children.Clear();
            GridPrincipial.Children.Add(new UserControlTab2());
            break;
        case 3:
            GridPrincipial.Children.Clear();
            GridPrincipial.Children.Add(new UserControlTab3());
            break;
        case 4:
            GridPrincipial.Children.Clear();
            GridPrincipial.Children.Add(new UserControlTab4(this));
            break;
        default:
            break;
    }
}

private void MoveCursorMenu(int index)
{
    TrainsitionigContentSlide.OnApplyTemplate();
    GridCursor.Margin = new Thickness(0, (150 + (60 * index)), 0, 0);
}

static class Data
{
    public static string Value { get; set; } // логин пользователя
    public static string ValueConnectionString { get; set; } // строка
    public static string ValueApartmentNumber { get; set; } // id квартиры
}

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						41
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А.3 Файл Magnifier.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using System.Data.SqlClient;
using System.Data;
using System.Configuration;

namespace Test02031950
{
    public partial class UserControlTab0_Magnifier : UserControl
    {
        string connectionString; // строка присоединения к БД
        string sql; // строка запроса
        List<string> numbers = new List<string>() {}; // лист сравнения
        string APPART = "0";
        DataGrid dg; // отслеживание нажатия на табл.
        ContractConclusion ContractConclusionWindow; // окно договора

        public UserControlTab0_Magnifier()
        {
            InitializeComponent();
            connectionString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
            BindComboBox(houseNumber);
            homeGrid.Columns[0].Visibility = Visibility.Hidden; // сккрытие столбца
id_квартиры
        }

        public void BindComboBox(ComboBox comboBoxName) // заполнение ComboBox
адресами
        {
            sql = "SELECT [Адрес] FROM [Дома]";
            SqlConnection connection = null;
            try
            {
                connection = new SqlConnection(connectionString);
                SqlCommand command = new SqlCommand(sql, connection);
                connection.Open();
                SqlDataReader SDR = command.ExecuteReader();
                while (SDR.Read())
                {
                    houseNumber.Items.Add(SDR.GetString(0));
                }
            } catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            } finally
            {
                if (connection != null)
                    connection.Close();
            }
        }

        public void HOMEgrid() // обработчик таблицы
        {
            int index = typeApartment.SelectedIndex - 2;
            if (houseNumber.Text == "Все")
            {
                if (index < 0)
                {

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						42
Изм.	Лист	№ докум.	Подпись	Дата		

```

        sql = "SELECT
[Квартиры].[IdКвартиры],[Адрес],[НомерЭтажа],[НомерКвартиры],[КолвоКомнат],[ОбщаяПлощадь],[
ЖилаяПлощадь],[ЦенаЗаКВм],[Стоимость],[Статус] FROM [Дома],[Квартиры]" +
        "WHERE [Дома].[НомерДома] = [Квартиры].[НомерДома] " +
        "AND [НомерЭтажа] >= '" + floorNumber1.Text.ToString() + "'
AND [НомерЭтажа] <= '" + floorNumber2.Text.ToString() + "'" +
        "AND [ОбщаяПлощадь] >= '" + Area1.Text.ToString() + "' AND
[ОбщаяПлощадь] <= '" + Area2.Text.ToString() + "'" +
        "AND [Стоимость] >= '" + Price1.Text.ToString() + "' AND
[Стоимость] <= '" + Price2.Text.ToString() + "'";
    } else
    {
        sql = "SELECT
[Квартиры].[IdКвартиры],[Адрес],[НомерЭтажа],[НомерКвартиры],[КолвоКомнат],[ОбщаяПлощадь],[
ЖилаяПлощадь],[ЦенаЗаКВм],[Стоимость],[Статус] FROM [Дома],[Квартиры]" +
        "WHERE [Дома].[НомерДома] = [Квартиры].[НомерДома] AND
[КолвоКомнат]=''" + index + "'" +
        "AND [НомерЭтажа] >= '" + floorNumber1.Text.ToString() + "'
AND [НомерЭтажа] <= '" + floorNumber2.Text.ToString() + "'" +
        "AND [ОбщаяПлощадь] >= '" + Area1.Text.ToString() + "' AND
[ОбщаяПлощадь] <= '" + Area2.Text.ToString() + "'" +
        "AND [Стоимость] >= '" + Price1.Text.ToString() + "' AND
[Стоимость] <= '" + Price2.Text.ToString() + "'";
    }
    } else
    {
        if (index < 0)
        {
            sql = "SELECT
[Квартиры].[IdКвартиры],[Адрес],[НомерЭтажа],[НомерКвартиры],[КолвоКомнат],[ОбщаяПлощадь],[
ЖилаяПлощадь],[ЦенаЗаКВм],[Стоимость],[Статус] FROM [Дома],[Квартиры]" +
            "WHERE [Дома].[НомерДома] = [Квартиры].[НомерДома] AND
[Адрес] = '" + houseNumber.Text + "'" +
            "AND [НомерЭтажа] >= '" + floorNumber1.Text.ToString() + "'
AND [НомерЭтажа] <= '" + floorNumber2.Text.ToString() + "'" +
            "AND [ОбщаяПлощадь] >= '" + Area1.Text.ToString() + "' AND
[ОбщаяПлощадь] <= '" + Area2.Text.ToString() + "'" +
            "AND [Стоимость] >= '" + Price1.Text.ToString() + "' AND
[Стоимость] <= '" + Price2.Text.ToString() + "'";
        } else
        {
            sql = "SELECT
[Квартиры].[IdКвартиры],[Адрес],[НомерЭтажа],[НомерКвартиры],[КолвоКомнат],[ОбщаяПлощадь],[
ЖилаяПлощадь],[ЦенаЗаКВм],[Стоимость],[Статус] FROM [Дома],[Квартиры]" +
            "WHERE [Дома].[НомерДома] = [Квартиры].[НомерДома] AND
[Адрес] = '" + houseNumber.Text + "' AND [КолвоКомнат] = '" + index + "'" +
            "AND [НомерЭтажа] >= '" + floorNumber1.Text.ToString() + "'
AND [НомерЭтажа] <= '" + floorNumber2.Text.ToString() + "'" +
            "AND [ОбщаяПлощадь] >= '" + Area1.Text.ToString() + "' AND
[ОбщаяПлощадь] <= '" + Area2.Text.ToString() + "'" +
            "AND [Стоимость] >= '" + Price1.Text.ToString() + "' AND
[Стоимость] <= '" + Price2.Text.ToString() + "'";
        }
    }
    if (CheckBoxApartment.IsChecked == true)
    {
        sql += " AND [Статус] = 'свободна'";
    }
    SqlConnection connection = null;
    try
    {
        connection = new SqlConnection(connectionString);
        SqlCommand command = new SqlCommand(sql, connection);
        connection.Open();

        DataTable dt = new DataTable();

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

```

        SqlDataAdapter da = new SqlDataAdapter(command);
        da.Fill(dt);
        homeGrid.ItemsSource = dt.AsDataView();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        if (connection != null)
            connection.Close();
    }
}

показать
private void AddButton_Click(object sender, RoutedEventArgs e) // кнопка
{
    HOMEgrid();
}

private void TypeApartment_SelectionChanged(object sender,
SelectionChangedEventArgs e)
{
    ComboBox comboBox = (ComboBox)sender;
    ComboBoxItem selectedItem = (ComboBoxItem)comboBox.SelectedItem;
}

сравнение квартир, "Показать сравнение"
private void ButtonComparison_Click(object sender, RoutedEventArgs e) //
{
    APPART = "0";
    for (int i = 0; i < numbers.Count; i++)
    {
        APPART += "," + numbers[i].ToString();
    }
    sql = "SELECT
[Квартиры].[IdКвартиры],[Адрес],[НомерЭтажа],[НомерКвартиры],[КолвоКомнат],[ОбщаяПлощадь],[
ЖилаяПлощадь],[ЦенаЗаКВм],[Стоимость],[Статус] FROM [Дома],[Квартиры]" +
        "WHERE [Дома].[НомерДома] = [Квартиры].[НомерДома] AND [IdКвартиры]
IN (" + APPART.ToString() + ")";
    SqlConnection connection = null;
    try
    {
        connection = new SqlConnection(connectionString);
        connection.Open();
        SqlCommand command = new SqlCommand(sql, connection);

        DataTable dt = new DataTable();
        SqlDataAdapter da = new SqlDataAdapter(command);
        da.Fill(dt);
        homeGrid.ItemsSource = dt.AsDataView();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        if (connection != null)
            connection.Close();
    }
}

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						44
Изм.	Лист	№ докум.	Подпись	Дата		

```

private void ButtonReset_Click(object sender, RoutedEventArgs e) // сброс фильтра
{
    houseNumber.SelectedIndex = 0;
    typeApartment.SelectedIndex = 0;
    Price1.Text = "0";
    Price2.Text = "5000000";
    Area1.Text = "0";
    Area2.Text = "120";
    floorNumber1.Text = "1";
    floorNumber2.Text = "20";
}

private void HomeGrid_SelectionChanged(object sender,
SelectionChangedEventArgs e) // выбор строки
{
    int ind = homeGrid.SelectedIndex; // индекс строки
    dg = sender as DataGrid;
    DataRowView dr = dg.SelectedItem as DataRowView;
    if (dr != null)
    {
        Data.ValueApartmentNumber = dr["IdКвартиры"].ToString();
        if (dr["Статус"].ToString() == "продана")
        {
            ButtonBuy.IsEnabled = false;
            ButtonBooking.IsEnabled = false;
            ButtonRemovalBooking.IsEnabled = false;
        } else if (dr["Статус"].ToString() == "забронирована")
        {
            ButtonBuy.IsEnabled = true;
            ButtonBooking.IsEnabled = false;
            ButtonRemovalBooking.IsEnabled = true;
        } else
        {
            ButtonBuy.IsEnabled = true;
            ButtonBooking.IsEnabled = true;
            ButtonRemovalBooking.IsEnabled = false;
        }
    }
}

private void HomeGrid_LoadingRow(object sender, DataGridRowEventArgs e)
{ try
    { for (int i = 0; i < numbers.Count; i++)
        { if
            (Convert.ToString(((System.Data.DataRowView)(e.Row.DataContext)).Row.ItemArray[0].ToString(
            )) == numbers[i])
                {
                    e.Row.Background = (SolidColorBrush)(new BrushConverter().ConvertFrom("#7FFF0000"));
                }
        }
    } catch
    { }
}

private void MenuItem_Click(object sender, RoutedEventArgs e) // ПКМ,
добавление в "СРАВНЕНИЕ"
{
    DataRowView dr = dg.SelectedItem as DataRowView;
    if (dr != null)
    {
        APPART = dr["IdКвартиры"].ToString();
        numbers.Add(APPART);
        int ind = homeGrid.SelectedIndex;
        DataGridRow row =
(DataGridRow)homeGrid.ItemContainerGenerator.ContainerFromIndex(ind);

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						45
Изм.	Лист	№ докум.	Подпись	Дата		

```

        row.Background = (SolidColorBrush)(new
BrushConverter().ConvertFrom("#7FFF0000"));
    }
}

private void MenuItem_Click_1(object sender, RoutedEventArgs e) // ПКМ,
"ОТМЕНИТЬ" сравнение
{
    DataRowView dr = dg.SelectedItem as DataRowView;
    if (dr != null)
    {
        APPART = dr["IdКвартиры"].ToString();
        for (int i = 0; i < numbers.Count; i++)
        {
            if (numbers[i] == APPART)
            {
                numbers.RemoveAt(i);
                i--;
                int ind = homeGrid.SelectedIndex;
                DataGridRow row =
(DataGridRow)homeGrid.ItemContainerGenerator.ContainerFromIndex(ind);
                row.Background = Brushes.WhiteSmoke;
            }
        }
    }
}

private void MenuItem_Click_2(object sender, RoutedEventArgs e) // ПКМ,
"КУПИТЬ"
{
    ContractConclusionWindow = new ContractConclusion();
    ContractConclusionWindow.ContractStatus.Items.Add("ПОКУПКА");
    ContractConclusionWindow.ShowDialog();
}

private void MenuItem_Click_3(object sender, RoutedEventArgs e) // ПКМ,
"ЗАБРОНИРОВАТЬ"
{
    ContractConclusionWindow = new ContractConclusion();
    ContractConclusionWindow.ContractStatus.Items.Add("БРОНИРОВАНИЕ");
    ContractConclusionWindow.ShowDialog();
}

private void MenuItem_Click_4(object sender, RoutedEventArgs e) // ПКМ,
"ОТМЕНА БРОНИ"
{
    ContractConclusionWindow = new ContractConclusion();
    ContractConclusionWindow.ContractStatus.Items.Add("ОТМЕНА БРОНИ");
    ContractConclusionWindow.ShowDialog();
}
}
}

```

Приложение А.4 Файл ContractConclusion.xaml.cs

```

using System;
using System.Windows;
using System.Windows.Controls;
using System.Data.SqlClient;
using System.Configuration;

namespace Test02031950
{
    public partial class ContractConclusion : Window
    {
        string connectionString; // строка присоединения к БД
        string sql, sql_1, sql_2, sql_3;
        string curDate = DateTime.Now.ToShortDateString(); // получение текущей даты
        string idKomplex, idWorker, idClient, statusTrade, addressHome,
numberApartment, idApartment;

        public ContractConclusion()
        {
            InitializeComponent();
            connectionString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
            ClientBase();
            FunctionFillContractConclusion();
            ContractStatus.SelectedIndex = 0;
        }

        private void ButtonExitLogin_Click(object sender, RoutedEventArgs e) //
закрытие окна
        {
            this.Visibility = Visibility.Collapsed;
        }

        private void ButtonContractConclusion_Click(object sender, RoutedEventArgs
e)
        {
            FunctionAddClient();
            LabelDecline.Text = "";
            FunctionContractConclusion();
        }

        private void ButtonAddClient_Click(object sender, RoutedEventArgs e)
        {
            FunctionAddClient();
        }

        public void FunctionFillContractConclusion() // отображение информации
        {
            SqlConnection connection = null;
            connection = new SqlConnection(connectionString);
            sql = "SELECT
[Дома].[Адрес],[Квартиры].[IdКвартиры],[НомерКвартиры],[НомерЭтажа],[КолвоКомнат],[ОбщаяПло
щадь],[ЖилаяПлощадь],[Стоимость],[Комплекс].[КодКомплекса]," +

"[НазваниеСтроительнойОрганизации],[КонтактноеЛицо],[ТелефонВладельца],[Пользователи].[НомерПользователя],[ФамилияПользователя],[ТелефонПользователя]" +
            "FROM[Дома],[Квартиры],[Владельцы],[Комплекс],[Пользователи]" +
            "WHERE[Дома].[НомерДома] = [Квартиры].[НомерДома]" +
            "AND[Дома].[КодКомплекса] = [Комплекс].[КодКомплекса]" +
            "AND[Комплекс].[КодВладельца] = [Владельцы].[КодВладельца]" +
            "AND[IdКвартиры] = '" + Data.ValueApartmentNumber + "'" +
            "AND[ИмейлПользователя] = '" + Data.Value + "'";
        }
    }
}

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						47
Изм.	Лист	№ докум.	Подпись	Дата		

```

try
{
    SqlCommand command = new SqlCommand(sql, connection);
    connection.Open();

    SqlDataReader SDR = command.ExecuteReader();
    while (SDR.Read())
    {
        idKomplex = SDR[8].ToString();
        idWorker = SDR[12].ToString();
        addressHome = SDR[0].ToString();
        numberApartment = SDR[2].ToString();
        idApartment = SDR[1].ToString();

        NameCompany.Text = SDR[9].ToString();
        ContactPerson.Text = SDR[10].ToString();
        PhoneBoxContactPerson.Text = SDR[11].ToString();

        ContactWorker.Text = SDR[13].ToString();
        PhoneBoxWorker.Text = SDR[14].ToString();

        AddressHouse.Text = SDR[0].ToString();
        ApartmentNumber.Text = SDR[2].ToString();
        RoomNumber.Text = SDR[3].ToString();
        ApartmentQuantity.Text = SDR[4].ToString();
        AreaTotal.Text = SDR[5].ToString();
        AreaLiving.Text = SDR[6].ToString();
        AreaPrice.Text = SDR[7].ToString();
    }
    SDR.Close();
} catch (Exception ex)
{
    MessageBox.Show(ex.Message);
} finally
{
    if (connection != null)
        connection.Close();
}
}

public void FunctionAddClient() // добавление нового клиента в БД
{
    SqlConnection connection = null;
    connection = new SqlConnection(connectionString);
    string PassportSDR = "";
    if (FamBoxClient.Text != "" || NameBoxClient.Text != "" ||
    PatrBoxClient.Text != "" || PassportSeries.Text != "" || PassportNumber.Text != "" ||
    AddressClient.Text != "" || PhoneBoxClient.Text != "")
    {
        sql_1 = "INSERT INTO [dbo].[Покупатели] (ФамилияПокупателя,
ИмяПокупателя, ОтчествоПокупателя, СерияПаспорта, НомерПаспорта, АдресПокупателя,
ТелефонПокупателя)" +
        "VALUES (@ФамилияПокупателя, @ИмяПокупателя,
@ОтчествоПокупателя, @СерияПаспорта, @НомерПаспорта, @АдресПокупателя,
@ТелефонПокупателя)";

        SqlCommand command_1 = new SqlCommand(sql_1, connection);
        command_1.Parameters.AddWithValue("@ФамилияПокупателя", FamBoxClient.Text);
        command_1.Parameters.AddWithValue("@ИмяПокупателя", NameBoxClient.Text);
        command_1.Parameters.AddWithValue("@ОтчествоПокупателя", PatrBoxClient.Text);
        command_1.Parameters.AddWithValue("@СерияПаспорта", PassportSeries.Text);
        command_1.Parameters.AddWithValue("@НомерПаспорта", PassportNumber.Text);
        command_1.Parameters.AddWithValue("@АдресПокупателя", AddressClient.Text);
        command_1.Parameters.AddWithValue("@ТелефонПокупателя", PhoneBoxClient.Text);

        sql_2 = "SELECT [СерияПаспорта],[НомерПаспорта]" +

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

```

        "FROM [Покупатели] WHERE [СерияПаспорта]+[НомерПаспорта]='" +
PassportSeries.Text.ToString() + PassportNumber.Text.ToString() + "'";
        SqlCommand command_2 = new SqlCommand(sql_2, connection);

        try
        {
            connection.Open();
            SqlDataReader SDR = command_2.ExecuteReader();
            while (SDR.Read())
            {
                PassportSDR = SDR[0].ToString() + SDR[1].ToString();
            }
            SDR.Close();
            if (PassportSeries.Text.ToString() +
PassportNumber.Text.ToString() != PassportSDR)
            {
                int n = command_1.ExecuteNonQuery();
                LabelDecline.Text = "Пользователь добавлен!";
            } else
            {
                LabelDecline.Text = "Пользователь уже существует!";
            }
        } catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        } finally
        { if (connection != null)
            connection.Close();
        }
    } else
    {
        LabelDecline.Text = "Проверьте данные о клиенте!";
    }
}

public void ClientBase() // заполнение клиентской базой
{
    SqlConnection connection = null;
    connection = new SqlConnection(connectionString);
    sql = "SELECT * FROM [Покупатели]";
    SqlCommand command = new SqlCommand(sql, connection);

    try
    {
        connection.Open();
        SqlDataReader SDR = command.ExecuteReader();
        while (SDR.Read())
        {
            ComboBoxClientBase.Items.Add(SDR[1].ToString() + " " +
SDR[2].ToString() + " " + SDR[3].ToString());
        }
        SDR.Close();
    } catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    } finally
    { if (connection != null)
        connection.Close();
    }
}

private void ComboBoxClientBase_SelectionChanged(object sender,
SelectionChangedEventArgs e) // автоматическое заполнение клиента
{
    SqlConnection connection = null;
    connection = new SqlConnection(connectionString);

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						49
Изм.	Лист	№ докум.	Подпись	Дата		

```

int comboBoxIndex = 0;
ComboBox comboBox = (ComboBox)sender;
comboBoxIndex = comboBox.SelectedIndex;
if (comboBox.SelectedItem != null || comboBoxIndex != 0)
{
    sql = "SELECT * FROM [Покупатели] " +
        "WHERE [ФамилияПокупателя]+' '[ИмяПокупателя]+' '[ОтчествоПокупателя]='' + comboBox.SelectedItem.ToString() + '";

    SqlCommand command = new SqlCommand(sql, connection);

    try
    {
        connection.Open();
        SqlDataReader SDR = command.ExecuteReader();
        while (SDR.Read())
        {
            FamBoxClient.Text = SDR[1].ToString();
            NameBoxClient.Text = SDR[2].ToString();
            PatrBoxClient.Text = SDR[3].ToString();
            PassportSeries.Text = SDR[4].ToString();
            PassportNumber.Text = SDR[5].ToString();
            AddressClient.Text = SDR[6].ToString();
            PhoneBoxClient.Text = SDR[7].ToString();
        }
        SDR.Close();
    } catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    } finally
    {
        if (connection != null)
            connection.Close();
    }
} else if (comboBoxIndex == 0)
{
    FamBoxClient.Text = "";
    NameBoxClient.Text = "";
    PatrBoxClient.Text = "";
    PassportSeries.Text = "";
    PassportNumber.Text = "";
    AddressClient.Text = "";
    PhoneBoxClient.Text = "";
}

}

public void FunctionContractConclusion() // заключение договора
{
    string idClient1 = "0";
    SqlConnection connection = null;
    connection = new SqlConnection(connectionString);
    if (ContractStatus.SelectedItem.ToString() == "ПОКУПКА")
        statusTrade = "продана";
    else if (ContractStatus.SelectedItem.ToString() == "БРОНИРОВАНИЕ")
        statusTrade = "забронирована";
    else if (ContractStatus.SelectedItem.ToString() == "ОТМЕНА БРОНИ")
        statusTrade = "отмена брони";
    sql_1 = "SELECT * FROM [Покупатели] " +
        "WHERE
[ФамилияПокупателя]+[ИмяПокупателя]+[ОтчествоПокупателя]='' + FamBoxClient.Text.ToString()
+ NameBoxClient.Text.ToString() + PatrBoxClient.Text.ToString() + '";

    sql_2 = "INSERT INTO [dbo].[Сделки] (ДатаЗаклучения, КодПокупателя,
КодКомплекса, НомерПользователя, СтатусСделки, АдресДома, НомерКвартиры)" +
        "VALUES (@ДатаЗаклучения, @КодПокупателя, @КодКомплекса,
@НомерПользователя, @СтатусСделки, @АдресДома, @НомерКвартиры)";

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

```

try
{
    connection.Open();
    SqlCommand command_1 = new SqlCommand(sql_1, connection);
    SqlDataReader SDR_1 = command_1.ExecuteReader();
    while (SDR_1.Read())
    {
        idClient = SDR_1[0].ToString();
    }
    SDR_1.Close();
    SqlCommand command_2 = new SqlCommand(sql_2, connection);
    command_2.Parameters.AddWithValue("@ДатаЗаклучения", curDate);
    command_2.Parameters.AddWithValue("@КодПокупателя", idClient);
    command_2.Parameters.AddWithValue("@КодКомплекса", idKomplex);
    command_2.Parameters.AddWithValue("@НомерПользователя", idWorker);
    command_2.Parameters.AddWithValue("@СтатусСделки", statusTrade);
    command_2.Parameters.AddWithValue("@АдресДома", addressHome);
    command_2.Parameters.AddWithValue("@НомерКвартиры",
numberApartment);

    if (statusTrade == "отмена брони")
        sql_3 = "UPDATE [Квартиры] SET [Статус]='свободна' WHERE
[IdКвартиры]='" + idApartment + "'";
    else
        sql_3 = "UPDATE [Квартиры] SET [Статус]='" + statusTrade + "'
WHERE [IdКвартиры]='" + idApartment + "'";
    SqlCommand command_3 = new SqlCommand(sql_3, connection);

    if (statusTrade == "отмена брони")
    {
        sql = "SELECT * FROM [Сделки] WHERE [АдресДома]='" + addressHome
+ "' AND [НомерКвартиры]='" + numberApartment + "' AND [СтатусСделки]='забронирована'";
        SqlCommand command = new SqlCommand(sql, connection);
        SqlDataReader SDR = command.ExecuteReader();
        while (SDR.Read())
        {
            idClient1 = SDR[2].ToString();
        }
        SDR.Close();
        if (idClient == idClient1)
        {
            int n = command_2.ExecuteNonQuery();
            int m = command_3.ExecuteNonQuery();
            LabelDecline.Text = "Данные внесены!";
        }
        else
        {
            LabelDecline.Text = "Клиент не совпадает!";
        }
    } else
    {
        int n = command_2.ExecuteNonQuery();
        int m = command_3.ExecuteNonQuery();
        LabelDecline.Text = "Данные внесены!";
    }
} catch (Exception ex)
{
    MessageBox.Show(ex.Message);
} finally
{
    if (connection != null)
        connection.Close();
}
}
}

```

					ПГУ 1.090301.27.001 ПЗ	Лист
						51
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А.5 Файл App.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6" />
  </startup>
  <connectionStrings>
    <add name="DefaultConnection"
      connectionString="Data Source=.\SQLEXPRESS;Initial
Catalog=DeveloperApartments;Integrated Security=True"
      providerName="System.Data.SqlClient"/>
  </connectionStrings>
</configuration>
```

					ПГУ 1.090301.27.001 ПЗ	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

ПРИЛОЖЕНИЕ Б
РИСУНКИ И ДИАГРАММЫ

					<i>ПГУ 1.090301.27.001 ПЗ</i>	<i>Лист</i>
						53
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

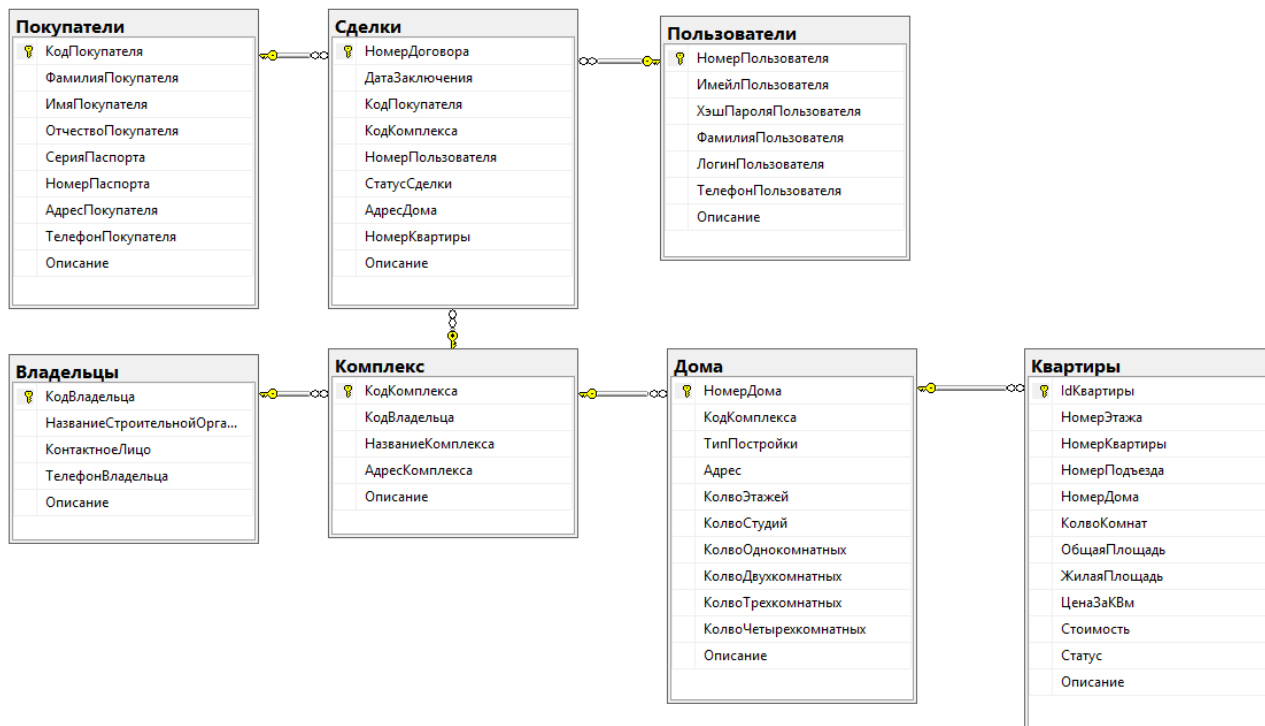


Рисунок 31 – Диаграмма базы данных

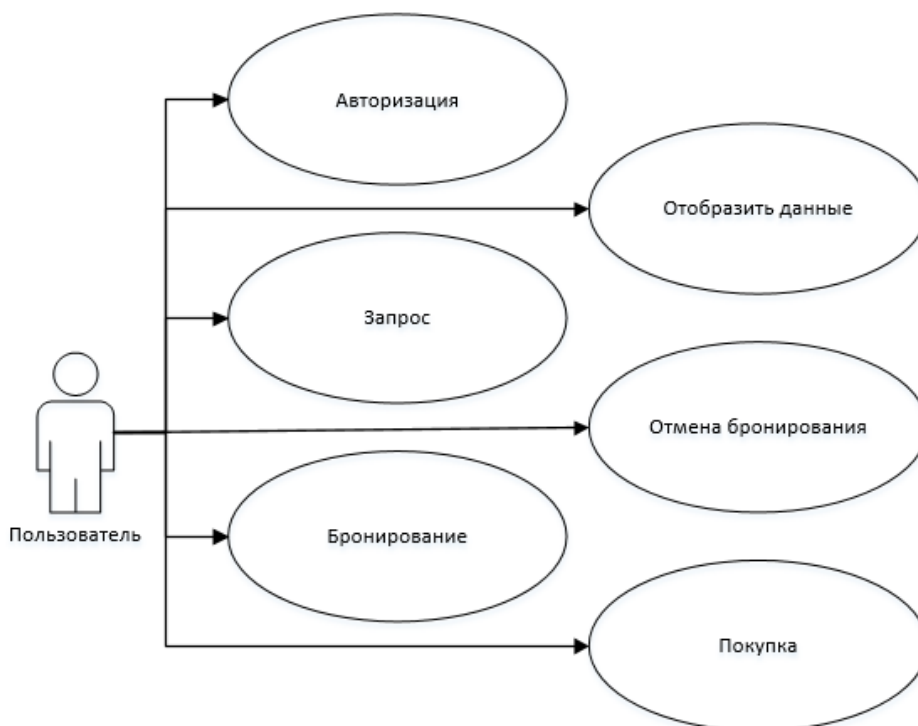


Рисунок 32 – Диаграмма вариантов использования

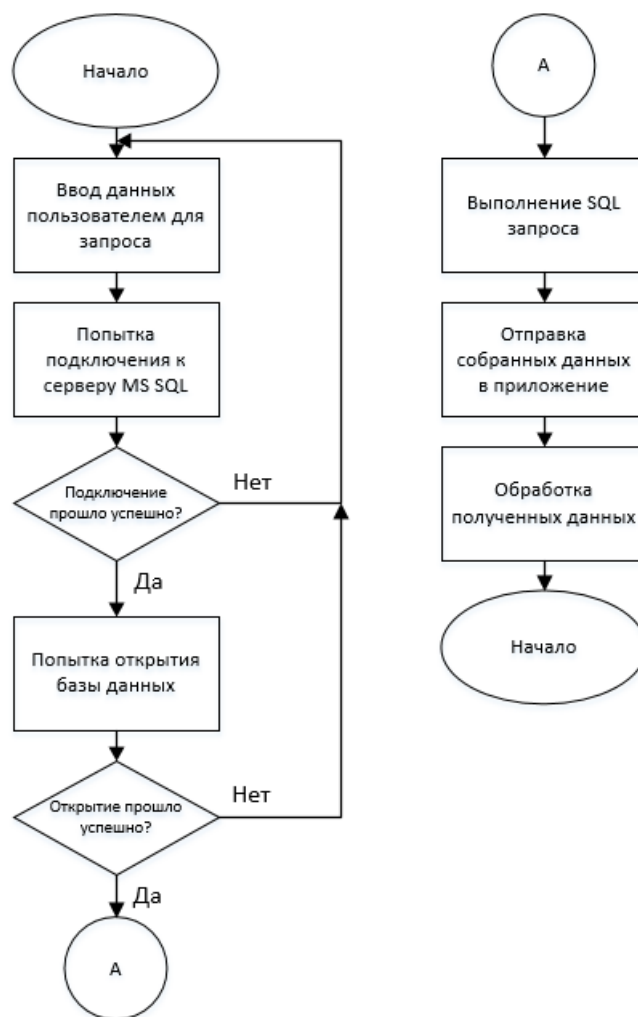


Рисунок 33 – Алгоритм обработки запроса

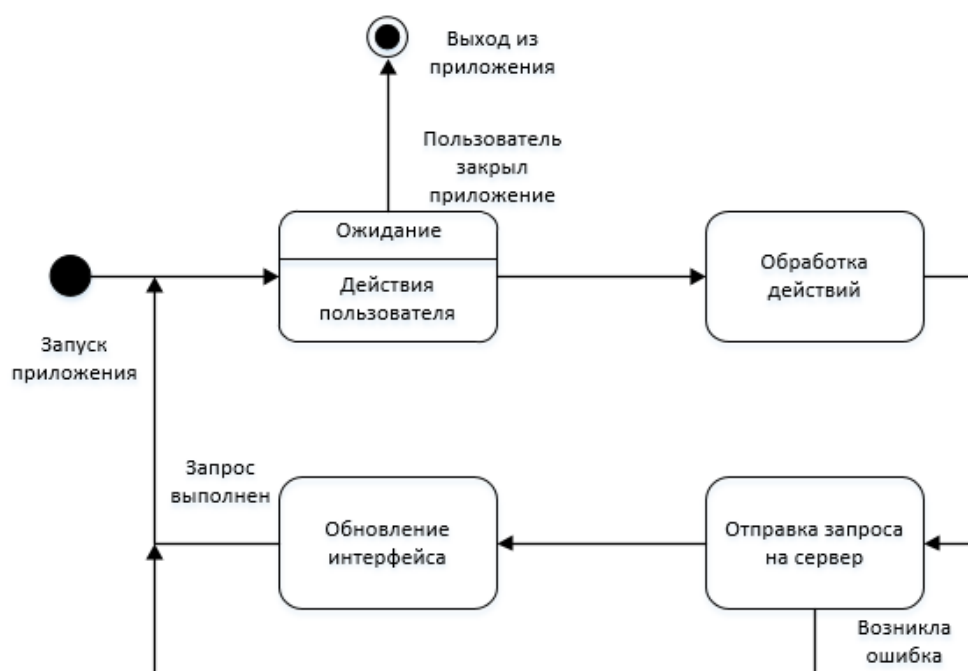


Рисунок 34 – Диаграмма состояний приложения

ПРИЛОЖЕНИЕ В
ПРЕЗЕНТАЦИЯ

					<i>ПГУ 1.090301.27.001 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		56

Тема работы: «Информационная система для учета квартир застройщика»

Научный руководитель:

к.т.н., доцент Карамышева Н. С.

Выполнил:

студент группы 15BB1 Федосеев Е. А.

ЦЕЛЬ РАБОТЫ

Цель работы – разработка информационной системы для учета квартир застройщика, предназначенной для быстрого отслеживания свободных, забронированных и купленных квартир, а также отслеживания договоров с клиентами.

					ПГУ 1.090301.27.001 ПЗ	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

СРЕДСТВА РАЗРАБОТКИ



- ☐ MS SQL Server;
- ☐ SQL;
- ☐ Visual Studio;
- ☐ C#;
- ☐ XAML.

3

АРХИТЕКТУРА БД

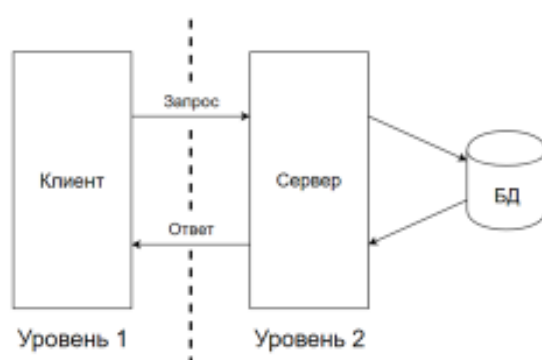


Рисунок 1 – Двухзвенная архитектура

- ☐ Производительность;
- ☐ Надежность;
- ☐ Безопасность;
- ☐ Простота.

4

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.27.001 ПЗ

Лист

58

WPF



Windows Presentation Foundation

Технология WPF – часть экосистемы платформы .NET и представляет собой подсистему для построения графических интерфейсов.



Преимущества:

- ☐ Языки .NET-платформы - C# и VB.NET;
- ☐ Язык разметки XAML;
- ☐ Независимость от разрешения экрана;
- ☐ Использование трехмерных моделей, привязка данных, стили, шаблоны, темы, анимации и т.д.
- ☐ Создание приложений под множество ОС семейства Windows.
- ☐ Аппаратное ускорение графики;

5

ПРИНИМАЕМЫЕ ПРИЛОЖЕНИЕМ СОСТОЯНИЯ



Рисунок 3 – Диаграмма состояний приложения

6

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.27.001 ПЗ

Лист

59

ИНТЕРФЕЙС ПРИЛОЖЕНИЯ

Рисунок 4 – Авторизация пользователя

Рисунок 5 – Регистрация пользователя

7

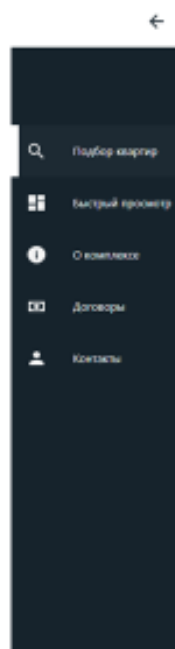


Рисунок 6 – Меню

Адрес	М/этаж	М/квартиры	Кол-во комнат	Общая площадь	Жил. площадь	Цена за кв.м	Статус	Статус
Иркутская, 1	1	1	1	40.7	15.1	35000	1344600	свободно
Иркутская, 1	1	7	1	40.7	15.1	35000	1344600	свободно
Иркутская, 1	2	8	1	40.6	15.1	35000	1334200	свободно
Иркутская, 1	2	14	1	40.6	15.2	35000	1334200	свободно
Иркутская, 1	3	10	1	40.6	15.2	35000	1334200	свободно
Иркутская, 1	3	21	1	40.6	15.2	35000	1334200	свободно
Иркутская, 1	4	22	1	40.6	15.2	35000	1334200	свободно
Иркутская, 1	4	28	1	40.6	15.1	35000	1334200	свободно
Иркутская, 1	5	28	1	40.6	15.1	35000	1330400	свободно
Иркутская, 1	5	20	1	40.6	15.1	35000	1330400	свободно
Иркутская, 1	6	26	1	40.6	15.1	35000	1331200	свободно
Иркутская, 1	6	42	1	40.6	15.1	35000	1331200	свободно

Рисунок 7 – Подбор квартир

8

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.27.001 ПЗ

Лист

60

Адрес	№ этажа	№ квартиры	Кол-во комнат	Общ. площадь	Жил. площадь	Цена за кв.м	Стоимость	Статус
Изумрудная, 1	2	9	2	51,8	26,2	38500	1994300	свободна
Изумрудная, 1	3	16	2	51,7	26,4	38500	1990450	свободна
Изумрудная, 1	5	30	2	51,8	26,3	38500	1994300	свободна
Изумрудная, 1	1	190	2	51,4	26,1	38000	1953200	свободна
Изумрудная, 1	2	197	2	51,6	26,2	38000	1958800	свободна

Рисунок 8 – Добавление в сравнение

WFF Design eugene@wff.ru

Номер дома: Изумрудная, 1

Цена: От 1500000 До 2000000

Общ. площадь: От 0 До 120

Номер этажа: От 1 До 20

Тип квартиры: ☒ Свободные квартиры

Детализация: Показать сравнение Сбросить Показать

Адрес	№ этажа	№ квартиры	Кол-во комнат	Общ. площадь	Жил. площадь	Цена за кв.м	Стоимость	Статус
Изумрудная, 1	3	16	2	51,7	26,4	38500	1990450	свободна
Изумрудная, 1	5	30	2	51,8	26,3	38500	1994300	свободна

Рисунок 9 – Сравнимые квартиры

ЗАКЛЮЧЕНИЕ ДОГОВОРА

Данные владельца

НАИМЕНОВАНИЕ ФИРМЫ
000 "Строй мастер"

КОНТАКТНОЕ ЛИЦО
Соловьева Эрика Васильевна

НОМЕР ТЕЛ. КОНТ. ЛИЦА
89568833517

Информация о квартире

АДРЕС ДОМА
Изумрудная, 1

НОМЕР КВАРТИРЫ
16

НОМЕР ЭТАЖА
3

КОЛИЧЕСТВО КОМНАТ
2

ОБЩАЯ ПЛОЩАДЬ
51,7

ЖИЛАЯ ПЛОЩАДЬ
26,4

СТОИМОСТЬ
1990450

Данные покупателя

ФАМИЛИЯ ПОКУПАТЕЛЯ
Рябов

ИМЯ ПОКУПАТЕЛЯ
Кирил

ОТЧЕСТВО ПОКУПАТЕЛЯ
Васильевич

Паспортные данные

СЕРИЯ
5656

НОМЕР
135791

АДРЕС
ул Глазнова 2 215

НОМЕР ТЕЛЕФОНА
89269631597

СТАТУС ДОГОВОРА
БРОНИРОВАНИЕ

КОММЕНТЫ
Рябов Кирил Васильевич

ЗАКЛЮЧИТЬ ДОГОВОР ДОБАВИТЬ КЛИЕНТА

Рисунок 10 – Форма бронирования квартиры

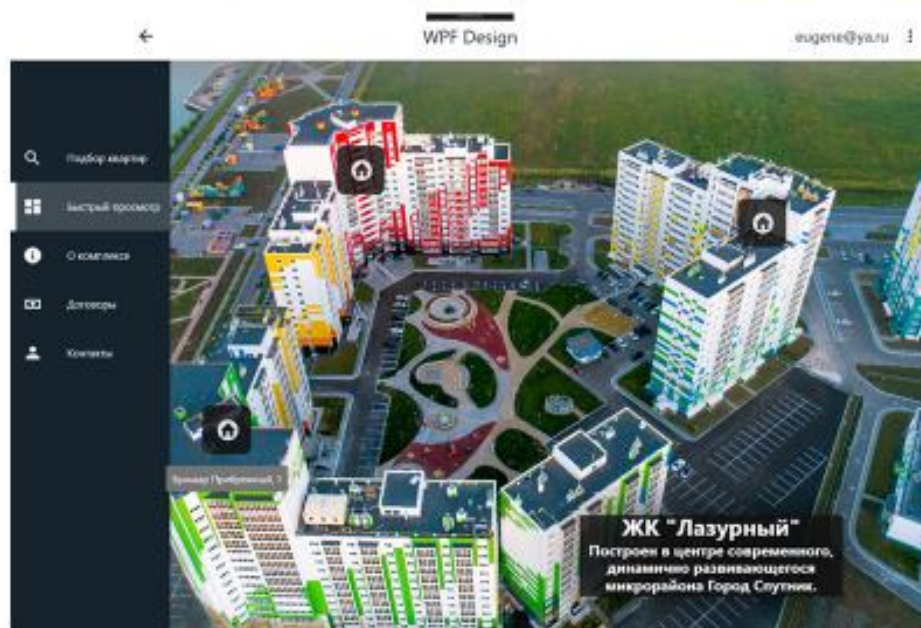


Рисунок 11 – Быстрый просмотр квартир

11

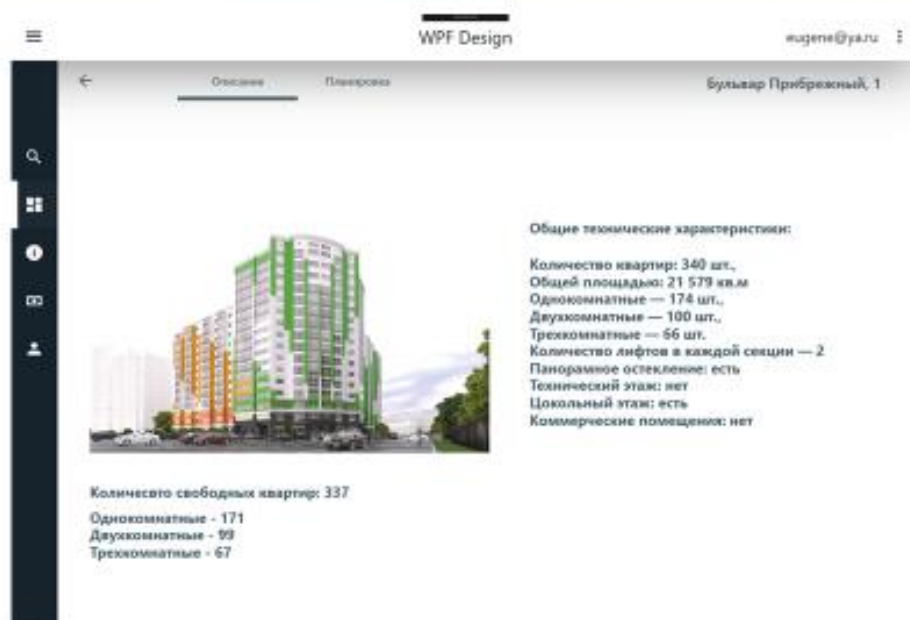


Рисунок 12 – Описание дома

12

					ПГУ 1.090301.27.001 ПЗ	Лист
						62
Изм.	Лист	№ докум.	Подпись	Дата		



Рисунок 13 – Планировка этажа

33



Рисунок 14 – Планировка этажа и описание квартиры

34

					ПГУ 1.090301.27.001 ПЗ	Лист
						63
Изм.	Лист	№ докум.	Подпись	Дата		

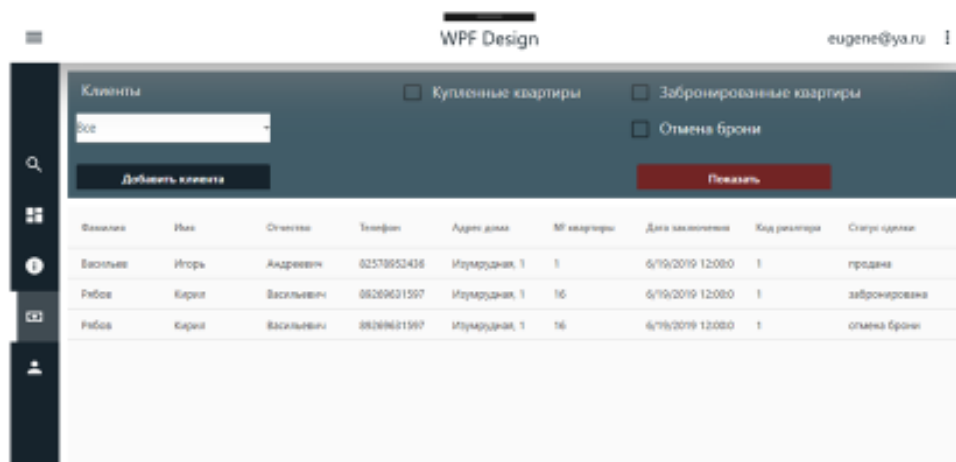


Рисунок 15 – Договоры

15

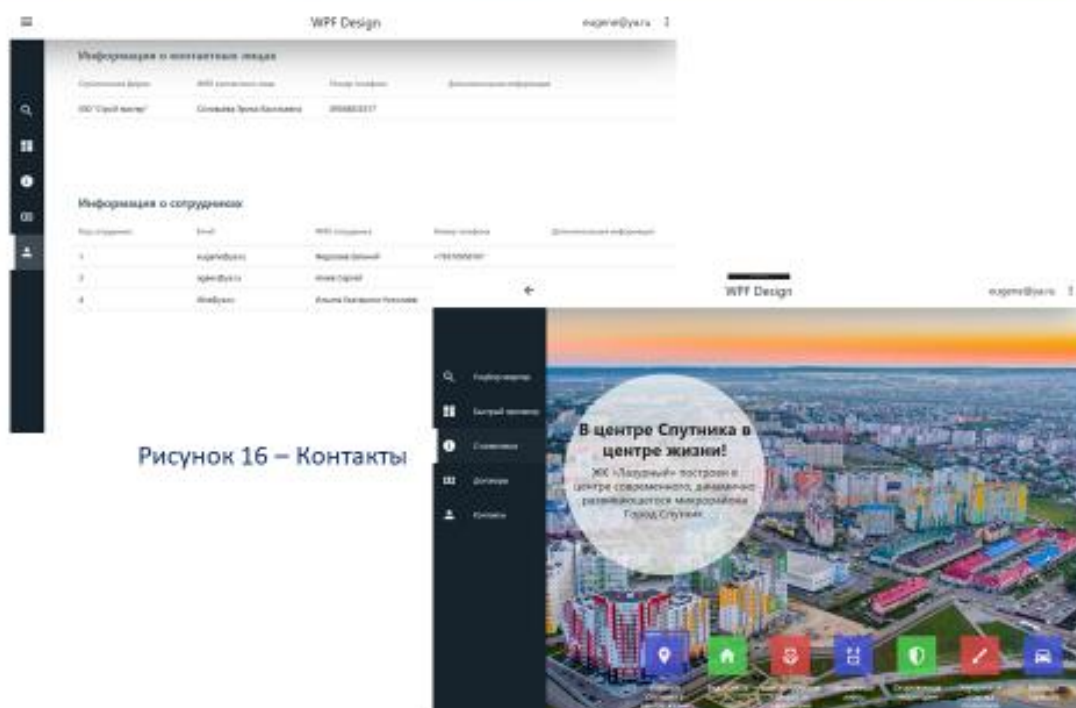


Рисунок 16 – Контакты

Рисунок 17 – О комплексе

16

					ПГУ 1.090301.27.001 ПЗ	Лист
						64
Изм.	Лист	№ докум.	Подпись	Дата		

ЗАКЛЮЧЕНИЕ

Результатом работы является:

- ☐ Информационная система с интуитивно понятным пользовательским интерфейсом;
- ☐ ИС позволяет пользователю вести учет квартир застройщика;
- ☐ Данные визуально отображаются в виде чисел, текста и таблиц;
- ☐ ИС полностью удовлетворяет поставленной цели.

38

СПАСИБО ЗА ВНИМАНИЕ!

					ПГУ 1.090301.27.001 ПЗ	Лист
						65
Изм.	Лист	№ докум.	Подпись	Дата		