

Содержание

Введение.....	4
Глава 1 Теоретическая часть.....	7
1.1 Предпроектной исследование предметной области	7
1.2 Характеристика инструментальных средств разработки.....	10
1.3 Анализ некоторых технологий, используемых при реализации программного комплекса.....	14
1.4 Анализ требований и определение спецификаций программного обеспечения.....	19
Глава 2 Практическая часть	22
2.1 Постановка задачи.....	22
2.2 Проектирование программного обеспечения.....	25
2.3 Разработка пользовательских интерфейсов программного обеспечения.....	25
2.4 Тестирование и отладка программного обеспечения.....	26
2.5 Руководство по использованию программного обеспечения.....	26
Заключение	27
Список использованных источников	28
Приложение А	29
Приложение Б.....	30
Приложение В	31

Введение

Цель данной выпускной квалификационной работы заключается в разработке автоматизированной информационной системы фирмы по ремонту квартир. В ход выполнения данной работы были приобретены и подтверждены навыки, приобретённые в ходе обучения по специальности «Программирование в компьютерных системах» в Колледже информатики и программирования.

Данная автоматизированная информационная система необходима для автоматизации некоторых процессов в работе небольшой фирмы, выполняющей ремонты квартир

Система будет состоять из приложения под операционную систему Windows, Rest API сервиса, который будет работать на удаленном сервере, что позволит работать нескольким работникам одновременно, а также база данных, с которой будет держать связь Rest API сервис.

Основной задачей будет автоматизация и повышения эффективности работы персонала в рамках определенных задач.

Стоит отметить, данная автоматизация решит вопросы, связанные с количеством времени, которое тратиться на создание заказа на ремонт, а также на некоторые другие технологические операции. После решения данных вопросах сократиться время обслуживания клиентов и повысится качество данного обслуживания, а также повысится обратная отдача между клиентом и теми работниками, которые будут выполнять данный ремонт.

Процесс автоматизации всех процессов в настоящее время с развитием технологий приобрело большой интерес у некоторых директоров и в целом у всего руководящего персонала, который хочет повысить качество и скорость обслуживания клиентов.

Как уже говорилось ранее, цели данного проекта заключается в разработке автоматизации информационной системы фирмы по ремонту

квартир, в котором будет реализован набор всех требуемых для данной программы функций.

Для достижения данной цели передо мной стоят следующие задачи:

- проектирование базы данных как для сервера, так и для локального компьютера;
- созданием универсальных методов для обработки ошибок и работы в рамках работы сервера;
- описание необходимых моделей для работы с клиентским приложением;
- создание контроллеров обработки данных поступающих от клиентского приложения;
- разработка интерфейсов для работы клиентского приложения;
- разработка модуля для загрузки справочных данных на сервер;
- создание необходимых обработок для работы со сторонним сервером;
- создание необходимых шаблонов для вывода документов о смете и договоре по каждому заказу;
- написание пояснительной о программе.

Для решения перечисленных выше задач в процессе работы над созданием мобильной игры необходимо пройти следующие стадии и этапы разработки:

- предпроектное исследование предметной области;
- выбор инструментальных средств разработки программы;
- проектирование игрового процесса;
- разработка пользовательского интерфейса;
- тестирование и отладка программного обеспечения;
- разработка справочной системы;
- разработка руководств по использованию программы.

На каждом этапе либо стадии разработки данной работы будут применяться различные инструментальные средства разработки программного обеспечения:

- Visual Studio 2019 Enterprise;
- draw.io;
- система управления базами данных DB Browser for SQLite;
- сервер базы данных SQL Server Management Studio;
- система управления базами данных SQL Server Management Studio.

Глава 1 Теоретическая часть

1.1 Предпроектное исследование предметной области

1.1.1 Анализ предметной области объекта автоматизации

Данный проект разработан в целях увеличения скорости работы компании и представляет из себя набор клиент серверного программного обеспечения:

- Базы данных для хранения все информации;
- Сервера, который выполняет прослойку между базой данных и прочими программа в данном комплексе;
- Приложения для операционной системы Windows.

Для актуализации понимания о предметной области были изучены программы аналоги и были изучены необходимые и важные функции:

- подсчет стоимости заказа на ремонт;
- создание договора на основе шаблона, созданного заранее;
- импорт информации о материалах и услугах, предоставляемых данной организацией из файла программы Excel;
- возможность добавления, редактирования информации об услугах;
- возможность работы с данными о сотрудниках (прием на работу, увольнение и кадровые перемещения) с документированием в формате Word;
- возможность отдельного доступа к программе путем назначения прав доступа для сотрудников, а также возможность их редактирования;
- учет данных о клиентах и сотрудниках, а также редактирование данных об этом факте;
- учет оказанных услуг и учет платежей, а также система отчетов о проделанных работах и проданных услугах;
- фиксирование замеров помещений и вычисление площади некоторых поверхностей.

1.1.2 Актуальность разрабатываемого программного обеспечения

Актуальность данного программного обеспечения заключается в том, что в настоящее время очень маленькое количество компаний занимается разработкой данного специализированного программного обеспечения для автоматизации работы фирм по ремонту квартир, но большое количество компаний требуются в данном программном обеспечении.

Основной причиной актуальности данного программного обеспечения можно считать рост рынка ремонтно-строительных работ, который стабилизировался в условиях нынешних экономических потрясений и состояния экономики России, связанных с ростом количества жилья сданных без отделки квартир относительно квартир сданных с минимальным ремонтом, а также в следствии роста количества вторичного жилья требуемого ремонта в том числе и капитального.

Стоит отметить, что в связи с сложными кризисными временами отделочники-непрофессионалы отсеиваются с рынка быстрее и уступают место специалистам, которые делают качественный ремонт за адекватную цену, что подталкивает людей платить нежелая делать ремонт самостоятельно.

[1]

1.1.3 Обзор существующих аналогов

В качестве существующих аналогов можно рассмотреть похожую по концепцию программу распространяемую платно по средствам лицензии и разработанную компанией «ПростойСофт»[2], конфигурацию «Ремонт квартир»[3] которая устанавливается на базовую версию их программного обеспечения под название «Учет клиентов».

Данное программно обеспечение имеет следующие основные функции:

- учет клиентов;
- учет событий;
- формирование отчетов;
- хранение справочников;

- формирование документов по шаблонам;
- учет платежей.

Стоит отметить, что данная система является слишком громоздкой и не может работать на нескольких компьютерах одновременно так как база располагается локально на компьютере, таким образом, представляется нецелесообразным использование данной программы невозможно при работе нескольких людей одновременно.

Также стоит отметить, что данное программное обеспечение поставляется за 8000 рублей за одно рабочее место, что не соизмеримо с функционалом при покупке более чем одной копии в компанию.

Также приведу фотографию пользовательского интерфейса которые представлены на рисунке 1

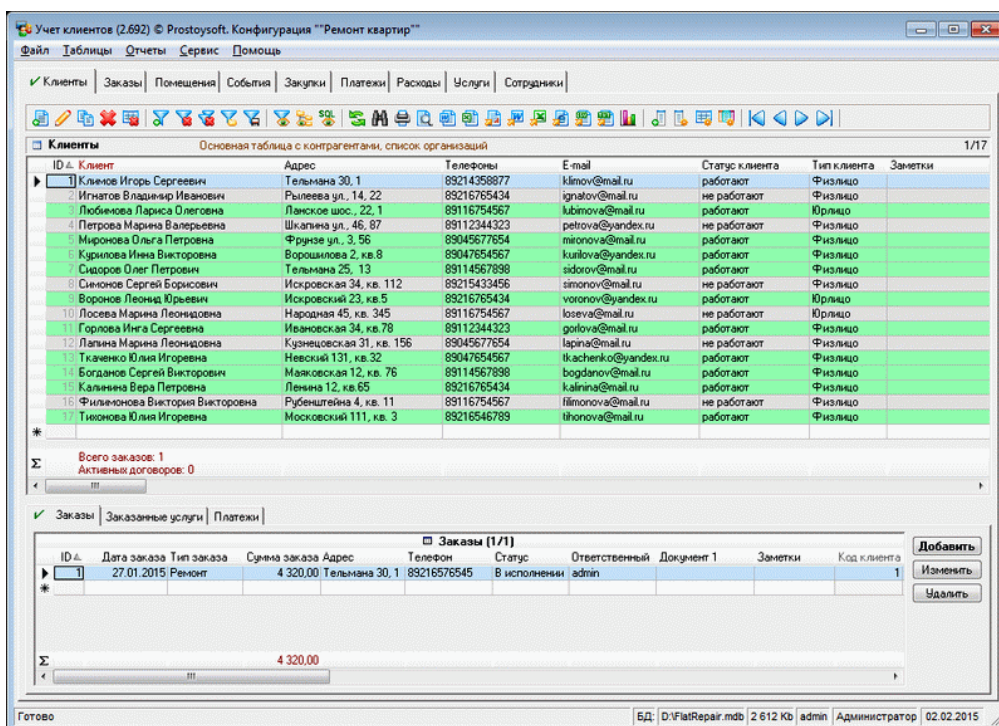


Рисунок 1— Пример пользовательского интерфейса аналога

Также аналогами данного программного комплекса можно представить некоторые конфигурации 1С Предприятия, но они помимо того, что имеют изначально не маленькую стоимость, так еще и потребуют некоторых доработок за отдельную сумму денег.

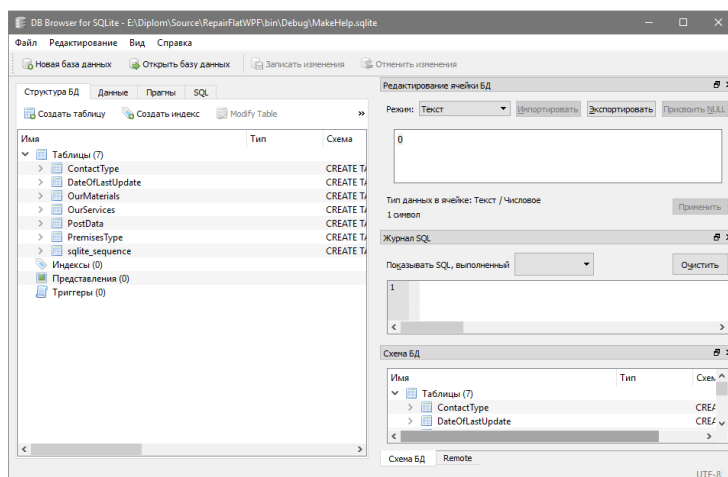


Рисунок 3 - Интерфейс DB Browser for SQLite

Кроме локальной базы данных, также встала необходимость использования серверной системы управления базами данных. В качестве данной системы управления базами данных был выбран «Microsoft SQL Server»[6]. Данная система управления базами данных была выбрана по причине простоты развертки данных баз данных, как на Linux, так и на Windows серверах, что повышает совместимость и уменьшает стоимость развертки данной системы управления базами данных на сервере.

В качестве инструмента взаимодействия с серверной системой управления базами данных был выбран «SQL Server Management Studio»[7]. Данное приложение представляет полный доступ к возможностям Microsoft SQL Server» через удобный пользовательский интерфейс пример которого представлен на рисунке 4.

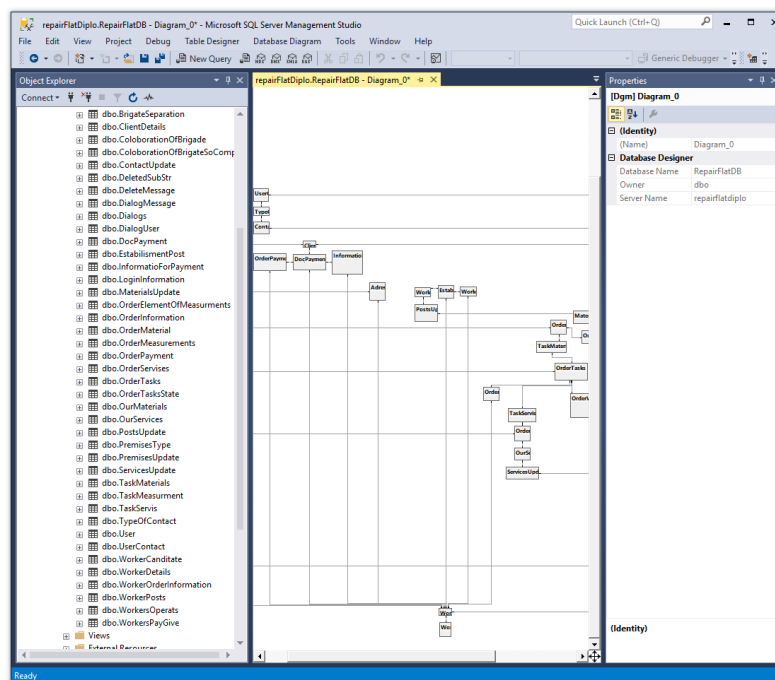


Рисунок 4 - Интерфейс SQL Server Management Studio

В качестве среды разработки была выбрана Visual Studio 19 с, удобным для пользователя интерфейсом и наличием отладчика кода.

Программирование выполнялось на языке программирования C#, тип проекта:

- для Windows приложения — проект графического окна, основанного на технологии WPF;
- для серверной части данного комплекса — ASP.NET Web Application (.NET Framework);
- для создания базы данных — проект базы данных.

Visual Studio 2019 (VS) представляет собой интегрированную среду разработки (Integrated Development Environment, IDE).

Пример пользовательского интерфейса VS представлен на рисунках 5-6. Стоит отметить, что в новой версии VS был переработан дизайн некоторых объектов, пример интерфейса которых и приведен ниже.

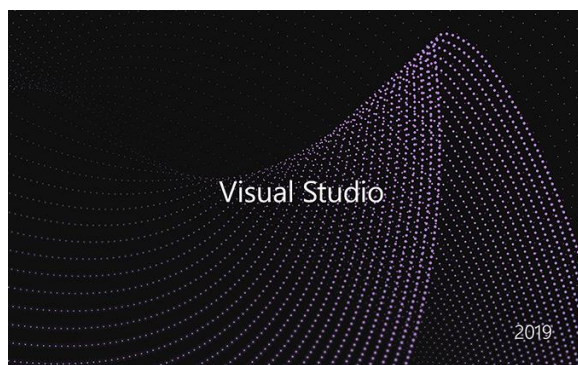


Рисунок 5 – Обновленный интерфейс загрузки Visual Studio 2019

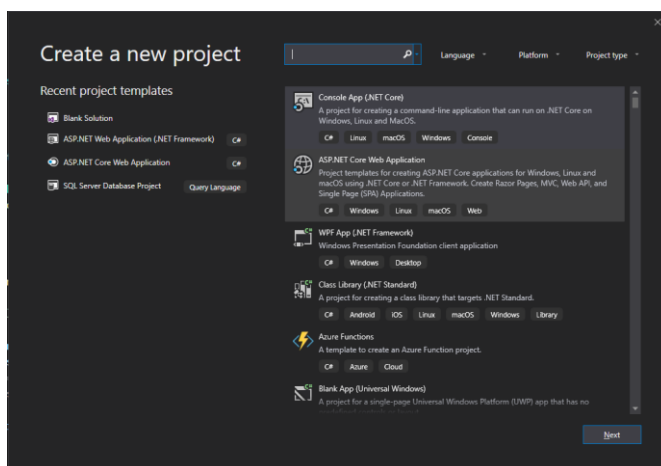


Рисунок 6 – Обновленный интерфейс выбора типа приложения Visual Studio 2019

С помощью VS многие из этих рутинных и трудоемких задач, связанных с повседневной работой программиста, можно упростить и облегчить.

В состав VS входит целый набор типовых проектов, из которых каждый разработчик может подобрать именно то, что ему в данный момент требуется. Каждый раз, когда вы создаете новый проект, VS автоматически создаст "скелет" будущего приложения, причем этот код можно немедленно скомпилировать и запустить на исполнение. В составе каждого типового проекта имеются элементы, которые по желанию добавлять в ваш проект.

Любой проект, в любом случае, содержит автоматически сгенерированный код, который представляет собой основу будущей программы. VS предлагает множество готовых к использованию элементов управления, включая и код, необходимый для их создания. Это экономит время разработчиков, избавляя их от необходимости каждый раз заново

создавать типовой программный код для решения часто встречающихся задач.
[8]

1.2.2 Характеристика языка программирования

Дипломная работа выполнена с использованием объектно-ориентированного языка программирования C# и некоторых сторонних библиотеках, предназначенных для данного языка программирования.

C# — язык программирования, сочетающий объектно—ориентированные и контекстно-ориентированные концепции. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств вроде Visual Studio.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML.[9]

1.3 Анализ некоторых технологий, используемых при реализации программного комплекса

1.3.1 Технология Entity Framework

Разработка автоматизированной информационной системы подразумевает использование базы данных как средство хранения обработанной информации.

Для работы с базами данных можно использовать язык запросов SQL. Данный язык программирования разработан для управления реляционными базами данных. Язык был создан в 1970х годах и назывался «SEQUEL» и был предназначен для управления системой баз данных System R. Через некоторое

время данный язык был переименован «SQL» и в 1979 году был опубликован в качестве коммерческого продукта Oracle V2.

В 1986 году был создан первый официальный стандарт языка, после чего были созданы еще несколько стандартов языка. Несмотря на существование стандартов, большинство средств управления СУБД используют встроенный язык запросов, который иногда очень кардинально отличается от стандартов это объясняется большим объемом и сложностью описанного стандарта, а также нехваткой в нем некоторых спецификаций в важных областях.[10]

При реализации программного комплекса появится необходимость работы с запросами. Данную работу можно проводить многими способами, но помимо использования языка T-SQL, который необходимо было бы использовать по причине использования в качестве системы управления базами данных «Microsoft SQL Server». Использование данного языка запросов может привести к ошибкам после переименования некоторых полей, что возможно при разработке системы при одновременном проектировании данной.

В целях обеспечения безопасности была выбрана соответствующая технология, а именно «ADO.NET Entity Framework».

Данная технология представляет возможность работы с базами данных как работу с некоторыми сущностями. Сущностями в данном случае называют некий набор данных, которые ассоциативны с определенным объектом, но считать таблицы базы данных набором сущностей не совсем корректно, так как работа выполняется с объектами и их наборами.

При этом сущности могут быть связаны ассоциативной связью один-ко-многим, один-ко-одному и многие-ко-многим, подобно тому, как в реальной базе данных происходит связь через внешние ключи.

Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ мы можем не только извлекать определенные строки, хранящие объекты, из базы данных, но и получать объекты, связанные различными ассоциативными связями.

Еще одним ключевым понятием Entity Data Model. Эта модель сопоставляет классы сущностей с реальными таблицами в БД.

Entity Data Model состоит из трех уровней:

- концептуальном на котором происходит определение классов сущностей, используемых в приложении;
- уровень хранилища, который определяет таблицы, столбцы, отношения между таблицами и типами данных, которыми сопоставляется используемая база данных;
- уровень сопоставления (маппинга) который служит посредником между свойствами класса и сущности и столбцами таблицы.

Таким образом, мы можем через классы, определенные в приложении, взаимодействовать с таблицами из базы данных.[11]

Для работы с Entity Framework необходимо изначально скачать NuGet пакет Entity Framework. Для скачивания NuGet пакетов необходимо правой кнопкой нажать на название своего проекта на платформе .Net, после чего выбрать пункт, выделенный на рисунке 7. После чего перейти на первую вкладку «Browse» и в строку поиска ввести Entity Framework и в результатах выбрать пункт «EntityFramework» и нажать кнопку «Install» пример выбора данного пакета представлен на рисунке 8.

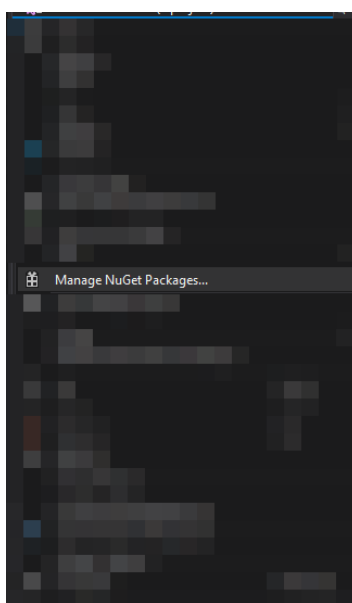


Рисунок 7 - Открытие менеджера пакетов NuGet

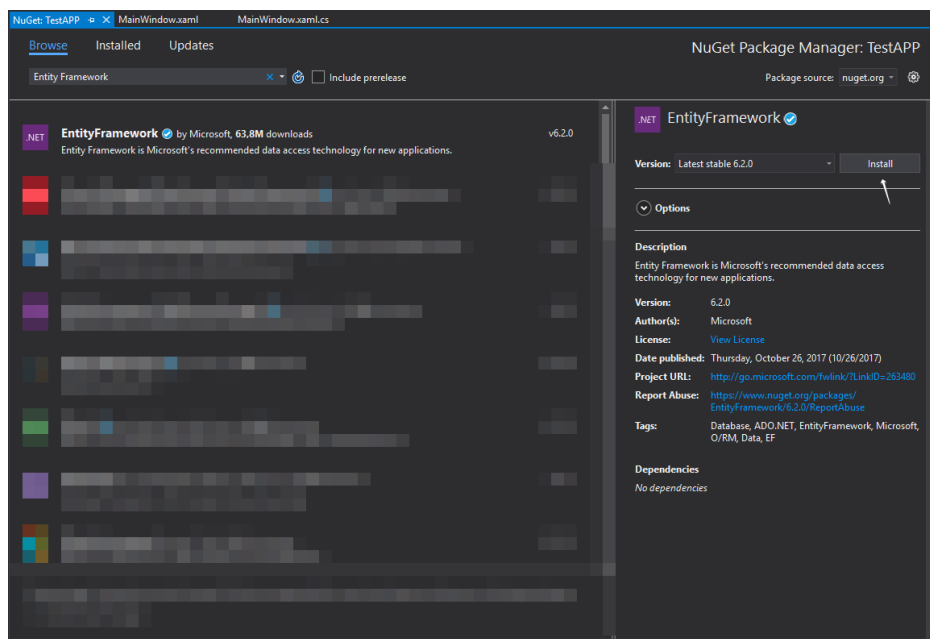


Рисунок 8 – Выбор необходимого пакета

После установки пакета необходимо в проект добавить объект «ADO.NET Entity Data Model» который добавляется путем нажатия правой кнопки на название проекта на платформе .Net после чего выбрать пункт «Добавить» затем «Новый элемент» после чего перейти в раздел «Данные» и выбрать пункт «ADO.NET Entity Data Model» после чего ввести название и нажать кнопку добавить пример добавления представлен на рисунке 9.

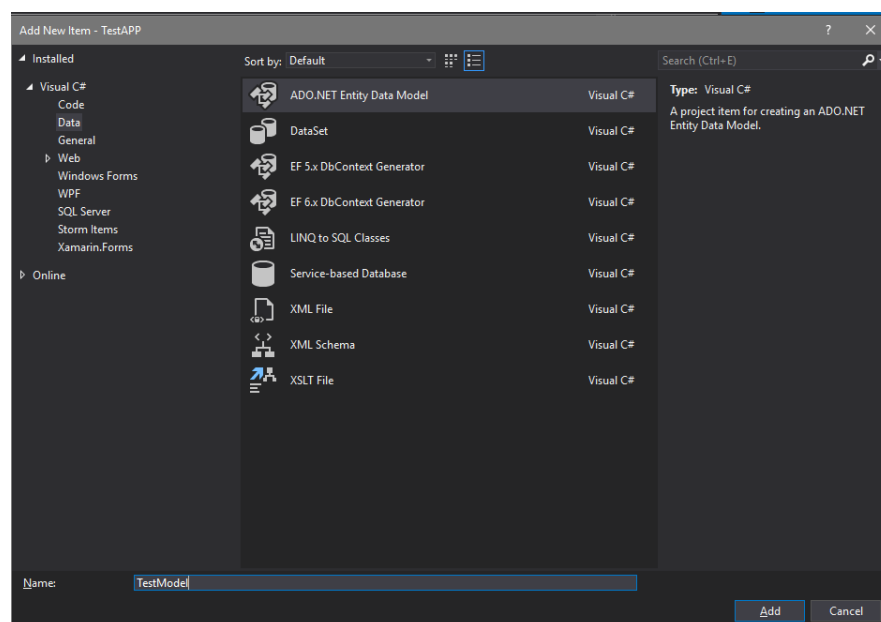


Рисунок 9 – Добавление «ADO.NET Entity Data Model» в проект

После нажатия кнопки добавить необходимо нажать далее, после чего указать строку подключения к базе данных путем нажатия кнопки «New Connection» на форме, представленной на рисунке 10. После указания строки подключения необходимо выбрать пункт «Yes, include sensitive data in the connection string», после чего нажать «Далее» и в открывшемся окне выбрать необходимые объекты для копирования в базу данных пример выбора данных представлен на рисунке 11, после чего нажать «Выполнить».

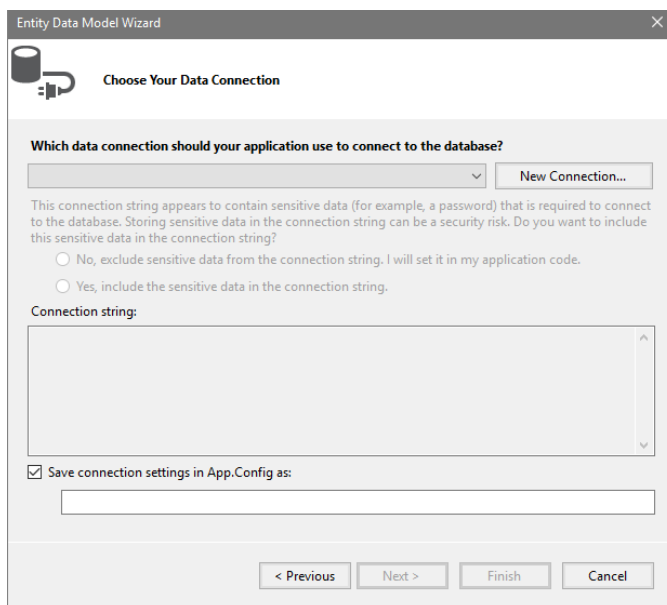


Рисунок 10 – Окно для указания строки подключения

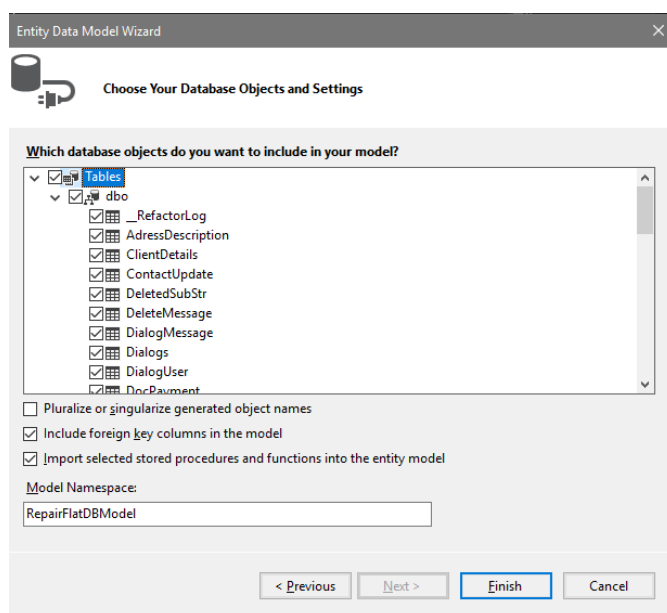


Рисунок 11 – Окно выбора необходимых компонентов базы данных

Через некоторое время данные будут скопированы и представлены в решении в соответствующем пункте в качестве примера, представленного на рисунке 12 указана модель, которая была скопированы из базы данных.

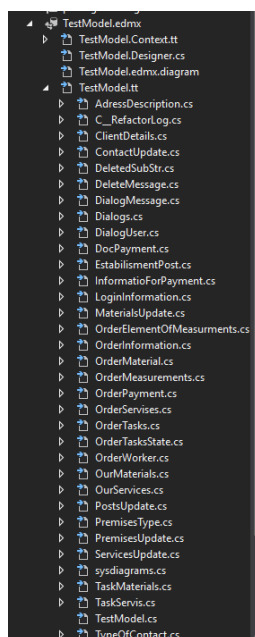


Рисунок 12 – Пример отображения классов, отображающих
«сущности»

1.4 Анализ требований и определение спецификаций программного обеспечения

Диаграмма вариантов использования клиентского приложения представлена на рисунке БЮ. Данная диаграмма описывает работу приложения по соответствующим сценариям, которые прописаны в зависимости от роли, которая выданная определенному логину и паролю.

Рисунок 13 - Диаграмма вариантов использования системы

Контекстная диаграмма потоков данных, которая представлена на рисунке БЮ. Описывает взаимодействие между пользователями и сервером обработки информации посредством клиентского приложения.

Рисунок 14 - Контекстная диаграмма потоков данных

На данной диаграмме отображены данные, которые предоставляются пользователем:

- о работниках;
- о клиентах;
- о деталях заказа;
- о справочной информации.

Программа в качестве ответа демонстрирует информацию, о том, что данные были добавлены, а также демонстрирует общий список данных, которые необходимы в данном контексте.

Также ниже представлена детализированная диаграмма потоков данных в программном комплексе, представленная на рисунке БЮ. Данная диаграмма детализирует контекстную диаграмму, приведенную выше.

Рисунок 15 - Детализированная диаграмма потоков данных

Для разрабатываемого программного комплекса была составлена контекстная функциональная диаграмма, описывающая схему взаимодействия серверного модуля с прочими модулями и представлена она на рисунке БЮ. На данной диаграмме показано, что в качестве основного модуля можно считать модуль сервера, который обрабатывает все необходимое для работы клиента. Сервер в свою очередь выполняет полную связь с базой данных.

Рисунок 16- Контекстная функциональная диаграмма

Далее была составлена детализированная функциональная диаграмма, представленная на рисунке БЮ. Данная диаграмма описывает основные функции со стороны серверной части при стандартной работе пользователя с системой, аналогичной детализированной диаграммы потоков данных.

На вход при стандартной работе подается строка подключения, которая берется из настроек и может быть изменена, а также данные от пользователя, либо вызов соответствующей операции по ссылке.

Рисунок 17 - Детализированная функциональная диаграмма

Глава 2 Практическая часть

2.1 Постановка задачи

2.1.1 Функции программного продукта

Среди основных функций разрабатываемого программного обеспечения можно отметить следующие:

- подсчет стоимости заказа на ремонт;
- создание договора на основе шаблона, созданного заранее;
- импорт информации о материалах и услугах, предоставляемых данной организацией из файла программы Excel;
- возможность добавления, редактирования информации об услугах;
- возможность работы с данными о сотрудниках (прием на работу, увольнение и кадровые перемещения) с документированием в формате Word;
- возможность раздельного доступа к программе путем назначения прав доступа для сотрудников, а также возможность их редактирования;
- учет данных о клиентах и сотрудниках, а также редактирование данных об этом факте;
- учет оказанных услуг и учет платежей, а также система отчетов о проделанных работах и проданных услугах;
- фиксирование замеров помещений и вычисление площади некоторых поверхностей.

2.1.2 Описание входной информации

В качестве входной информации можно отметить следующие основные данные необходимые для работы приложения:

- данные о сотрудниках и клиентах;
- данные о заказе в том числе размеры помещений;
- данные о материалах и услугах, предоставляемых данной компанией;

- данные о должностях в данной компании;
- данные о выполненных этапах и услугах;
- данные о платежах и выплатах.

2.1.3 Описание выходной информации

В качестве выходной информации в данной программе служат отчеты об операциях, выполняемых в данной АИС, а также смета и договор по каждому заказу.

2.1.4 Требование к программному обеспечению

Системные требования

Необходимыми программными средствами для работы будут являться:

- операционная система Windows 10;
- приложение Microsoft Word и Microsoft Excel;
- наличие .Net Framework 4.7 версии и выше.

Необходимыми аппаратными средствами является:

- процессор с тактовой частотой 2,7 ГГц или выше. Рекомендуется использовать двухъядерный процессор;
- 1 ГБ ОЗУ (3 ГБ при выполнении в виртуальной машине);
- 10 ГБ доступного пространства на жестком диске;
- видеоадаптер с поддержкой DirectX 9 (разрешение 1600 x 900 или больше);
- монитор с разрешением не меньше, чем (1600 x 900);
- клавиатура и мышь.

Функциональные и нефункциональные требования

В качестве функциональных требований можно считать следующие:

- программа должна осуществлять добавление, редактирование, удаление, просмотр информации, а также ограничить доступ к данным посторонним лицам;
- работать с заданным алгоритмом функционирования;

- поддерживать диалоговый режим в рамках предоставляемых пользователю возможностей;

- производить бесперебойную работу по преобразованию информации.

В качестве нефункционального требования можно считать следующее:

- требования к интерфейсу.

Требования к реализации

Данная АИС должна быть реализована с использованием языка программирования С# и принципов объектно ориентированного программирования.

Требования к надёжности

В программном продукте предусмотрена защита от действий, пользователя которые могли бы были привести к выдаче ошибки и экстренного закрытия программного продукта:

- защита от SQL- инъекций;

- защита от некорректного ввода значений в программе;

Требования к интерфейсу

Интерфейс данной АИС должен быть понятен пользователю и не иметь в себе что-то раздражающее так как данный программный продукт пользователь будет использовать большую часть рабочего времени.

Требования к программной документации

Программная документация оформляется в соответствии с требованиями документа «МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОФОРМЛЕНИЮ КУРСОВОЙ И ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ» и указанными в нём ГОСТами.

2.2 Проектирование программного обеспечения

2.2.1 Проектирование REST API сервера

2.2.2 Проектирование баз данных

2.3 Разработка пользовательских интерфейсов программного обеспечения

Разработка пользовательского интерфейса являлась одной из основной стадией работы над проектом по причине необходимости работы приложения в адаптивном режиме.

Для реализации данного требования со стороны заказчиков программного обеспечения была выполнена разработка основного окна который повторял бы все стандартные функции окна WPF проекта C#.

Среди основных функций можно считать следующие:

- возможность перемещать окно путем зажатия шапки окна;
- возможность сворачивать окно путем нажатия на соответствующую кнопку в шапке окна;
- возможность разворачивать окно на весь экран путем нажатия на соответствующую кнопку в шапке окна;
- возможность разворачивать окно путем двойного нажатия на его шапку.

Все вышеперечисленные требования удалось решить путем использования следующих библиотек:

- `Microsoft.Windows.Shell` – предоставляет возможность использовать стандартные функции окна windows там, где это не предусмотрено изначально, что позволяет использовать данную библиотеку как средство работы с окном;
- `Fody` – необходима для синхронизации работы окна и внутренних свойств некоторых классов, которые будут использоваться в качестве классов для расчета некоторых свойств окна.

При использовании данного набора библиотек было создано адаптивное окно, представленное на рисунке БЮ.

Рисунок 18 - Базовое окно приложения

Пример адаптивности окна представлена на рисунке БЮ-БЮ.

Рисунок 19 – Пример адаптивности окна

Рисунок 20 - Пример адаптивности окна

В качестве

2.4 Тестирование и отладка программного обеспечения

2.5 Руководство по использованию программного обеспечения

Заключение

Список использованных источников

Интернет-ресурсы

1. Рынок ремонтно-строительных работ: тенденции и прогнозы – [Электронный ресурс]. – Режим доступа: <https://www.vira.ru/exp/news/detail.php?ID=48005>
2. Простой Софт - разработка программ– [Электронный ресурс]. – Режим доступа: <http://www.prostoysoft.ru/index.htm>
3. Конфигурация "Ремонт квартир" – [Электронный ресурс]. – Режим доступа: <http://www.prostoysoft.ru/FlatRepair.htm>
4. Draw.io – [Электронный ресурс]. – Режим доступа: <https://www.draw.io/>
5. DB Browser for SQLite – [Электронный ресурс]. – Режим доступа: <https://sqlitebrowser.org/>
6. SQL Server 2017 — Выпуски – [Электронный ресурс]. – Режим доступа: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2017-editions>
7. Скачивание SQL Server Management Studio (SSMS) – [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017>
8. Описание среды программирования VS – [Электронный ресурс]. – Режим доступа: <https://studfiles.net/preview/4114690/>
9. C# Язык программирования – [Электронный ресурс]. – Режим доступа: <http://progopedia.ru/language/csharp/>
10. SQL Язык программирования – [Электронный ресурс]. – Режим доступа: <http://progopedia.ru/language/sql/>
11. Введение в Entity Framework 6– [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/entityframework/1.1.php>

Приложение А

Приложение Б

Приложение В