МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Специализация Программирование интернет-приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированное программирование»

Тема Программное средство «Личный дневник»

Исполнитель

Студент (ка) 2 курса группы 4 Лещенко Анастасия Олеговна

(Ф.И.О.)

Руководитель Радиванович Дмитрий Александрович

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

Курсовой проект защищен с оценкой

Председатель Пацей Н.В.

(подпись)

Минск 2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПЕУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра программной инженерии

Утверждаю

Заведующий кафедрой

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

подпись инициалы и фамилия

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018г.

**ЗАДАНИЕ**

**к курсовому проектированию**

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

|  |  |
| --- | --- |
| Специальность: 1-40 01 01Программное обеспечение информационных технологий  Студент: Лещенко А.О. | Группа:\_\_4\_\_ |
| **Тема: Программное средство «Личный дневник»** | |

**1. Срок сдачи студентом законченной работы**: "20 мая 2018 г."

**2. Исходные данные к проекту:**

**2.1**. Функционально ПС поддерживает:

* Выполняет основные функции дневника:
  + Поддержание работы c базой данных;
  + Создание, изменение и удаление новой заметки.
* Выполняет дополнительные функции:
  + Выполнять регистрацию, авторизацию и восстановление пароля;
  + Заполнять, изменять и удалять информацию о контактах;
  + Добавлять, изменять и удалять напоминания;
  + Добавлять фотографии к заметкам.

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

**3. Содержание расчетно-пояснительной записки**

(перечень вопросов подлежащих разработке)

* Введение
* Постановка задачи (алгоритмы решения, обзор прототипов, актуальность задачи)
* Разработка архитектуры проекта (структура классов).
* Разработка функциональной модели и модели данных ПС (выполняемые функции)
* Тестирование
* Заключение
* Список используемых источников
* Приложения

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

* + Теоретическая часть курсового проекта должны быть представлены в формате docx. Оформление записки должно быть согласно выданным правилам.
  + Листинги программы представляются частично в приложении.
  + К записке необходимо приложить CD (DVD), который должен содержать: пояснительную записку, листинги и инсталляцию проекта.

***Календарный план***

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта | Примечание |
| 1 | Введение | 19.02.2018 |  |
| 2 | Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования | 12.03.2018 |  |
| 3 | Анализ и проектирование архитектуры приложения (построение диаграмм, проектирование бизнес-слоя, представления и данных) | 26.03.2018 |  |
| 4 | Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса | 2.04.2018 |  |
| 5 | Кодирование программного средства | 23.04.2018 |  |
| 6 | Тестирования и отладка программного средства | 30.04.2018 |  |
| 7 | Оформление пояснительной записки | 7.05.2018 |  |
| 9 | Сдача проекта | 20.05.2018 |  |

**5. Дата выдачи задания \_\_\_\_**9.02.2018**\_\_\_\_**

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

(подпись)

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

(дата и подпись студента)

**Оглавление**

[**Введение** 5](#_Toc515470919)

[**1. Постановка задачи** 6](#_Toc515470920)

[**1.1 Актуальность решаемой задачи** 6](#_Toc515470921)

[**1.2 Средства разработки** 6](#_Toc515470922)

[**2. Разработка архитектуры проекта** 7](#_Toc515470923)

[**2.1 Постановка задачи** 7](#_Toc515470924)

[**2.2 Обобщенная структура** 8](#_Toc515470925)

[**2.3 Проектирование базы данных** 10](#_Toc515470926)

[**3. Руководство пользователя** 13](#_Toc515470927)

[**4. Тестирование** 22](#_Toc515470928)

[**Заключение** 24](#_Toc515470929)

[**Список используемых источников** 25](#_Toc515470930)

[**Приложение А** 26](#_Toc515470931)

[**Приложение Б** 27](#_Toc515470932)

[**Приложение В** 28](#_Toc515470933)

# **Введение**

В наше время у всех людей есть много планов, мыслей, задумок. Запомнить большое количество информации иногда становится проблематичным. Монотонная работа сильно нагружает, а также может привести к стрессу. Не всегда есть возможность с кем-то поделиться волнующей нас темой. Для этого служит личный дневник, где можно поместить все, что угодно, начиная от простых мыслей и переживаний, заканчивая планами на будущее, историей прогресса, важными напоминаниями и запоминающимися событиями.

Программное средство «Личный дневник» отлично поможет в решении данных проблем. Здесь есть возможность записать свои мысли, важные моменты, произошедшие за день.

Также есть возможность вести информацию о своих контактах, будь то просто знакомые, коллеги или друзья. Возможно хранения различной информации: телефоны, фотографии, почтовые ящики и т.д.

С возрастом количество дел на день, месяц, год увеличивается, и все события не всегда легко запомнить, для этого в личном дневнике могут быть добавлены напоминания.

Целью курсового проекта является создание своеобразного «органайзера», нацеленного на записывание и хранение личной информации, связанного с базой данных.

Базой данных называется набор логически связанных данных, который предназначен для удовлетворения информационных потребностей организации/предприятия. Для выполнения задания, поставленного в курсовой работе, необходимо освоить работу по проектированию баз данных и работу с базами данных в выбранной среде программирования.

В качестве интерфейса прикладного программирования был выбран обширный API-интерфейс — Windows PresentationFoundation (WPF), предназначенный для создания настольных программ с графически насыщенным пользовательским интерфейсом.

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

Кроме того, используемая в WPF модель разделения кода и дизайна предполагала работу с декларативным языком описания интерфейса — eXtensibleApplicationMarkupLanguage (XAML).

1. **Постановка задачи**
   1. **Актуальность решаемой задачи**

Существует мнение от том, что личный дневник создан только для маленьких девочек, но я считаю, что это не так.

На мой взгляд, дневник — это то, о чём вам нужно подумать прямо сейчас. Это ваше окно в мир воспоминаний. В мир, который нельзя доверять лишь мозгу.

Дневник помогает смотреть в прошлое. Вы можете открыть свою запись, которую сделали несколько лет назад, и посмеяться над собой, или поплакать, вспомнить об эмоциях, которые обуревали вас в то время, и о том, какие события их вызвали.

* 1. **Средства разработки**

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2017;
* программная платформа .NET Framework 5.0;
* язык программирования C#;
* расширяемый язык разметки XAML;
* технология WPF;
* Entity Framework;
* Material Design Library;
* шаблон (паттерн) проектирования архитектуры приложения Repository;
* паттерн проектирования Unit of Work.

*C#* – объектно-ориентированный язык программирования. Разработан в 1998 — 2001 под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework.

*XAML* – это декларативный язык разметки. С точки зрения модели программирования .NET Framework язык XAML упрощает создание пользовательского интерфейса для приложения .NET Framework.

Repository – это паттерн, позволяющий абстрагироваться от конкретных подключений к источникам данных, с которыми работает программа, и являющийся промежуточным звеном между классами, непосредственно взаимодействующими с данными и остальной программой.

Unit of Work – это паттерн, который используется в случае, если репозитории используют одно и тоже же подключение.

1. **Разработка архитектуры проекта**
   1. **Постановка задачи**

Целью курсового проекта является разработка программного средства для ведения записей.

Функции приложения продемонстрированы на рисунке 2.1.

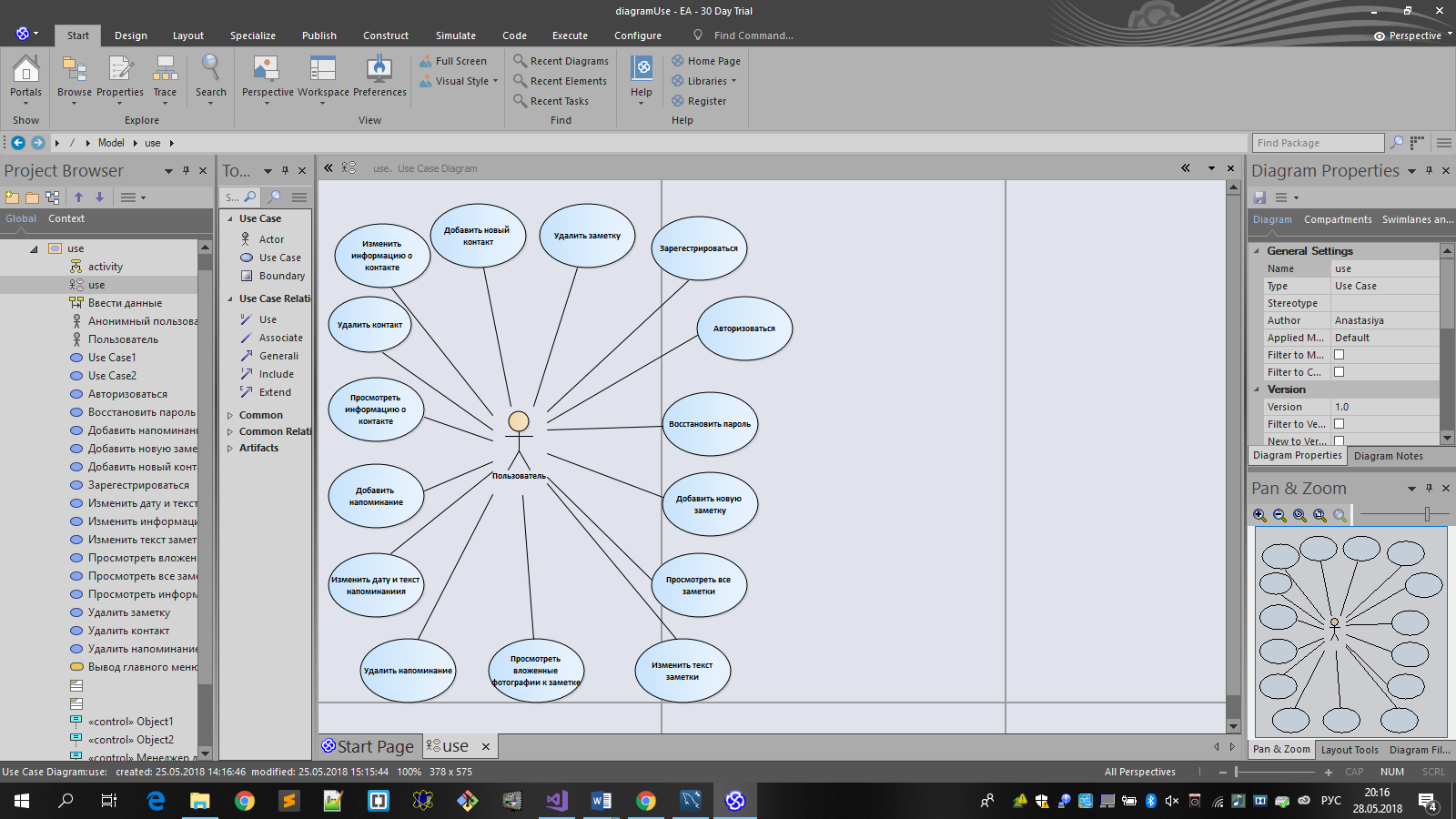


Рисунок 2.1 – Диаграмма вариантов использования

Пользователь имеет возможность зарегистрироваться, авторизоваться, восстановить пароль.

Предусмотрено создание, изменение и удаление новой заметки, контакта и напоминания.

Возможно добавление вложенных фотографий для заметки с последующим просмотром.

Диаграммы классов, деятельности и последовательности представлены в приложениях А, Б и В соответственно.

* 1. **Обобщенная структура**

Решение DIARY\_V4 представлено одним проектом DIARY\_V4, имеющим структуру, представленную на рисунке 2.2.

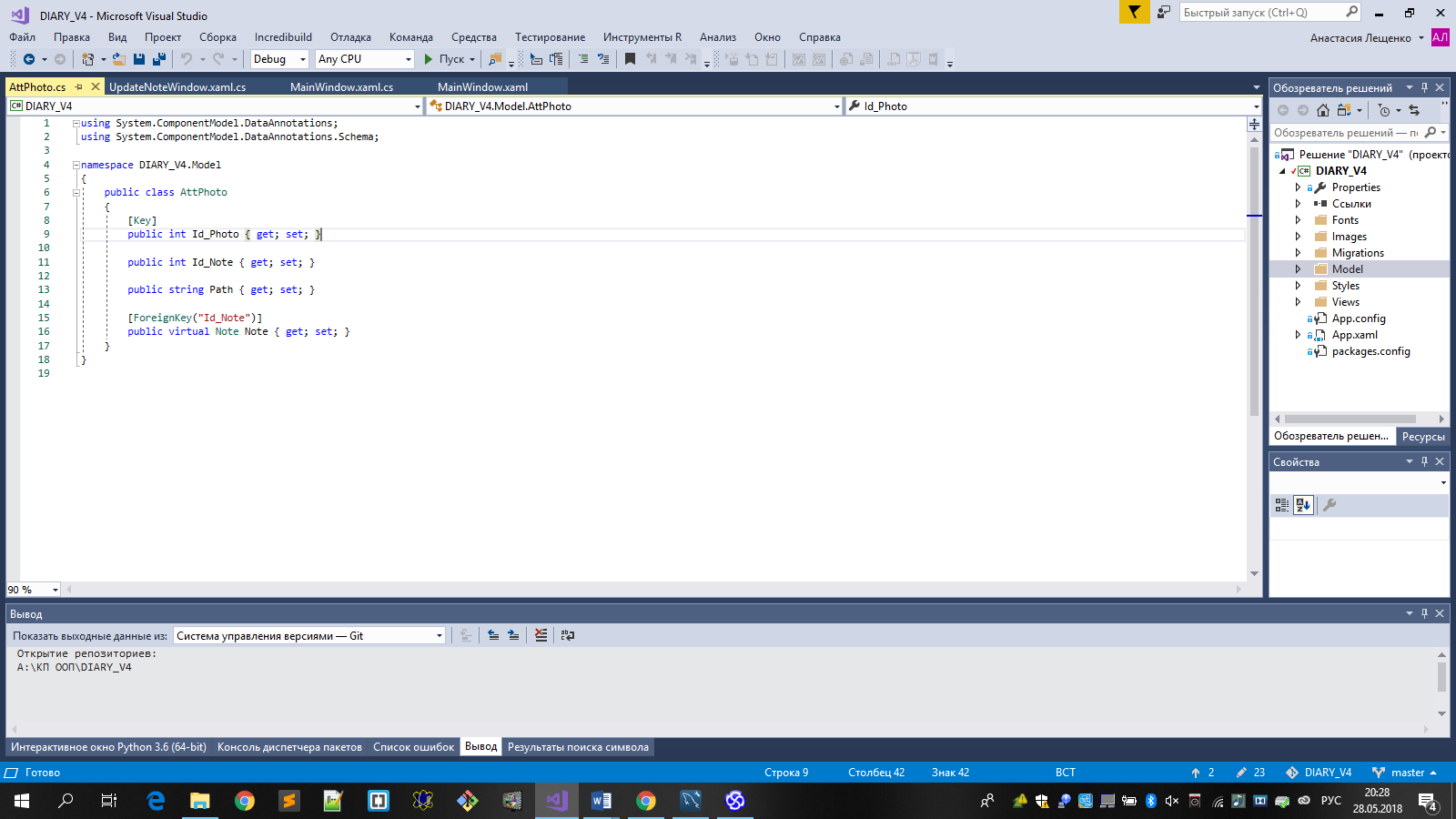


Рисунок 2.2 – Структура проекта

Разбиение по папкам служит для наилучшего представления и понимания нахождения составных частей проекта. Описание компонентов каждой папки представлено в таблице 2.1.

Таблица 2.1 – Описание компонентов каждой папки проекта

|  |  |
| --- | --- |
| Папка | Описание |
| Fonts | Папка с используемыми шрифтами |
| Images | Папка с используемыми приложениями |
| Model | Папка с классами и интерфейсами |
| Styles | Папка с файлами словарей ресурсов |
| Views | Папка с окнами |

Данное разбиение позволяет быстро найти нужный раздел и фрагмент кода.

Содержимое папки Model и Styles продемонстрировано на рисунке 2.3.

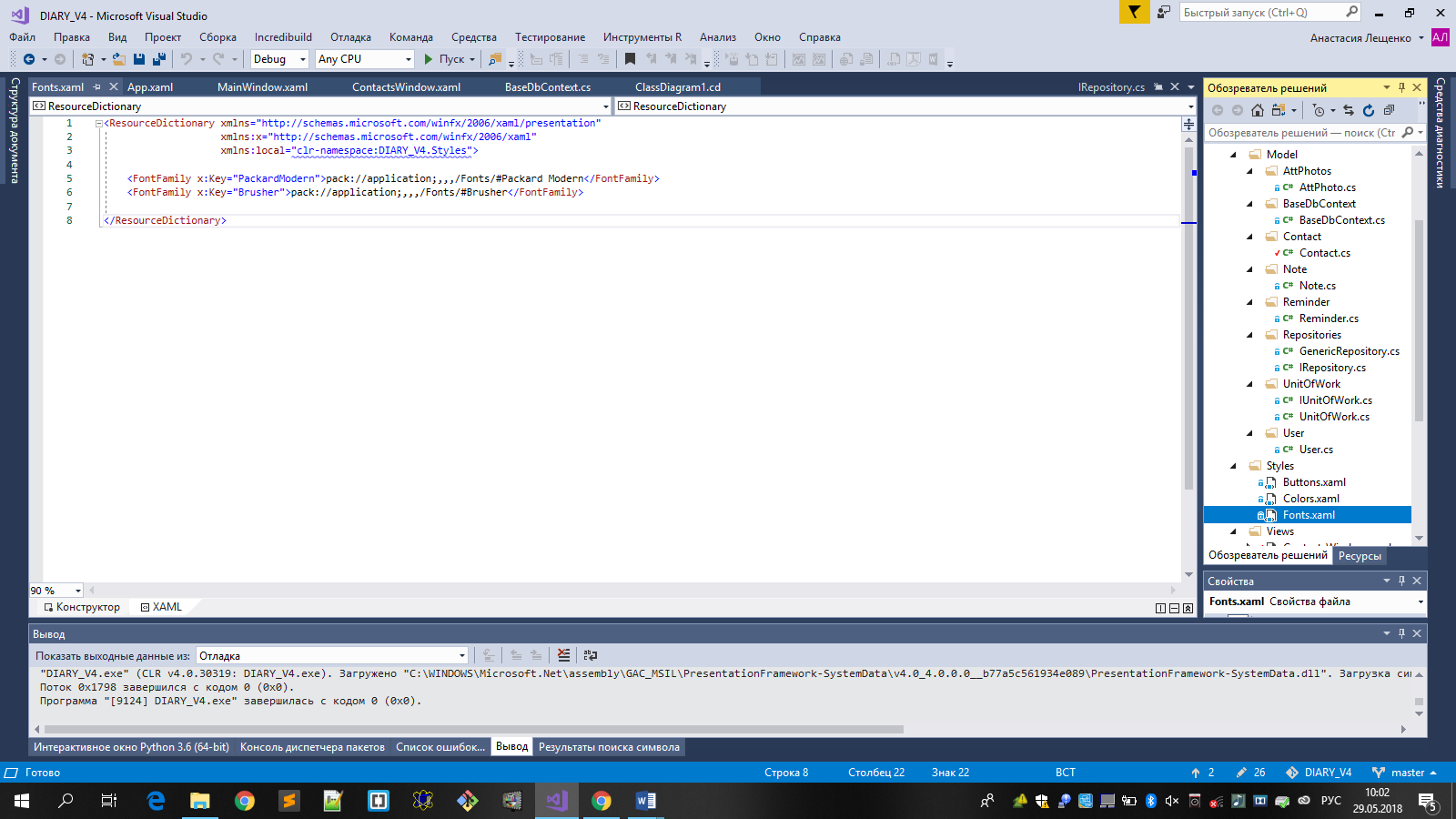


Рисунок 2.3 – Содержимое папок Model и Styles

В папке Model находятся классы и интерфейсы для работы с приложением и соединением с базой данных.

В папке Styles находятся файлы словарей ресурсов для задания стиля различных элементов управления.

В папке Views располагаются все окна приложения. Они продемонстрированы на рисунке 2.4.

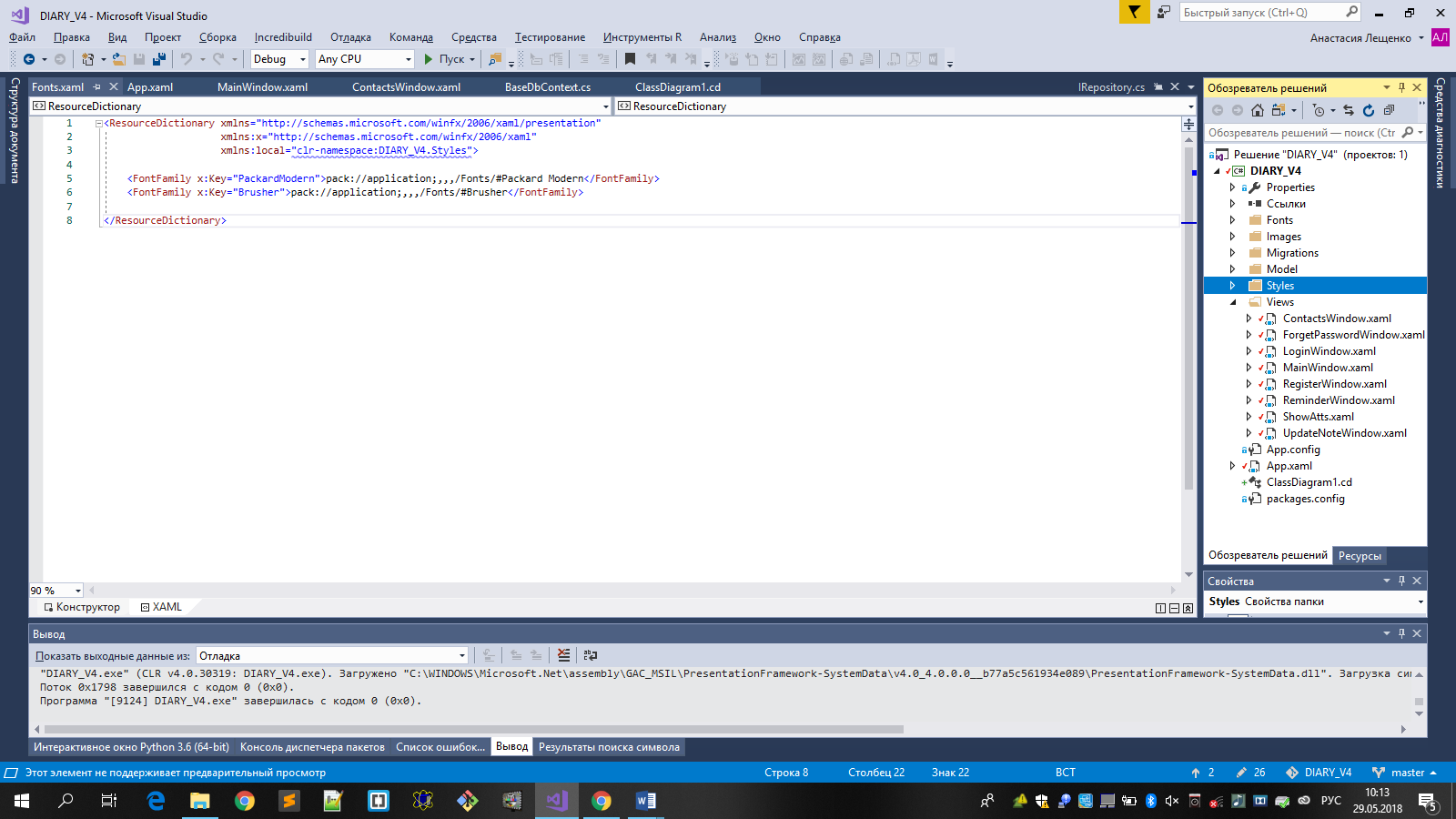


Рисунок 2.4 – Содержимое папки Views

Более подробное описание структуры папки Views находится в таблице 2.2.

Таблица 2.2 – Описание структуры папки Views

|  |  |
| --- | --- |
| Название окна | Описание |
| ContactsWindow | Окно для отображения информации о контакте |
| ForgetPasswordWindow | Окно восстановления пароля |
| LoginWindow | Окно авторизации |
| MainWindow | Главное окно |
| RegisterWindow | Окно регистрации |
| ReminderWindow | Окно напоминаний |
| ShowAtts | Окно вложенных фотографий |
| UpdateNoteWindow | Окно обновления заметки |

У каждого файла xaml есть свой файл кода, в котором находится вся логика для данного окна.

* 1. **Проектирование базы данных**

Для реализации приложения была разработана база данных DIARY. Для создания базы данных был выбран Microsoft SQL Server.

Microsoft SQL Server – система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

Преимущества Microsoft SQL Server 2012 – отсутствие необходимости в установке программы, а также совместимость с API .NET Framework (поддержка технологий ADO.NET, Entity Framework, NHibernate и др.).

Диаграмма полученной базы данных продемонстрирована на рисунке 2.5.

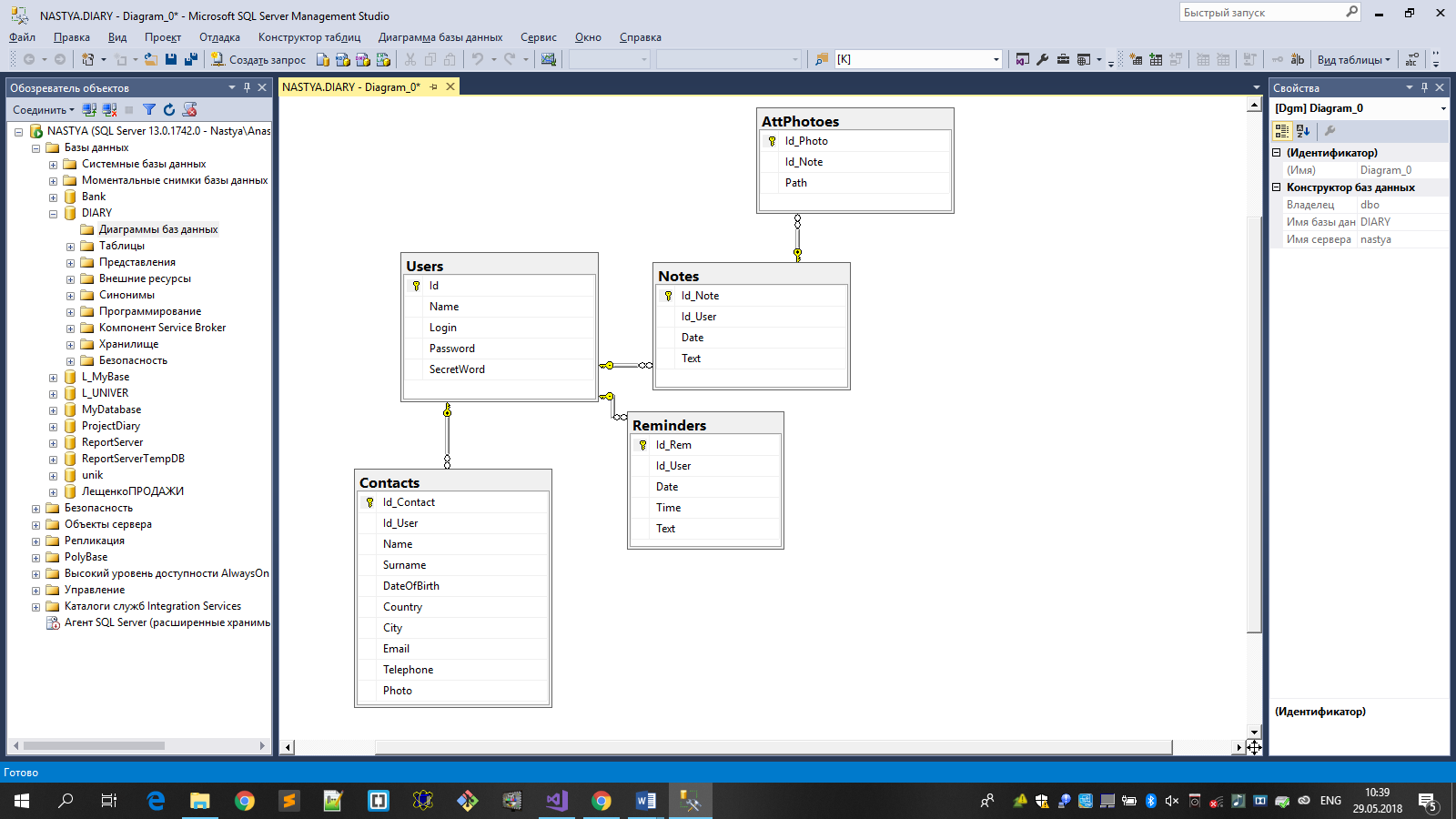


Рисунок 2.5 – Диаграмма базы данных

Всего в базе данных находится 5 таблиц. Таблица Users хранит всех зарегистрированных пользователей, Notes – заметки пользователей, Reminders – напоминания пользователей, Contacts – контакты пользователей, AttPhotoes – вложенные фотографии к заметке.

Таблица Users состоит из пяти столбцов:

* Id – идентификатор пользователя;
* Name – имя пользователя;
* Login – логин пользователя;
* Password – пароль пользователя;
* SecretWord – секретное слово пользователя.

Таблица Notes состоит из 4 столбцов

* Id\_Note – идентификатор заметки;
* Id\_User – идентификатор пользователя;
* Date – дата заметки;
* Text – текст заметки.

Таблица Contacts состоит из 10 столбцов:

* Id\_Contact – идентификатор контакта;
* Id\_User – идентификатор пользователя;
* Name – имя контакта;
* Surname – фамилия контакта;
* DateOfBirth – дату рождения контакта;
* Country – страну проживания/рождения контакта;
* City – страну проживания/рождения контакта;
* Email – email контакта;
* Telephone – мобильный телефон контакта;
* Photo – путь к фотографии контакта.

Таблица Reminders состоит из 5 столбцов:

* Id\_Rem – идентификатор напоминания;
* Id\_User – идентификатор пользователя;
* Date – дата напоминания;
* Time – время напоминания;
* Text – текст напоминания.

Таблица AttPhotoes содержит 3 столбца:

* Id\_Photo – идентификатор фотографии;
* Id\_Note – идентификатор заметки;
* Path – путь к фотографии.

1. **Руководство пользователя**

После запуска приложения открывается окно авторизации, представленное на рисунке 3.1.

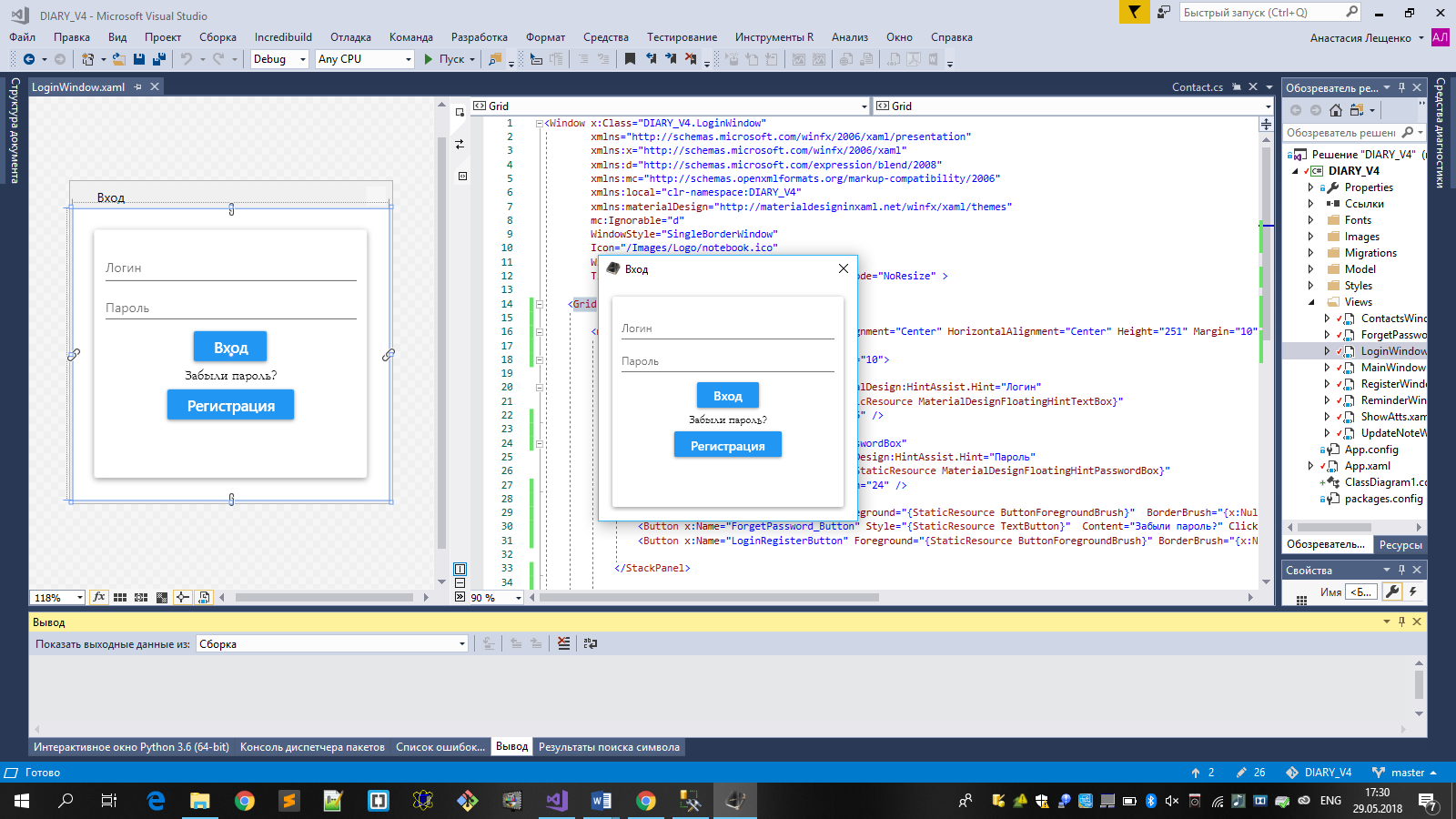


Рисунок 3.1 – Окно авторизации

Если у Вас уже существует аккаунт, то необходимо авторизоваться, введя логин и пароль, иначе следует нажать на кнопку «Регистрация», после чего появится новое окно, продемонстрированное на рисунке 3.2.

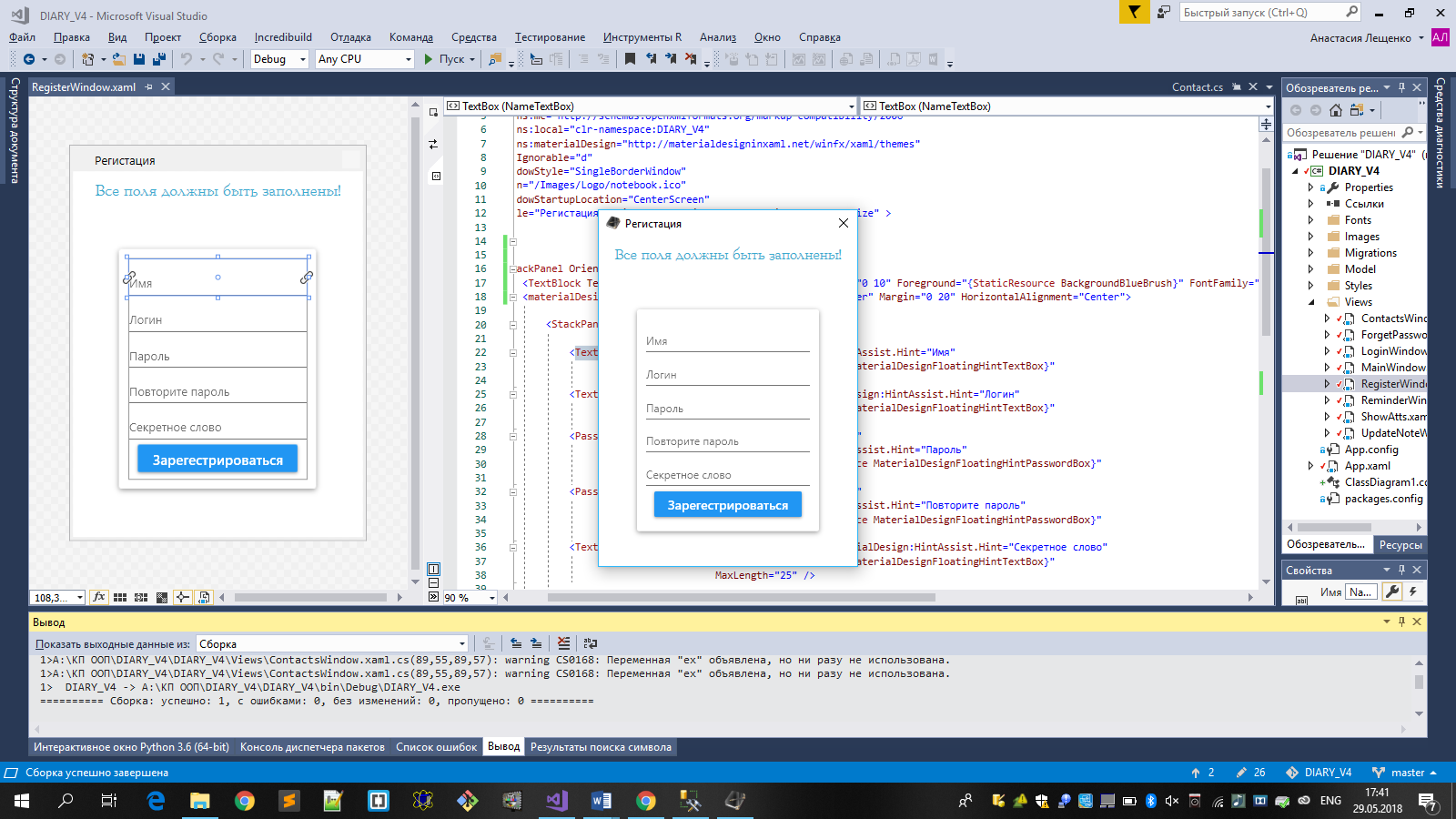


Рисунок 3.2 – Окно регистрации

Здесь необходимо заполнить информацию о себе. Обратите внимание, что все поля должны быть обязательно заполнены. После заполнения информации Вы попадаете на окно авторизации и можете либо войти в свой аккаунт, либо выйти из приложения, нажав в правом верхнем углу на знак «крестика».

В случае, если Вы забыли пароль, его можно восстановить, нажав на кнопку «Забыли пароль?». В результате откроется окно, продемонстрированное на рисунке 3.3.

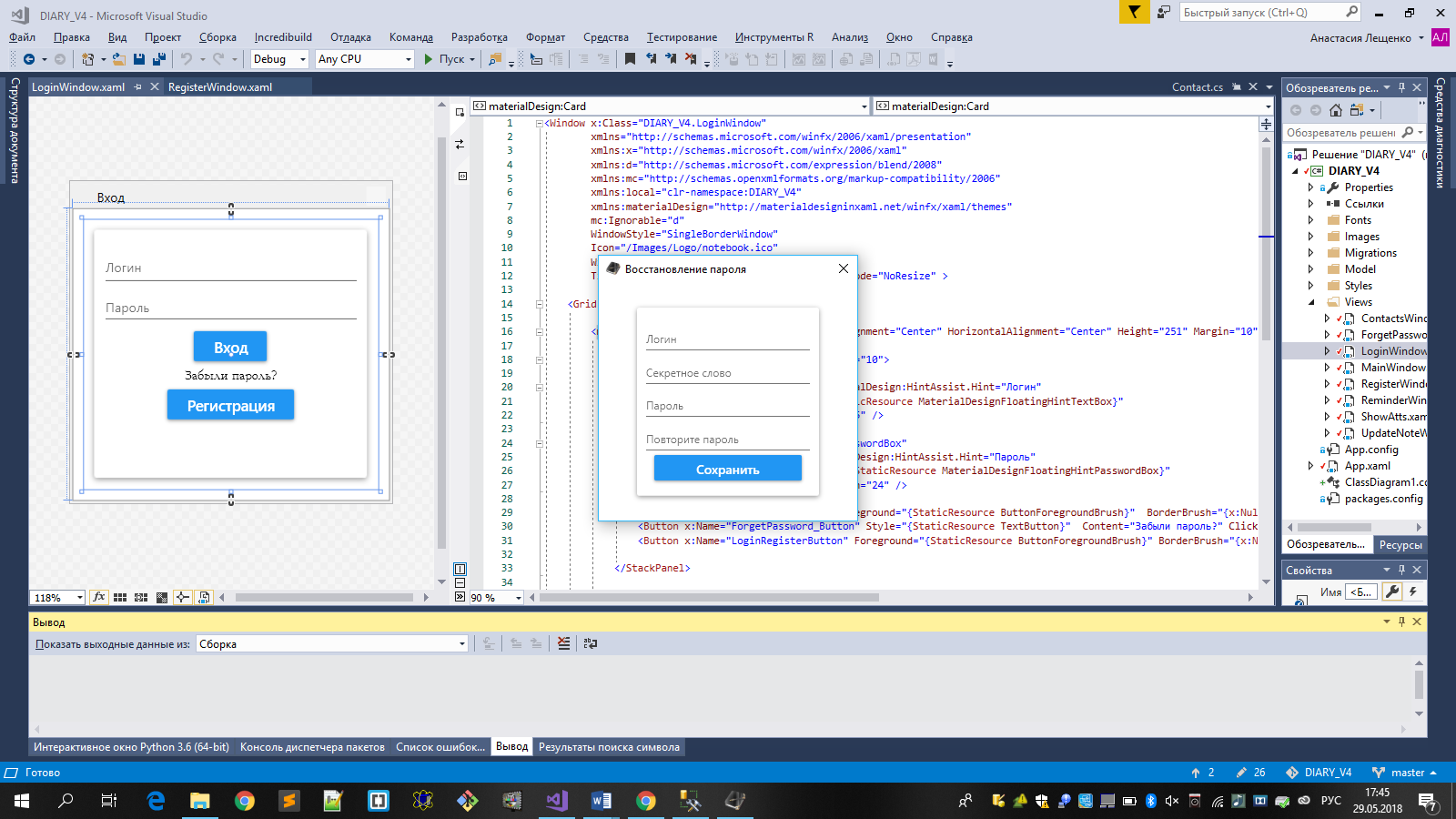


Рисунок 3.3 – Восстановление пароля

Для восстановления пароля необходимо использовать секретное слово, использованное при регистрации. В случае, если Вы забыли секретное слово, восстановление пароля невозможно. После успешного восстановления пароля появляется окно авторизации.

В случае успешной авторизации появится приветственное сообщение, представленное на рисунке 3.4.

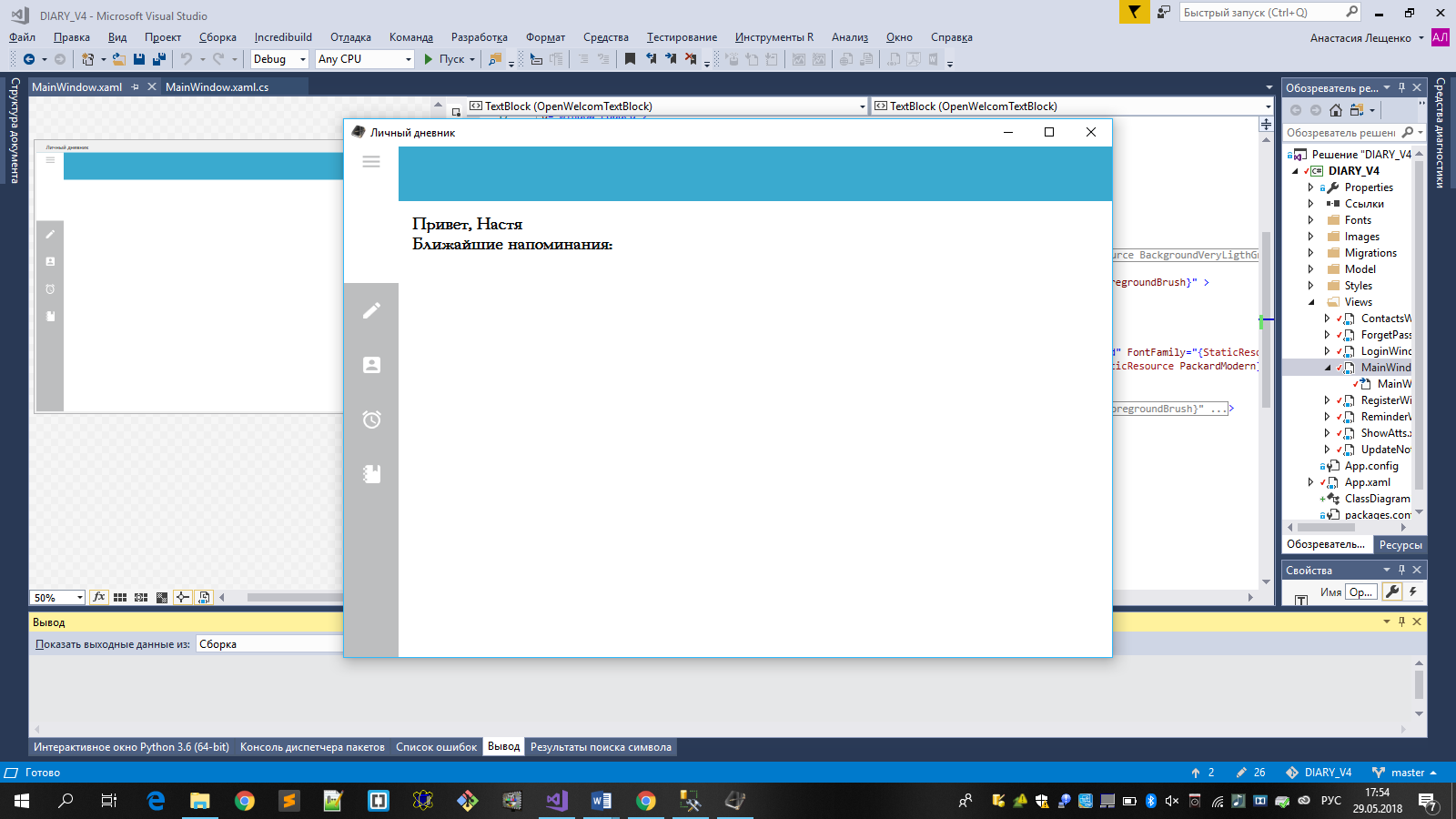


Рисунок 3.4 – Приветственное сообщение

Формат сообщения представлен следующим образом: «Привет, Ваше имя».

Ниже отображаются ближайшие три напоминания, если они есть.

В левом верхнем углу находится знак трех горизонтальных линий, нажав на который, можно просмотреть меню приложения. После нажатия он поменяется на знак стрелки, нажав на который можно свернуть панель в первоначальное состояние. Результат продемонстрирован на рисунке 3.5.



Рисунок 3.5 – Выдвигающаяся панель

Для создания новой заметки необходимо нажать на вкладку «Новая заметка», после чего появится окно, представленное на рисунке 3.6.

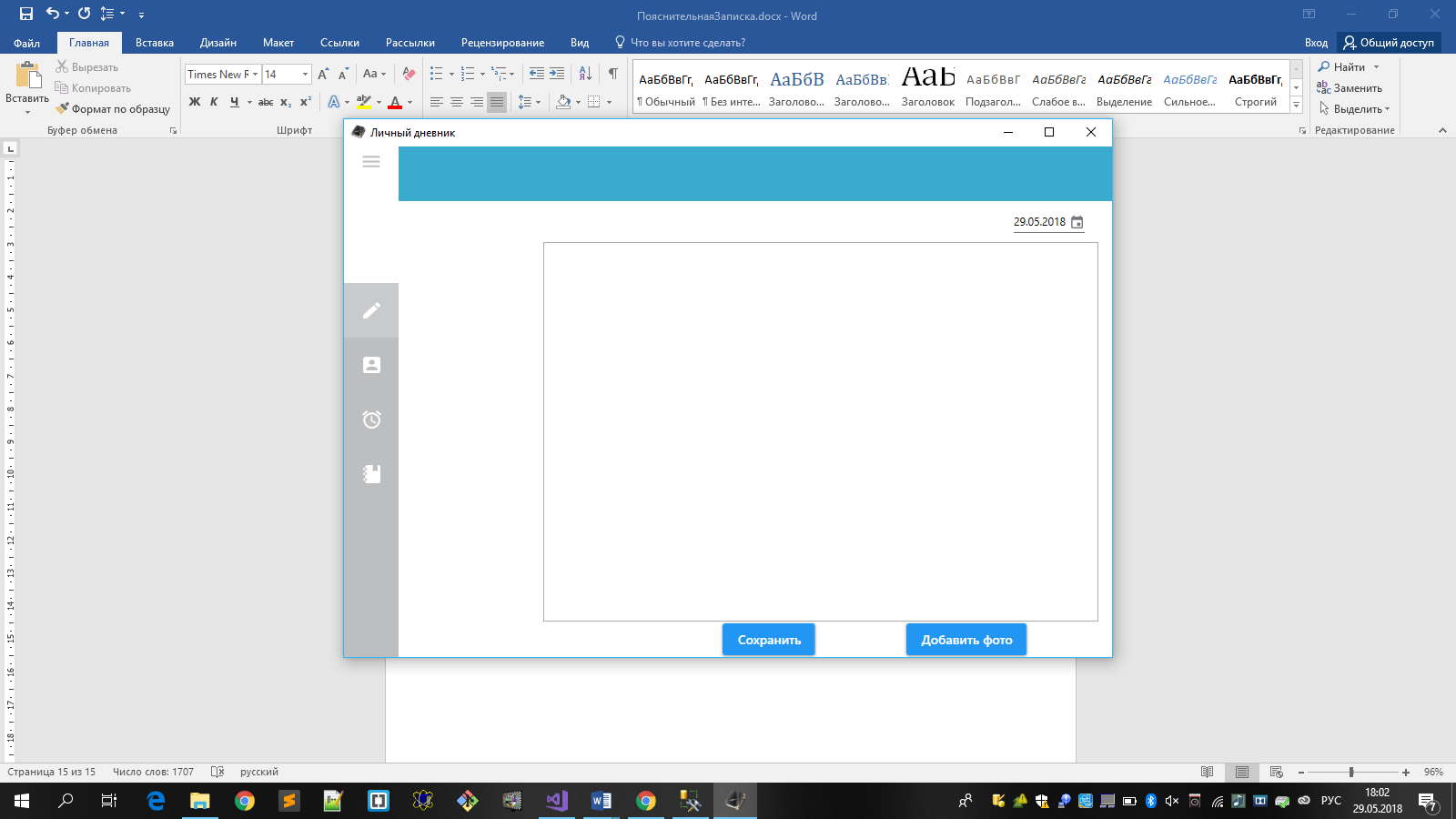


Рисунок 3.6 – Новая заметка

Выделенный прямоугольник представляет собой область, в которую можно занести текст заметки. В верхнем правом углу необходимо выбрать дату. По умолчанию она равна текущей.

Слева от прямоугольника находится область для добавления вложенных фотографий для заметки. Максимальное количество равно 3. Добавить фотографию можно с помощью кнопки «Добавить фото». Результат продемонстрирован на рисунке 3.7.

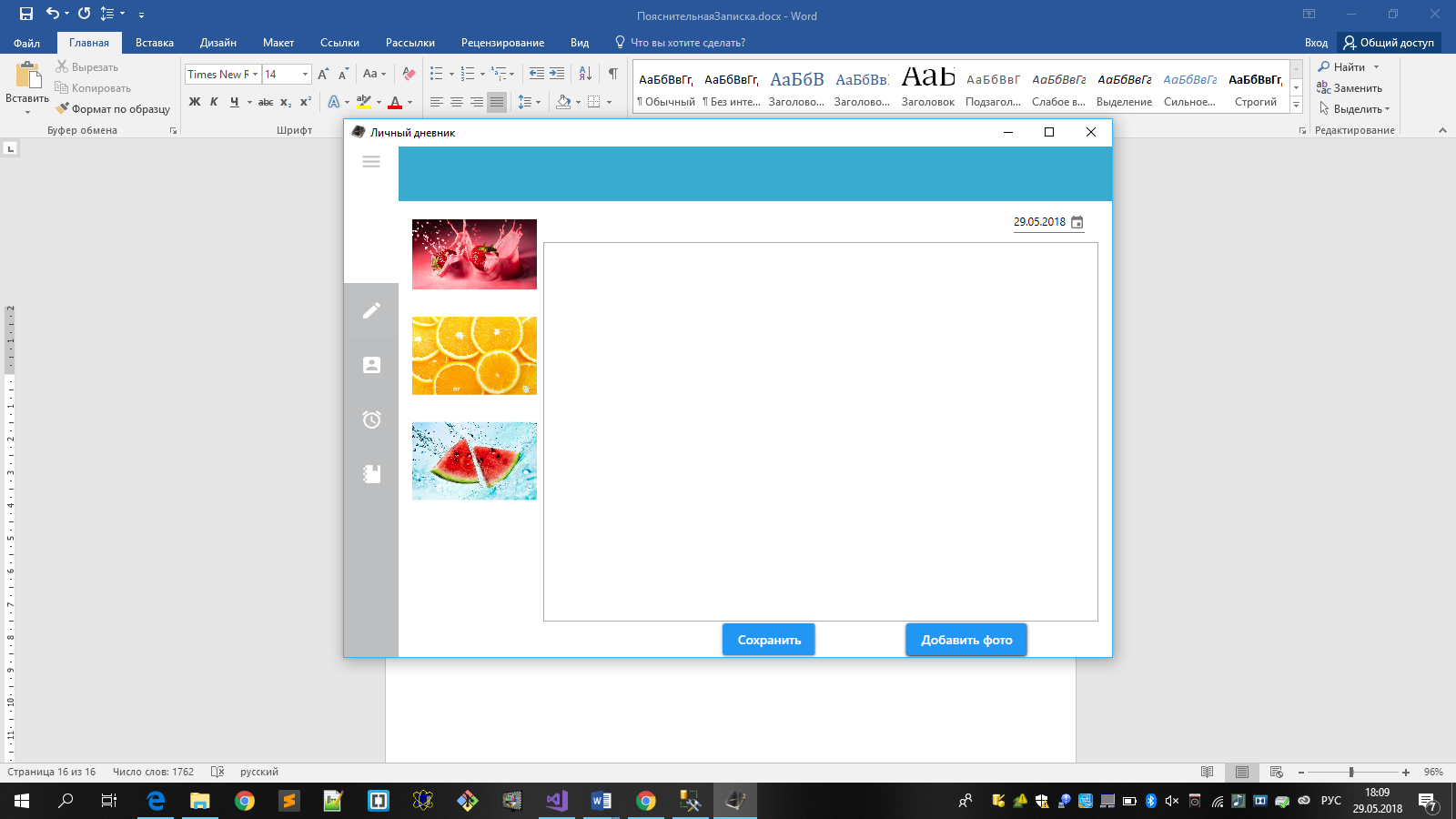


Рисунок 3.7 – Добавление фотографий

Сохранить заметку можно, нажав на кнопку «Сохранить», однако стоит учесть, что успешного выполнения данной операции, необходимо, чтобы в заметке присутствовал текст.

Если заметка для данной даты уже существует, то будет выведено предупреждающее об этом сообщение с предложением дописать текст в конец заметки.

Добавление фотографий возможно только при первом создании заметки.

Для просмотра всех заметок необходимо нажать на кнопку «Все заметки», после чего появится содержимое, представленное на рисунке 3.8.

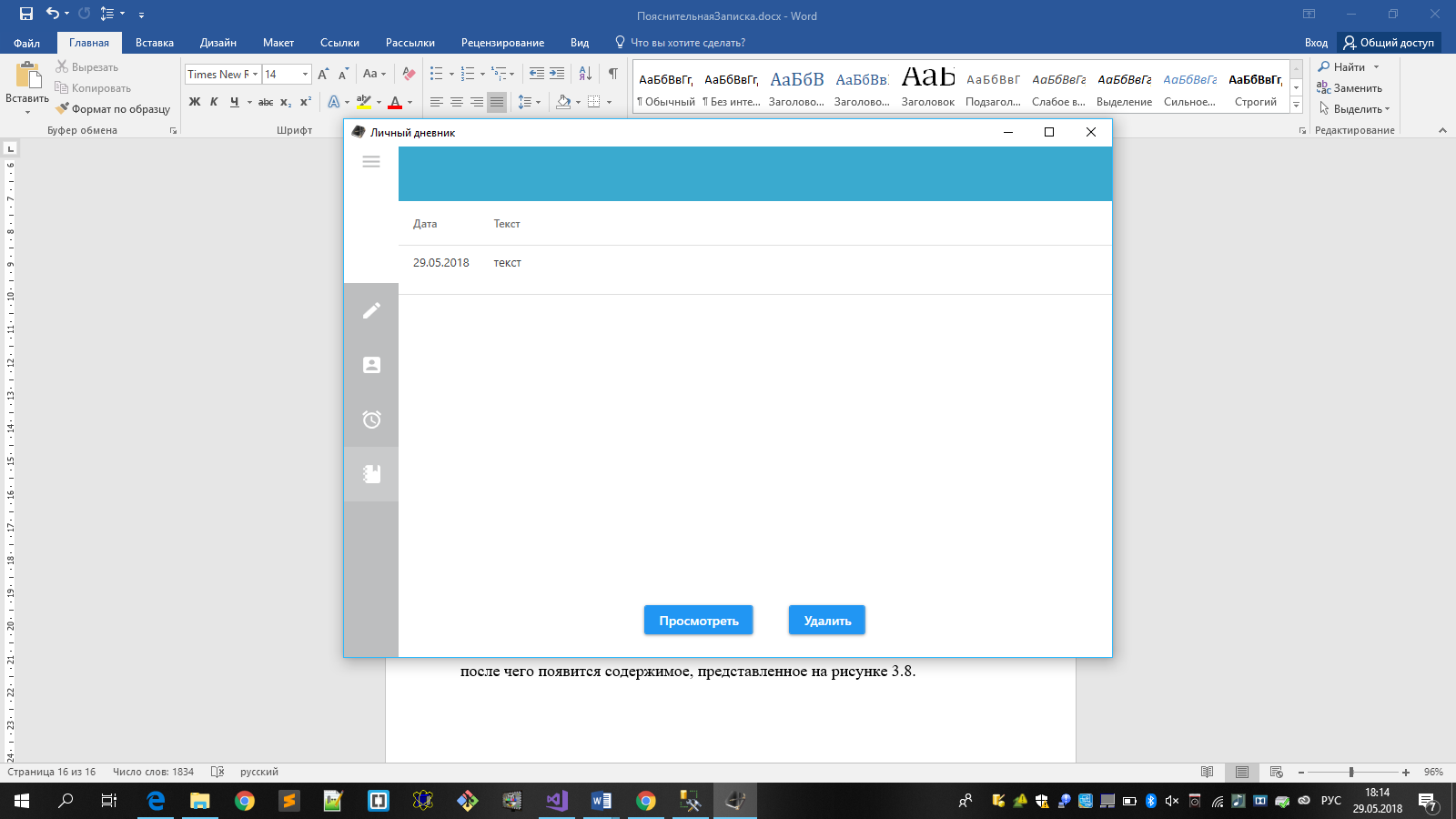


Рисунок 3.8 – Все заметки

На данном рисунке можно увидеть дату и текст заметки. Для удаления заметки необходимо выделить ее нажатием левой кнопки мыши и нажатием кнопки «Удалить». Просмотреть заметку с последующим изменением можно, выделив нужную заметку и нажав на кнопку «Просмотреть», либо кликнув по нужно строке с заметкой двойным щелчком. Результат продемонстрирован на рисунке 3.9.



Рисунок 3.9 – Просмотр записи

Для сохранения изменений необходимо нажать на кнопку «ОК», для отмены – на кнопку «Отмена».

Для просмотра вложенных фотографий, если таковые имеются, необходимо нажать на кнопку «Вложения». Результат продемонстрирован на рисунке 3.10.

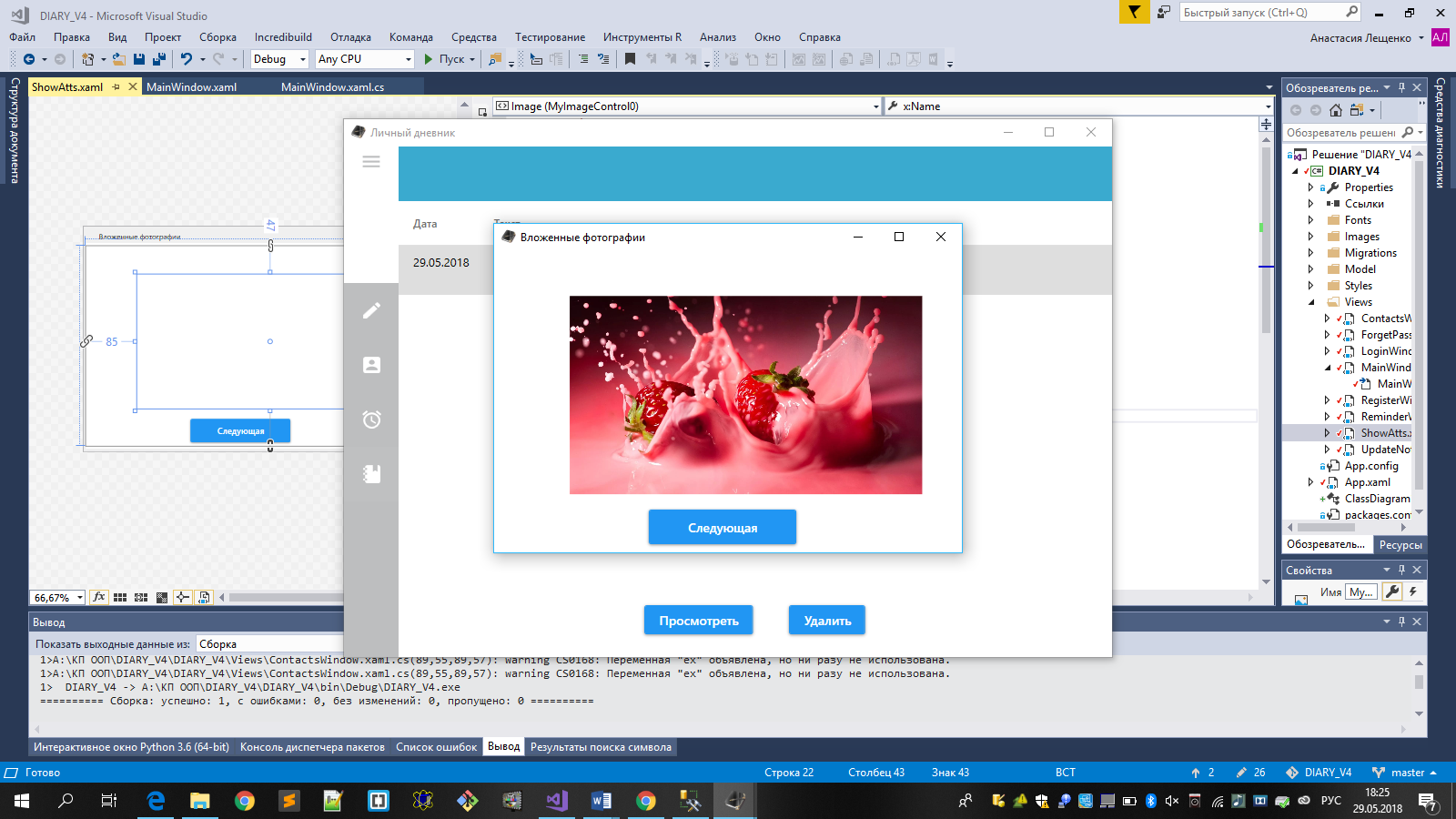


Рисунок 3.10 – Вложенные фотографии

Для просмотра следующей картинки необходимо нажать на кнопку «Следующая».

При нажатии на вкладку «Контакты» Вы увидите окно, продемонстрированное на рисунке 3.11.

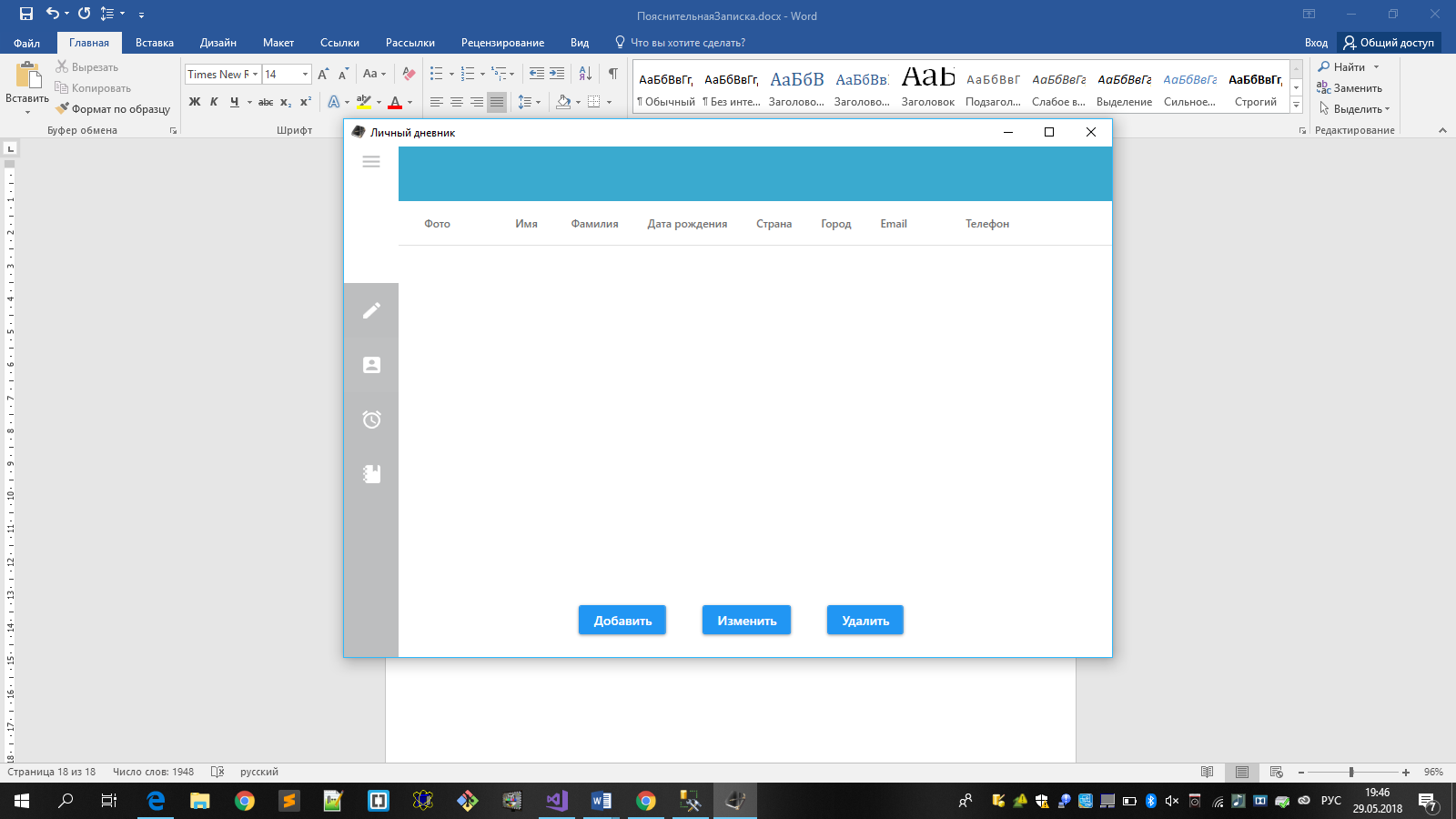


Рисунок 3.11 – Контакты

Для добавления нового контакта необходимо нажать на кнопку «Добавить», после чего появится новое окно. Результат продемонстрирован на рисунке 3.12.

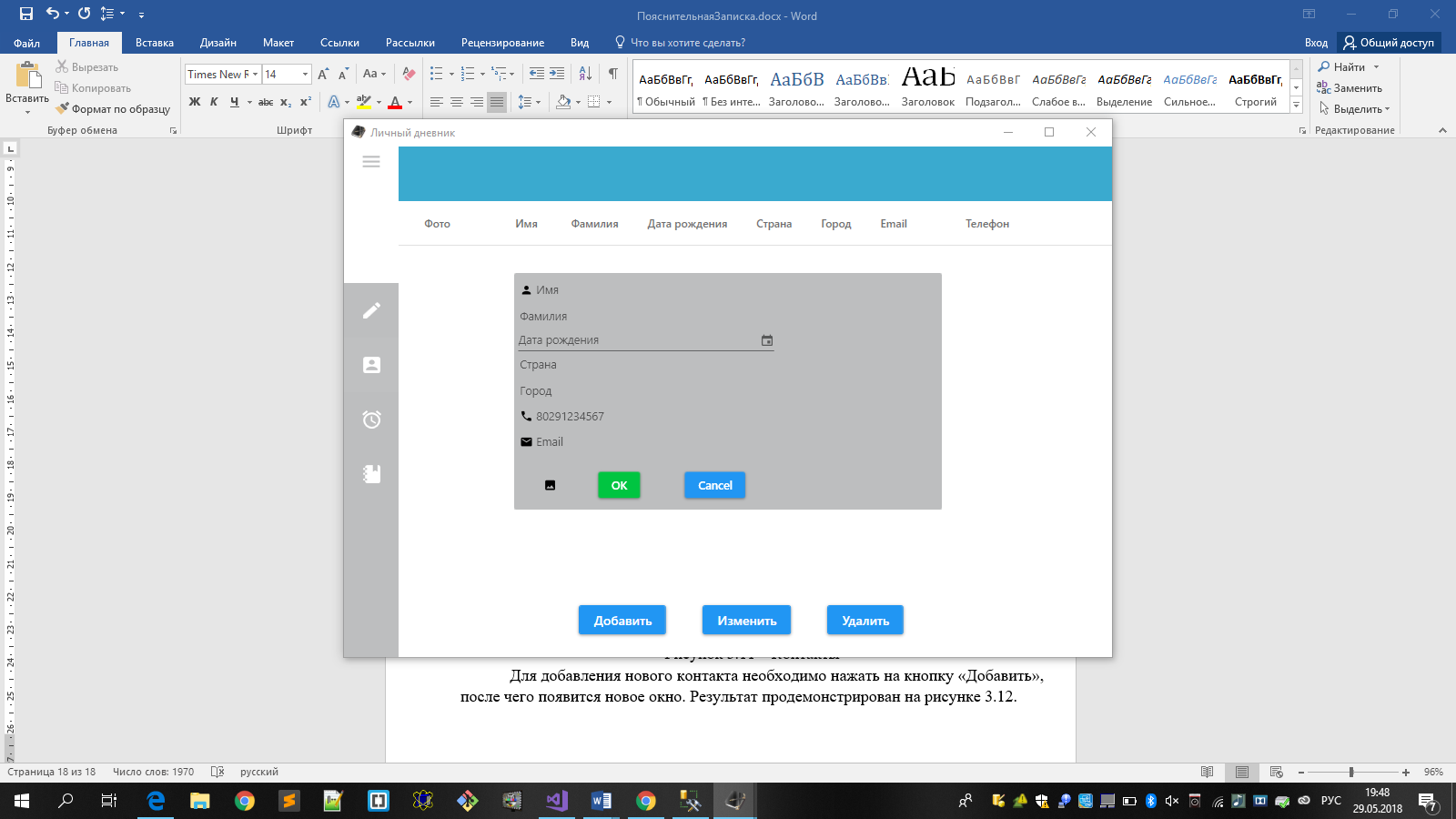


Рисунок 3.12 – Добавление контакта

Необходимо заполнить информацию о контакте. Должно быть заполнено хотя бы одно поле. Иначе добавления не произойдет. После заполнения нужных полей необходимо нажать на кнопку «Сохранить». Результат добавления виден на рисунке 3.13.

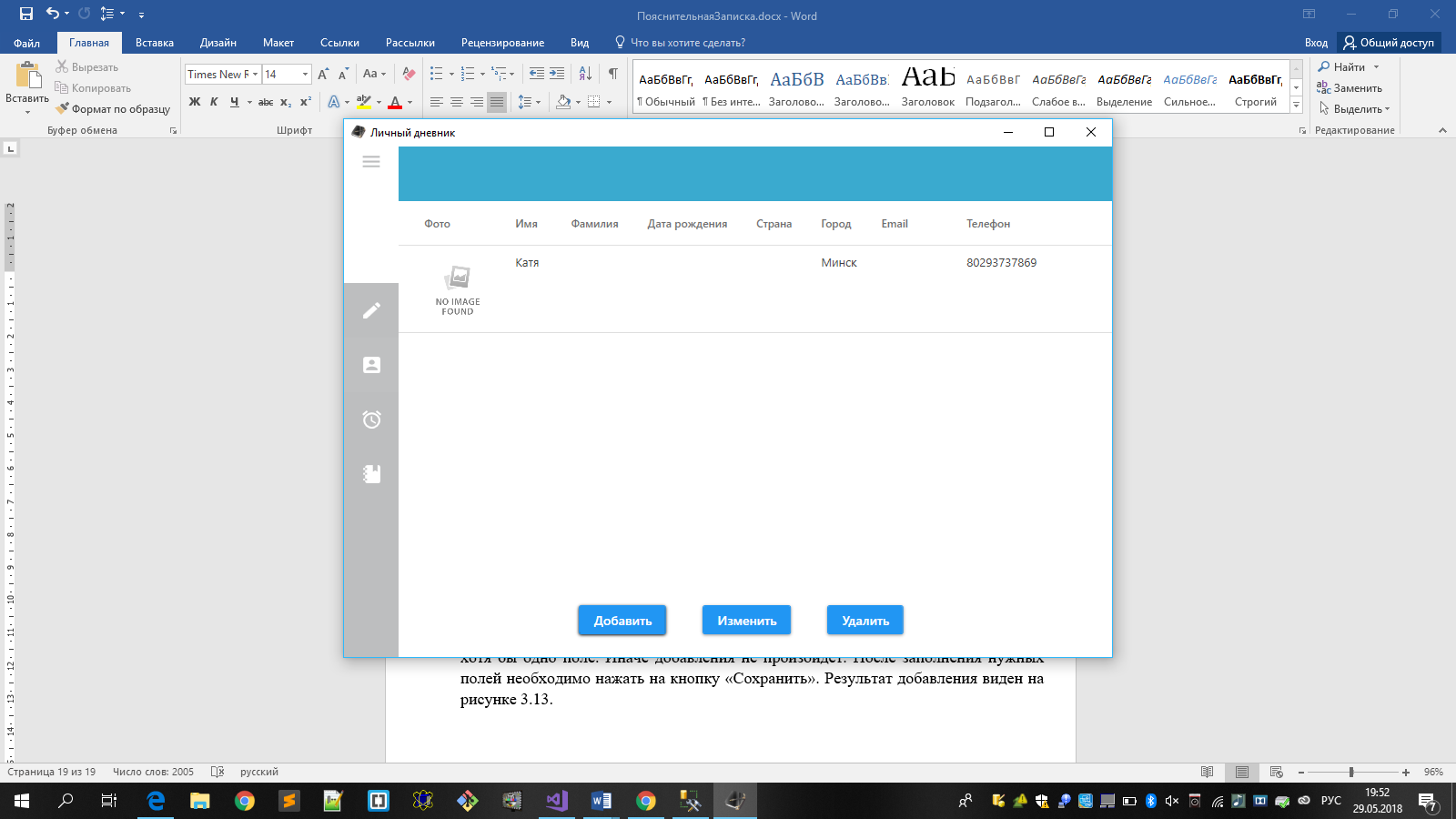


Рисунок 3.13 – Результат добавления контакта

Как видно из рисунка 3.13, если не добавить фотографию, то будет выведена фотография по умолчанию.

Для изменения или удаления контакта необходимо выделить запись и нажать на кнопку «Изменить» или «Удалить соответственно». Также для изменения возможно кликнуть по нужно ряду двойным щелчком мыши.

При нажатии на вкладку «Напоминания» будет выведено следующее окно, продемонстрированное на рисунке 3.14.

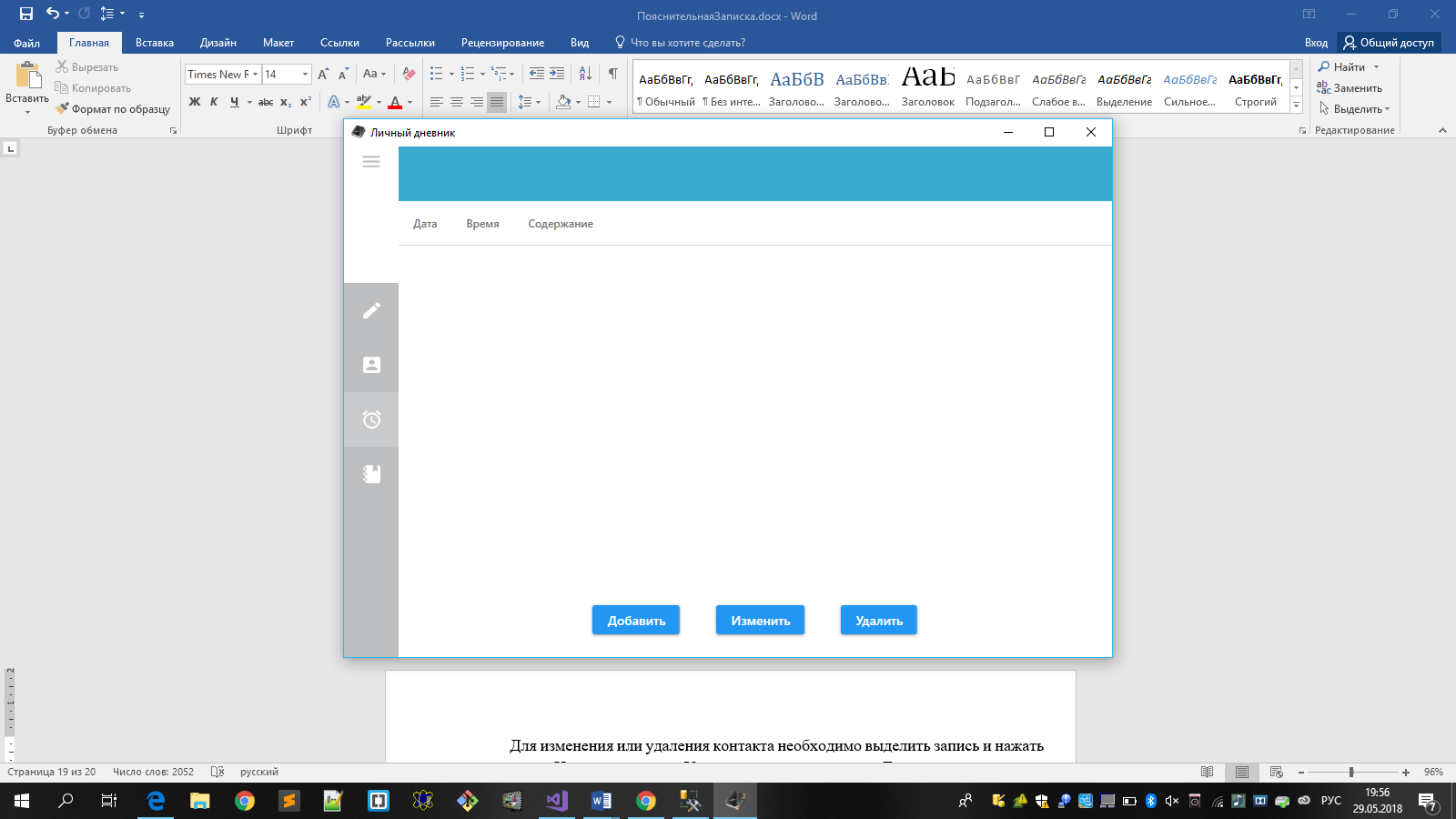


Рисунок 3.14 – Напоминания

Для добавления напоминания необходимо нажать на кнопку «Добавить», после чего появится окно, продемонстрированное на рисунке 3.15.

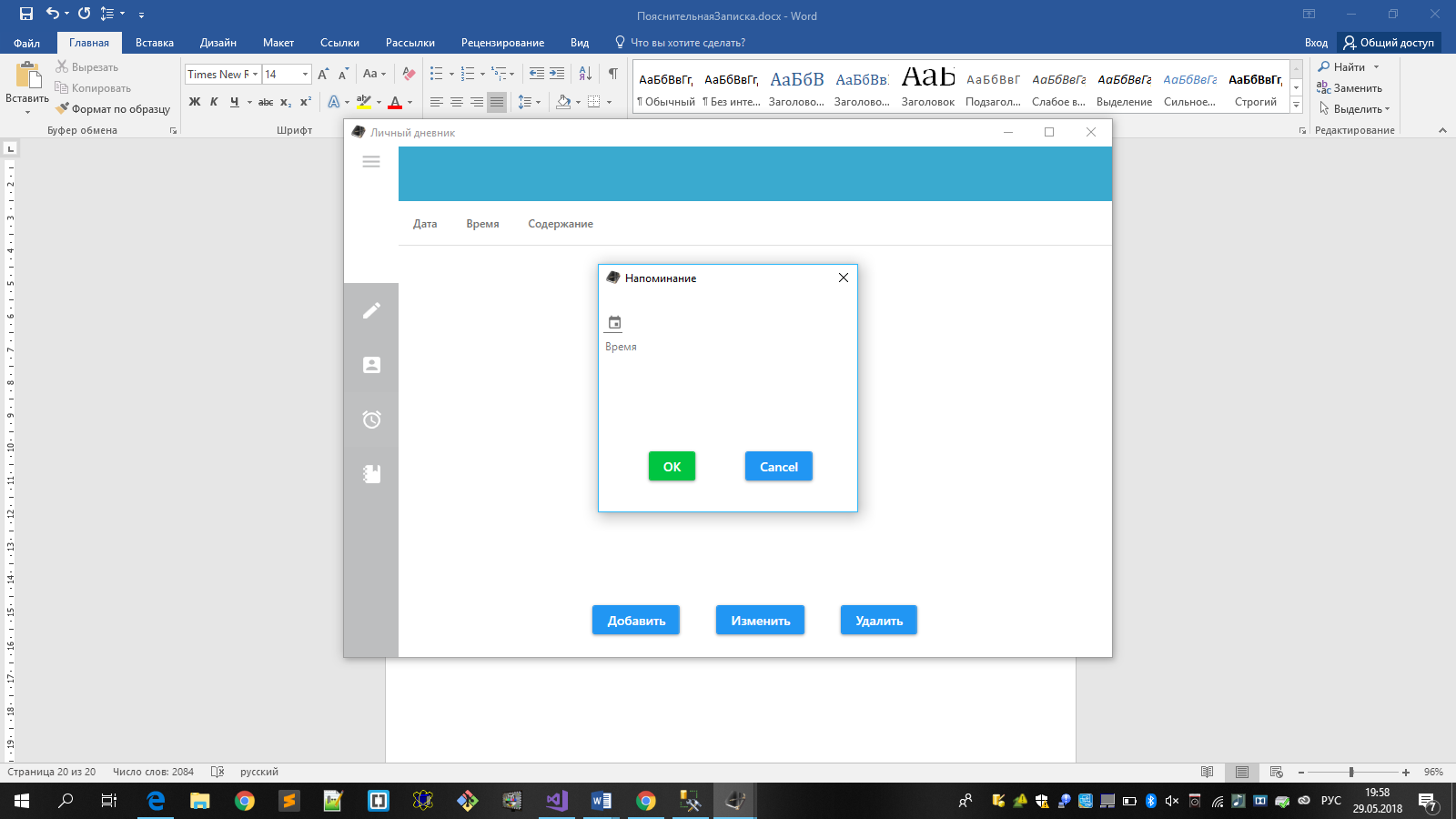


Рисунок 3.15 – Добавить напоминание

Здесь необходимо выбрать день, время и текст напоминания. После нажатия на кнопку «ОК» напоминание появится в списке напоминаний, как показано на рисунке 3.16.



Рисунок 3.16 – Список напоминаний

Для изменения напоминания необходимо кликнуть по нужной записи или выделить и нажать кнопку «Изменить». Для удаления используется кнопка «Удалить».

1. **Тестирование**

Данное программное средство использует подключение к базе данных, следовательно, неправильно введенные данные или их отсутствие может повлечь к ошибкам. В связи с этим в приложении практически в каждом методе реализован отлов и обработка исключений.

На рисунке 4.1 представлен результат обработки исключения при отсутствии данных в каком-либо поле. Высвечивается подсказка для пользователя.

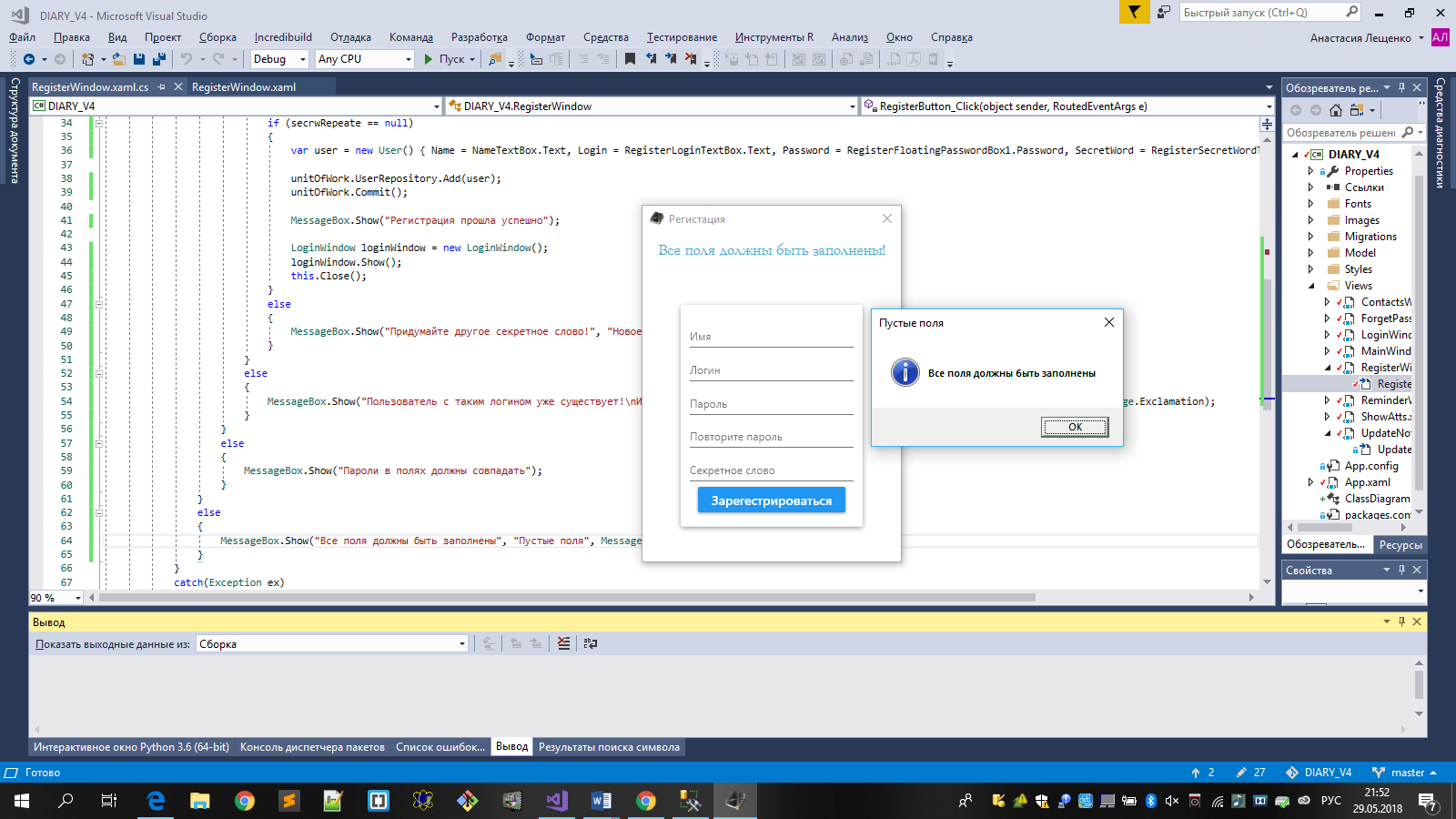


Рисунок 4.1 – Пустые поля

Также возникает ошибка при совпадении логина с существующим в БД. Это продемонстрировано на рисунке 4.2.

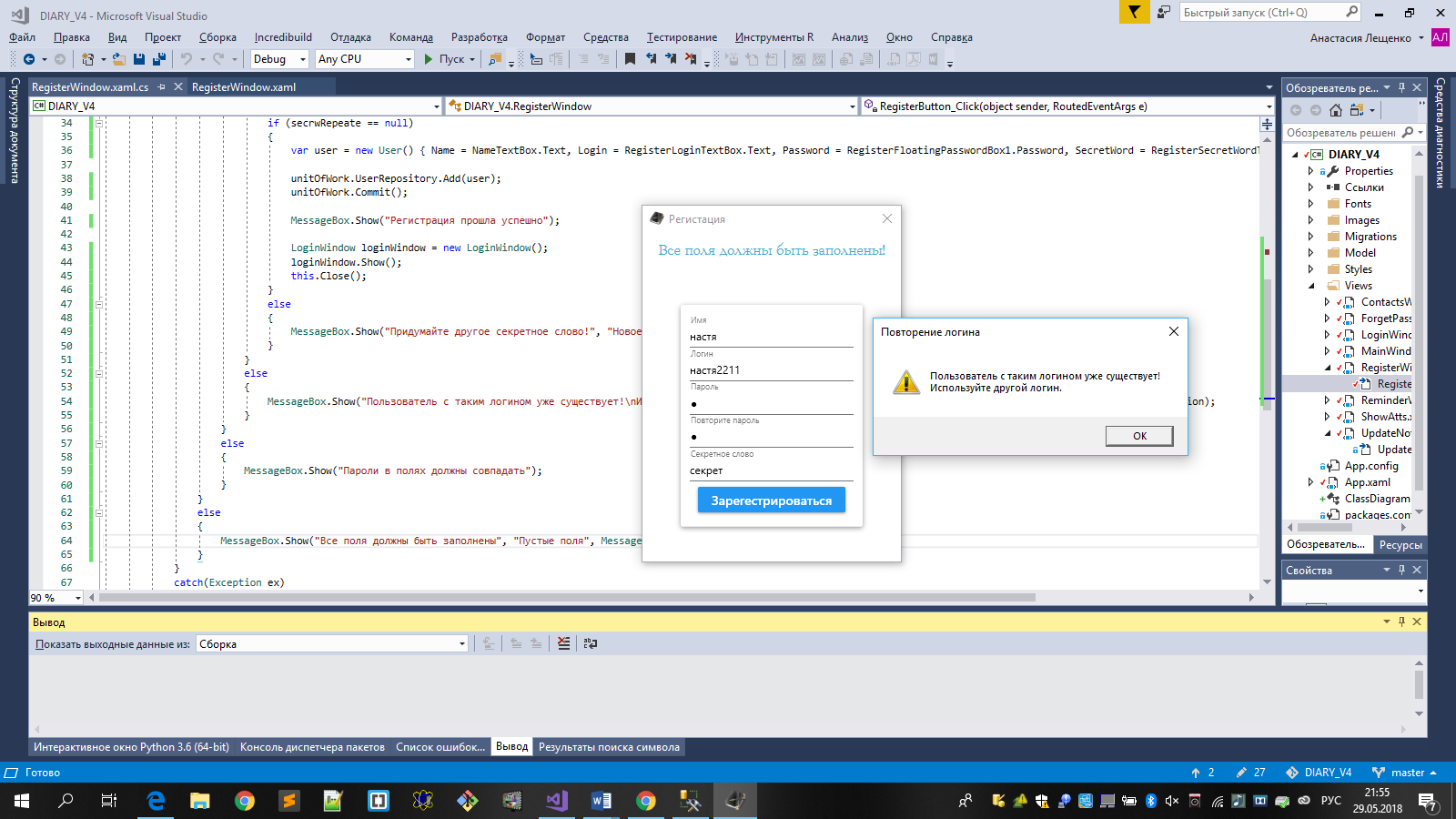


Рисунок 4.2 – Совпадение логина

На рисунке 4.3 продемонстрирована обработка ошибки при неправильном вводе логина или пароля.



Рисунок 4.3 – Неверный логин или пароль

На рисунке 4.4 представлена обработка ошибки при удалении не выделенной заметки.

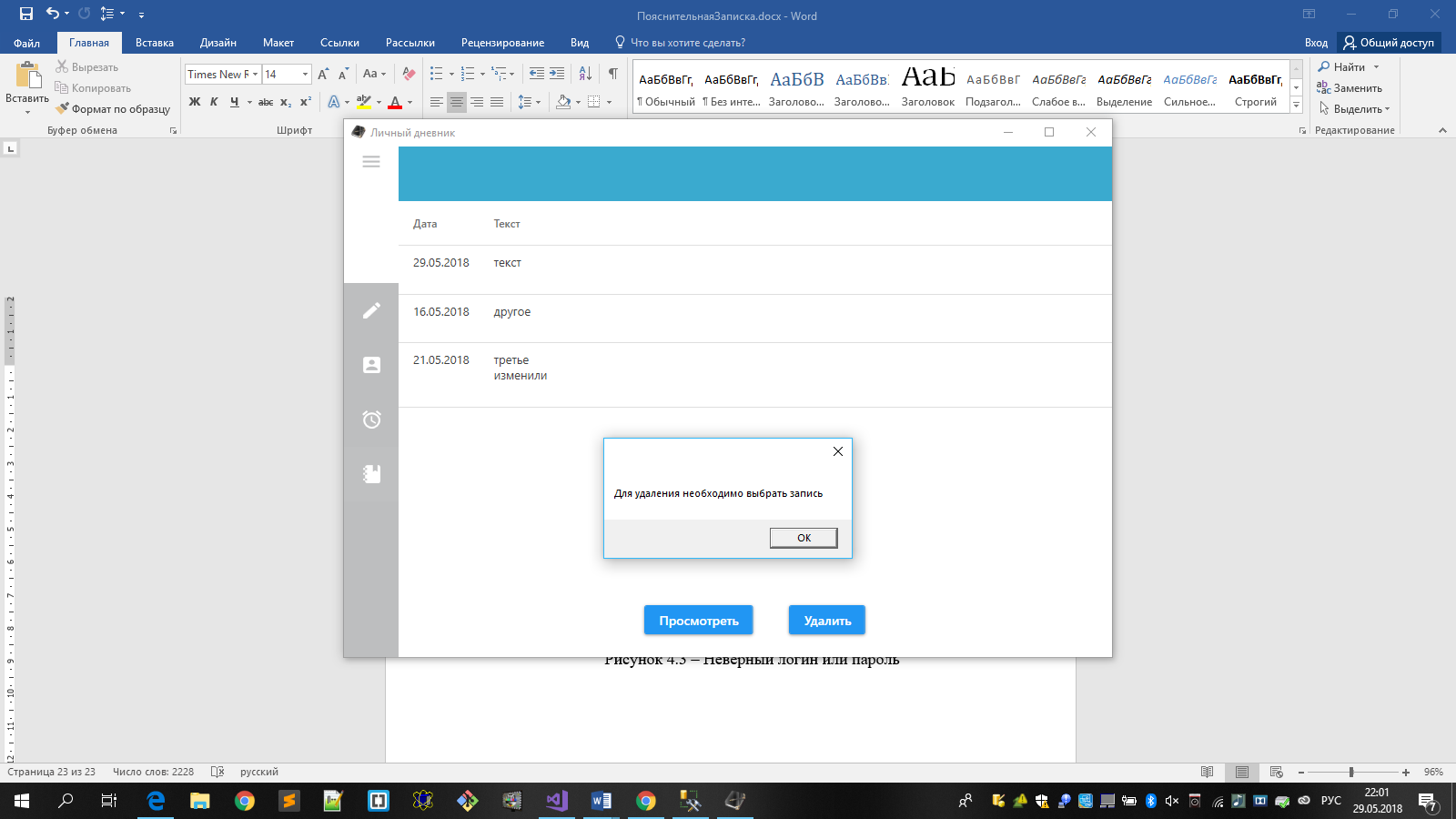


Рисунок 4.4 – Удаление невыделенной заметки

В данной главе были просмотрены основные типы обрабатываемых ошибок.

**Заключение**

В курсовом проекте описана и реализована программа, написанная на языке C#, с использованием технологий Entity Framework, WPF.

Данное программное средство имеет удобный и понятный интерфейс, реагирует на ошибочный ввод данных (например, проверка на пустоту вводимых значений).

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

Данный курсовой проект позволил закрепить полученные знания на практике, к тому же он может быть расширен в дальнейшем.

**Список используемых источников**

[1] Пацей, Н.В. Курс лекций по языку программирования С# / Н.В. Пацей. – Минск: БГТУ, 2016. – 175 с.

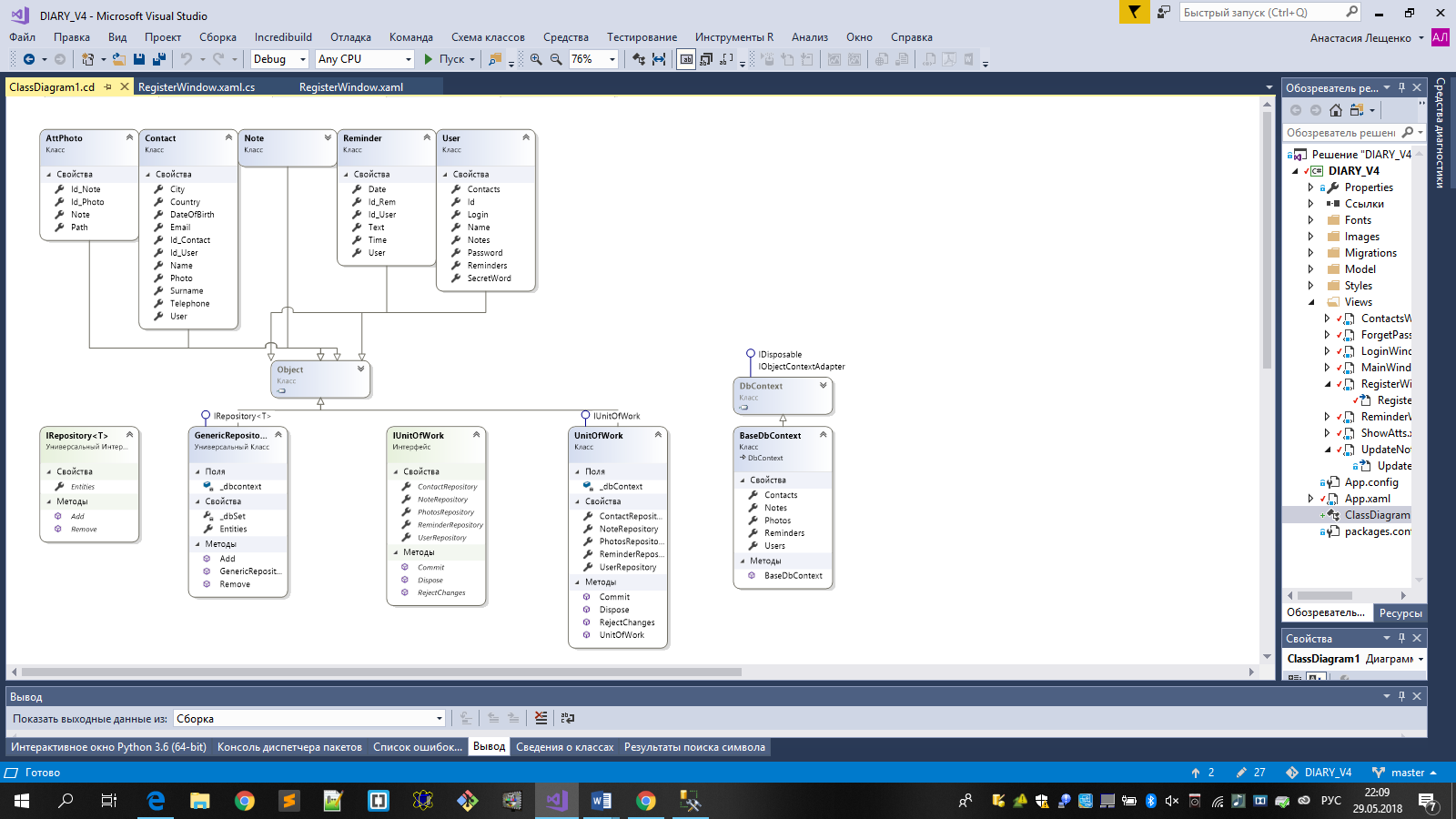
[2] Пацей, Н.В. Технология разработки программного обеспечения / Н.В. Пацей. – Минск: БГТУ, 2016. – 129 с.

[3] MSDN сеть разработчиков в Microsoft [Электронный ресурс] / Режим доступа: <http://msdn.microsoft.com/library/rus/>. Дата доступа: 20.04.2018

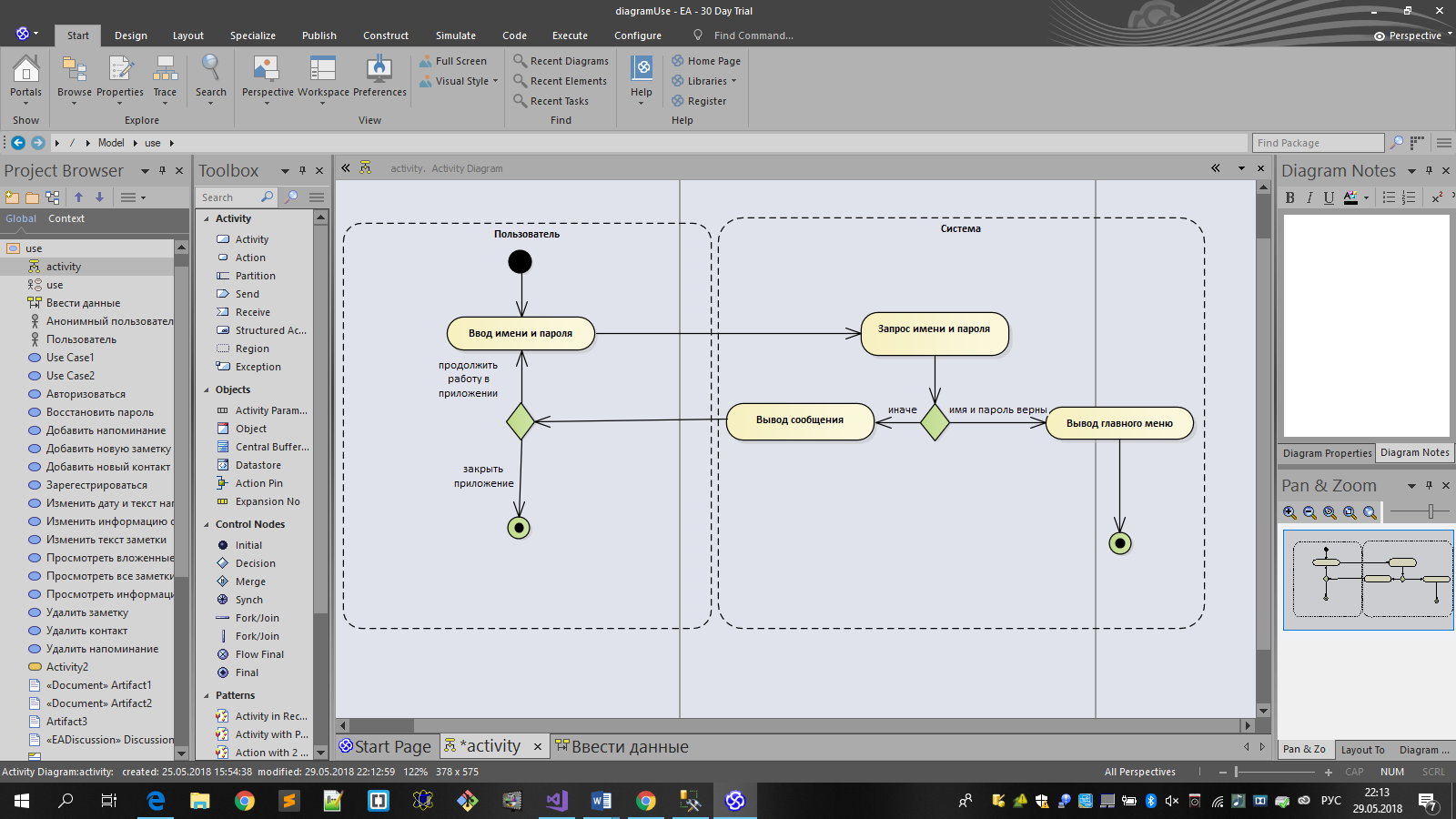
[4] METANIT.COM Сайт о программировании [Электронный ресурс] / Режим доступа: <https://metanit.com>. Дата доступа: 20.05.2018

[5] ProfessorWeb .NET & Web Programming [Электронный ресурс] / Режим доступа: <https://professorweb.ru> Дата доступа: 13.04.2018

**Приложение А**



**Приложение Б**



**Приложение В**

