МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования

Тема «Программное средство «Diabeta.net» »

Исполнитель

студент 2 курса 6 группы Трошко Валерия Николаевна

(Ф.И.О.)

Руководитель работы Радиванович Д. А.

(подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

(подпись)

Минск 2021

# Содержание

|  |  |
| --- | --- |
| Введение | 4 |
| 1. Обзор литературы и аналогов | 5 |
| * 1. Everyone Piano | 5 |
| * 1. Dream Computer Piano 2.0 | 5 |
| * 1. PC 73 Virtual Piano Keyboard | 6 |
| * 1. Аналогичность и различие программных средств | 7 |
| 1. Анализ требований к программному средству и разработка функциональных требований | 8 |
| * 1. Режим свободной игры | 8 |
| * 1. Режим обучения | 9 |
| 1. Проектирование и реализация программного средства | 10 |
| * 1. Архитектура проектируемого программного средства | 10 |
| * 1. Разработка базы данных | 10 |
| * 1. Общая структура проекта | 12 |
| * 1. Структура классов проекта | 13 |
| 1. Описание интерфейсов программного средства и его тестирование | 15 |
| * 1. Описание интерфейсов программного средства | 15 |
| * 1. Тестирование программного средства | 20 |
| 1. Руководство по установке и использованию | 22 |
| * 1. Установка программного средства | 22 |
| * 1. Руководство пользователя по работе с программным средством | 23 |
| Заключение | 28 |
| Список использованной литературы | 29 |
| Приложение 1. Листинг скриптов | 30 |

# Введение

В современном мире информационные технологии развиваются с невероятной скоростью. Так, за относительно небольшой промежуток времени истории человечества они стали играть немаловажную роль во всех сферах жизни общества. С развитием технологий человеку становятся доступны все новые и новые возможности, о которых ранее приходилось только мечтать.

Стоит отметить, среди трендов последних лет очень четко выделяется направление самоорганизации. Приложения для повышения продуктивности занимают все более высокие позиции в рейтингах по количеству скачиваний в PlayMarket и AppStore. Тренд очевиден: современный человек сегодня – это человек, стремящийся контролировать и анализировать свою жизнь. И, безусловно, такой человек не откажется от удобного приложения, которое позволит контролировать дневной рацион питания и норму калорий. В связи с этим была выбрана главная цель курсового проекта: разработка программного средства, которое сочетало бы в себе возможности современных компьютеров и ноутбуков для организации обучения пользователя игре на фортепиано в развлекательной форме. Определяющим фактором выбора инструмента для разработки программного средства стала огромная любовь автора данного курсового проекта к игре на фортепиано, что и подтолкнуло его к созданию простого и современного приложения.

Актуальность данной темы заключается в том, что многие люди заинтересованы в том, чтобы научиться играть несложные мелодии на фортепиано, однако не у всех есть возможность приобретения дорогостоящих музыкальных инструментов или посещения преподавателей. Исходя из этого при разработке программного средства были поставлены следующие задачи:

* разработка десктопного приложения, позволяющее использовать компьютерную клавиатуру в качестве клавиатуры фортепиано;
* разработка удобного и приятного пользовательского интерфейса;
* создание уроков, которые дадут пользователю начальные навыки игры на музыкальном инструменте;
* разработка функций, позволяющих отслеживать прогресс пользователя, а также воспроизводить и сохранять музыкальные фрагменты, сыгранные пользователем.

Программное средство рассчитано в первую очередь на тех, у кого есть большое желание научиться играть на фортепиано, однако нет каких-либо знаний в этой области. Но также приложение подойдёт и для профессионалов, которые смогут обучиться новым произведениям или поделиться своим творчеством с другими пользователями.

# Обзор литературы и аналогов

При изучении программ-прототипов было выявлено несколько схожих по функциональности программ. Все программные средства представлены в качестве приложений для поддержания здоровья. Результат анализа представлен ниже.

## Glucose Buddy Diabetes Tracker

Данное приложение было разработано под мобильные операционные системы IOS и Android. Имеет следующие возможности:

* + Легко регистрировать уровень глюкозы крови, лекарства, приемы пищи, в одной записи
  + Отслеживание изменения уровня сахара в крови, инсулина, веса, артериального давления, A1C и многого другого!
  + Добавление заметок к вашим записям для дальнейшего использования
  + Запись своих блюд, используя нашу обширную базу данных продуктов питания

Интерфейс приложения представлен на рисунке 1.1.

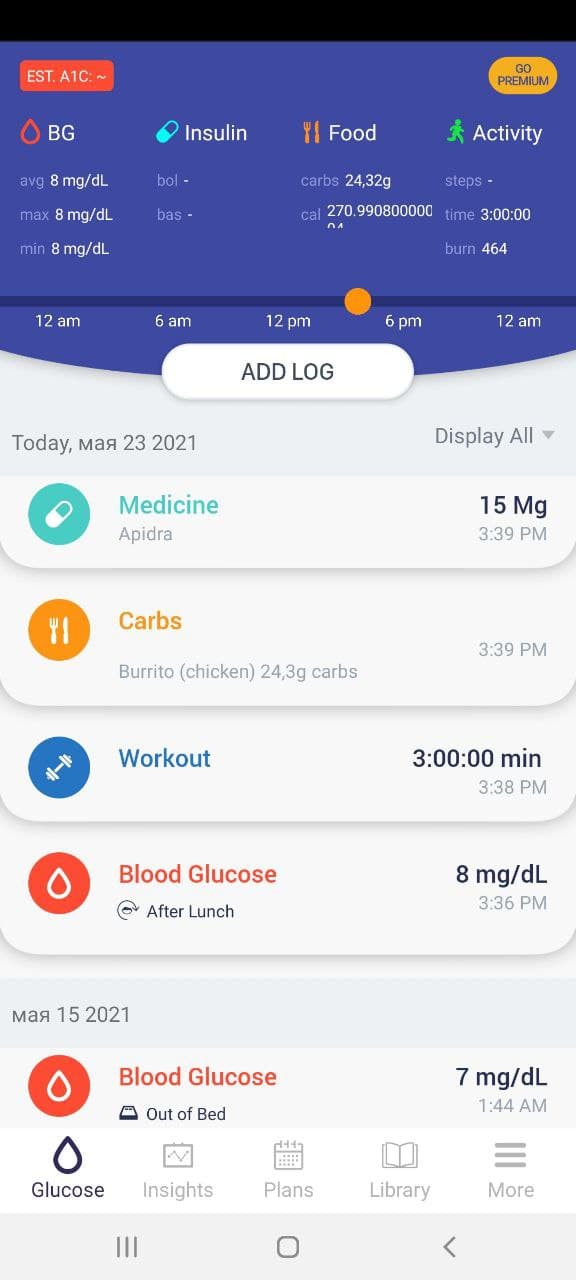


Рисунок 1.1 − Интерфейс приложения Glucose Buddy Diabetes Tracker

Из недостатков необходимо отметить не самый приятный и интуитивно непонятный дизайн программного средства. Большинство возможностей доступны только в платной версии приложения. Отсутствие контроля водного баланса.

## Диабет

Данное приложение так же было разработано под мобильные операционные системы IOS и Android. Имеет следующие возможности:

* Ведение дневника контроля глюкозы
* Ведение дневника питания
* Точные расчеты ХЕ на еду (самая большая база продуктов + возможность добавлять свои продукты)
* Расчет доз инсулина (в том числе дробный инсулин для помпы)
* Экспорт дневника в .pdf и .xls, печать дневника, отправка прямо из приложения в любой меседжер или на почту.

Внешний вид программы представлен на рисунке 1.2.

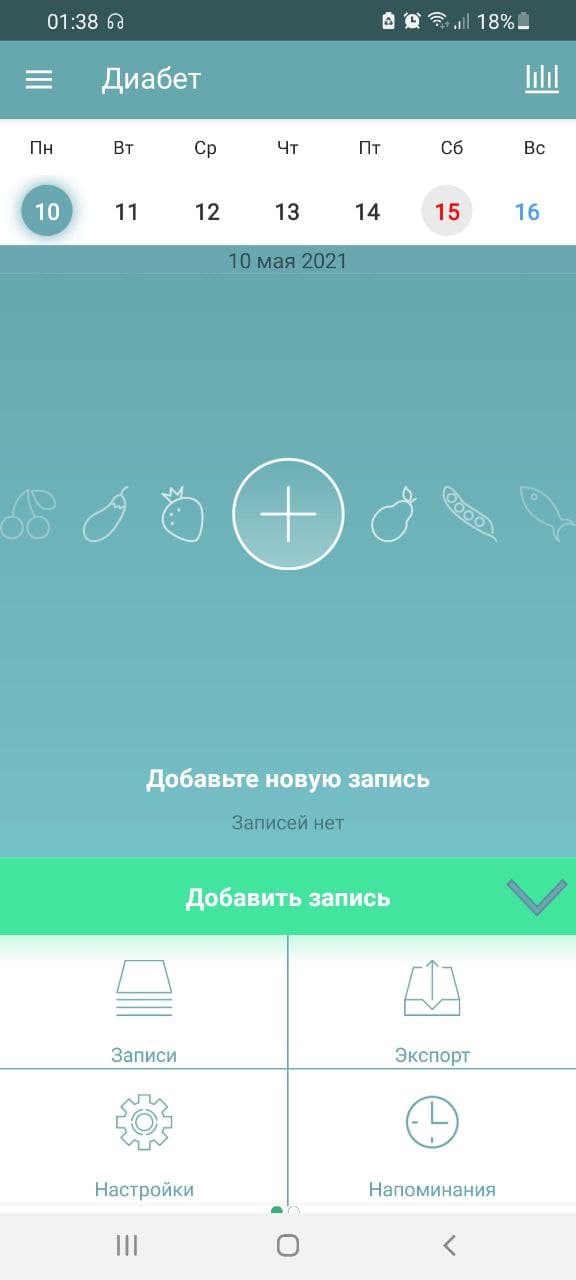


Рисунок 1.2 — Интерфейс приложения Диабет

Диабет, как и приложение, рассмотренное ранее, не отличается удобством в использовании. Его интерфейс выглядит устаревшим и однозначно будет не очень приятен пользователю. В нём так же отсутствует функция записи употреблённых лекарств. Отсутствие контроля водного баланса.

## Diabetes Tracker

Дневник контроля глюкозы крови и питания – важная вещь в жизни диабетика. Приложение Diabetes Tracker создано в помощь как тем, кто впервые столкнулся с этим диагнозом, так и давно болеющим людям. Функционал:

* Удобный ежедневник с графами для внесения результатов многократных измерений
* Заметки, куда можно вписывать любую другую важную информацию: объем текста – 140 знаков
* Архив, который поможет при необходимости получить давние сведения, сохраненные ранее трех месяцев назад
* Статистика – раздел, где анализируются и интерпретируются результаты контроля сахара
* Цветные графики - "сахарная кривая" визуального контроля измерений сахара крови за разные промежутки времени: 2 недели, месяц, 3 месяца. Пользователям также доступен график изменений HbA1c за год
* Интеграция с Google Drive - способ быстро и надёжно разместить в облачном хранилище все результаты измерений сахара крови
* Настройки интерфейса

Главное окно программы представлено на рисунке 1.3.



Рисунок 1.3 — Интерфейс приложения Diabetes Tracker

К недостаткам можно отнести отсутствие контроля водного баланса и то, что приложение поддерживает только один язык.

## Аналогичность и различие программных средств

Схожесть рассмотренных программных средств заключается в том, что все они позволяют следить за нормой КБЖУ, употреблением медицинских препаратов и инсулина.

Главные отличия программ кроются в их функциональности. Так, например, в приложениях Diabetes Tracker и Glucose Buddy Diabetes Tracker доступны заметки, в приложении Diabetes Tracker пользователь может сам регулировать внешний вид приложения, а в приложении Диабет существует возможность сохранения в .pdf и .xls.

Таким образом, можно сделать вывод о том, что в каждом приложении есть свои плюсы и минусы, что и было учтено при выполнении данного курсового проектирования.

# Анализ требований к программному средству и разработка функциональных требований

Для того чтобы приступить к этапу разработки проекта, необходимо четко сформулировать функциональные требования к программному средству.

## 2.1 Определение требований к программному средству

Функционально ПС должно выполнять следующие задачи:

* регистрация и авторизация пользователей;
* сохранение информации в базе данных;
* для администратора одобрение или отклонение предложенных продуктов;
* для пользователя изменение личных данных;
* для пользователя учёт выпитых медикаментов,
* для пользователя водного и пищевого баланса;
* уведомление пользователя об высоком или низком уровне сахара;
* для администратора добавление новых продуктов и рецептов.

## 2.2 Описание средств разработки

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2019;
* программная платформа .NET Framework 5.0.202;
* язык программирования C#;
* расширяемый язык разметки XAML;
* технология WPF;
* технология ADO.NET;
* Microsoft SQL Server.

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

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

Если различные механизмы работы с БД в С#. Т.к. моя база данных небольшая, мой выбор пал на ADO.Net. ADO.Net предоставляет самый прямой способ доступа к данным в .Net Framework.

## 2.3 Описание функциональности программного средства

Описание функциональности программного средства представлено с помощью UML-диаграммы вариантов использования (Приложение А).

Спецификация функциональности программного средства представлена в следующем пункте

## 2.4 Спецификация функциональных требований

Для функциональности ПС необходимо создание базы данных для хранения информации приложения. Подробно о базе данных описано в главе 3.

В программном средстве «Diabeta.net» при запуске необходимо реализовать регистрацию и авторизацию пользователей для дальнейшего использования приложения. Для авторизации входными параметрами являются логин и пароль пользователя, которые содержатся в базе данных. Для того чтобы зарегистрироваться необходимо ввести логин, пароль, фамилию, имя, возраст, рост, вес, пол, желаемый уровень сахара, цель и уровень подвижности. Введенные данные, успешно прошедшие валидацию, заносятся в базу данных, а пользователь получает доступ к приложению.

При авторизации администратора он может на странице «*Одобрить*» просматривать продукты, добавлять и удалять, при нажатии на кнопку «*Продукты*» может просматривать все продукты и добавлять новые. На странице «*Рецепты»* просматривать имеющиеся рецепты, при нажатии на «*Добавить рецепт*» администратор может создать и добавить новый рецепт.

При авторизации пользователя он может на главной странице просматривать и отмечать все съеденные продукты и блюда, весь инсулин и все таблетки, которые употреблял, уровень сахара и выпитые стаканы воды. На странице «*Инфо»* он может просмотреть личную информацию и редактировать её. На странице «*Рецепты*» он может увидеть существующие рецепты и просмотреть их. На страницах статистик пользователь может просмотреть статистику по весу и статистику по уровню сахара в крови.

При авторизации администратора он может на странице «*Товары*» просматривать все товары от всех продавцов и удалять их. На странице «*Заказы»* просматривать заказы и удалять их. На странице «*Продавцы*» просматривать всех продавцов и удалять их. На странице «*Клиенты*» просматривать всех клиентов и удалять их.

Кнопка для выхода из аккаунта возвращает пользователя к окну авторизации, а кнопка для выхода из приложения закрывает приложение.

Все данные о продуктах, рецептах, таблетках и сахаре в крови должны заноситься в базу данных в соответствующие таблицы.

# Проектирование программного средства

## Архитектура проектируемого программного средства

При создании программного средства использовался архитектурный паттерн Model-View-ViewModel (MVVM). Он был представлен архитектором программного обеспечения компании Microsoft Джоном Госсманом в 2005 году. Одной из главных целей шаблона является отделение и абстрагирование видимого пользовательского интерфейса от логики представления. MVVM состоит из трех компонентов: модель (Model), представление (View) и модель представления (ViewModel). Наглядное представление паттерна представлено на рисунке 3.1.

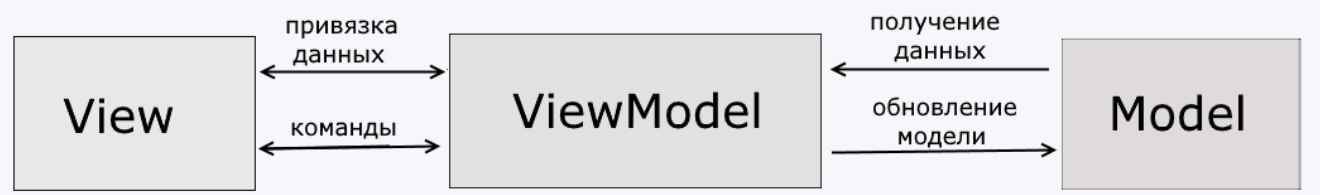


Рисунок 3.1 − Наглядное представление MVVM

Модель (Model) — это субъект, который описывает используемые в приложении данные. Это может быть что угодно, будь то обычный продукт или же сложный рецепт. Представление (View) — это графический элемент управления или набор элементов управления, отвечающих за отображение данных на экране. Модель представления (ViewModel) — это и есть та магия, стоящая за всем. Она содержит логику пользовательского интерфейса, команды, события и ссылку на модель. В MVVM модель преставления не обязана обновлять непосредственно данные представления. Благодаря мощному механизму привязки, представленному в WPF, пользовательский интерфейс обновляется, как только обновляется модель преставления. Для этого модель преставления должна реализовывать интерфейс INotifyPropertyChanged и запустить событие PropertyChanged [4].

### **3.1.1** **Реализация MVVM**

Для работы с паттерном использовалась библиотека DevExpressMVVM. Для реализации паттерна файлы программы были распределены по соответствующим пространствам имен (см. 2.3) и реализованы соответствующие функции. На рисунке 3.1 видим, что класс MainViewModel наследует класс ViewModelBase, который лежит в пространстве имен DevExpress.Mvvm. Метод

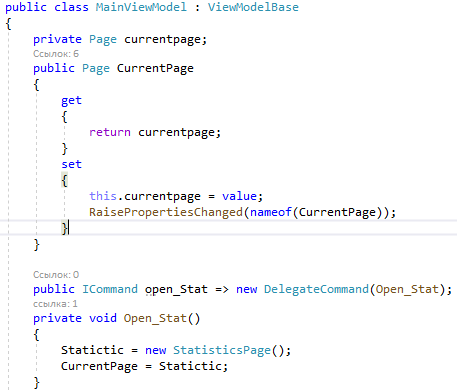


Рис. 3.1 – Пример использования DevExpressMVVM

### **3.1.2 Схема и описание авторизации**

Схема авторизации представлена в приложении B. Сразу после запуска программу пользователь попадает в окно авторизации. Если поля логина и пароля будут пустыми, в окне будет выведено сообщение об ошибке. После программа обращается к базе данных и проверяет существует ли пользователь с соответствующим логином и паролем. Если пользователь не найден, будет выведено сообщение об ошибке, в противном случае вход в приложение.

### **3.1.3 Построение графиков в WPF**

NuGet предоставляет достаточное количество библиотек для построения графиков. Мой выбор упал на библиотеку LiveCharts. Она очень проста в использовании (пример заполнения рисунок 3.2).

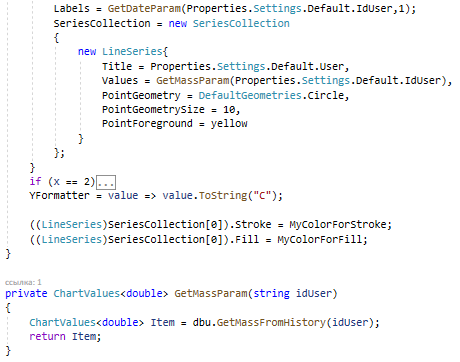


Рис. 3.2 – Пример использования LiveCharts

### **3.1.4** **Подключение к БД**

Если различные механизмы работы с БД в С#. Т.к. моя база данных небольшая, мой выбор пал на ADO.Net. ADO.Net предоставляет самый прямой способ доступа к данным в .Net Framework. В Приложении Г продемонстрирован листинг одного из запросов к БД.

## Разработка базы данных

Для обеспечения работоспособности программного средства была разработана база данных, что являлось обязательным условием курсового проекта. База данных была тщательно сконструирована до начала написания самого приложения. Для ее создания использовалась реляционная система управления базами данных (РСУБД) MS SQL.

Таким образом была разработана база данных KP\_DataBase, которая включает в себя таблицы Users, Daily\_Food, Recipe, Prod\_Rec, Products, Type\_of\_Food, Daily\_Insulin, Type\_of\_Insulin, Daily\_Pill, Daily\_Cal, Daily\_Water, History, History\_Blood\_Sugar, Products\_Awaiting\_Approval. Схема базы данных представлена на рисунке 3.2

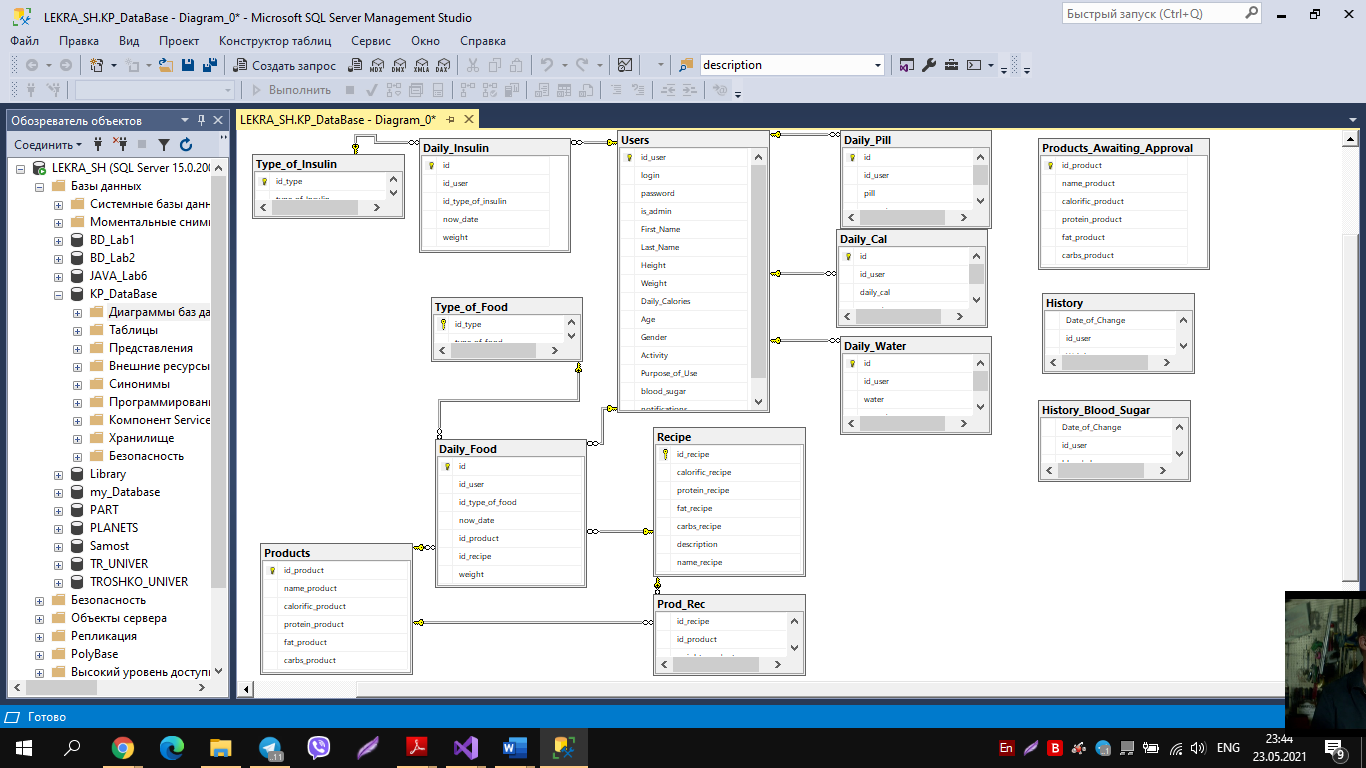


Рисунок 3.2 — Схема базы данных KP\_DataBase

Для хранения информации о пользователях была создана таблица Users. Включает 15 столбцов: id\_user – id пользователя, login – логин пользователя, password – пароль пользователя, при записи в базу данных хэшируется, isadmin – 0 – пользователь, 1 - администратор, First\_Name – имя пользователя, Last\_Name – фамилия пользователя, Height – рост, Weight – вес, Daily\_Calories – дневная норма калорий, Age – возраст, Gender – пол, Activity – коэффициент активности, Purpose\_of\_Use – цель использования, blood\_sugar – коэффициент уровня сахара в крови, notifications – поле доступа для уведомлений.

Таблица Daily\_Food используется для хранения съеденной пищи пользователем в течении дня.

Таблица Recipe используется для хранения информации о рецептах. Включает 7 столбцов: id\_recipe – id рецепта, name\_ recipe – название рецепта, calorific\_ recipe – калорийность рецепта на 100г, protein\_ recipe - белки рецепта на 100г, fat\_ recipe – жиры рецепта на 100г, carbs\_ recipe - углеводы рецепта на 100г, description – способ приготовления.

Таблица Prod\_Rec используется для связи между таблицами Recipe и Products. Используется для определения того, какие ингредиенты входят в состав блюда. Включает 3 столбца: id\_recipe – id рецепта, id\_product – id продукта, weight\_product – вес продукта, необходимого для рецепта.

Таблица Products используется для хранения продуктов. Включает 6 столбцов: id\_product – id продукта, name\_product – название продукта, calorific\_product – калорийность продукта на 100г, protein\_ product – белки продукта на 100г, fat\_ product – жиры продукта на 100г, carbs\_ product - углеводы продукта на 100г.

Таблица Type\_of\_Food используется для хранения данных о типах приема пищи. Включает 2 столбца: id\_type – id приема пищи, type\_of\_food – тип приема пищи (завтрак, обед, ужин, перекус).

Таблица Daily\_Cal хранит оставшееся число калорий пользователя на день.

Таблица Daily\_Water используется для хранения выпитой воды пользователем в течении дня.

Таблица Daily\_Pill используется для хранения выпитых медикаментов пользователем в течении дня.

Таблица Daily\_Insulin используется для хранения сделанных уколов пользователем в течении дня.

Таблица Type\_of\_Insulin используется для хранения данных о типах уколов. Включает 2 столбца: id\_type – id приема инсулина, type\_of\_insulin – тип укола инсулина (ночной, дневной).

Таблица Product\_Awaiting\_Approval используется для хранения добавленных пользователем новых продуктов. Элементы жду подтверждения администратора.

Таблица History используется для хранения истории веса пользователя. Включает 3 столбца: Date\_of\_Change – дата изменения веса,id\_user- id пользователя, Weight – обновленный вес. Таблица History автозаполняется, когда срабатывает триггер таблицы Users.

Таблица History\_Blood\_Sugar используется для хранения истории уровня сахара пользователя. Включает 3 столбца: Date\_of\_Change – дата изменения веса,id\_user- id пользователя, blood\_shugar – обновленный Уровень сахара. Таблица History автозаполняется, когда срабатывает триггер таблицы Users.

В базе данных между таблицами Users и Daily\_Insulin реализована связь один-ко-многим, т. к. одному пользователю может соответствовать несколько записанных уколов, а каждый укол в таблице Daily\_Insulin соответствует только одному пользователю. Аналогичная связь реализована в таблицах Users и Daily\_Pill и ещё много где. Схематическое представление связи представлено на рисунке 3.3.

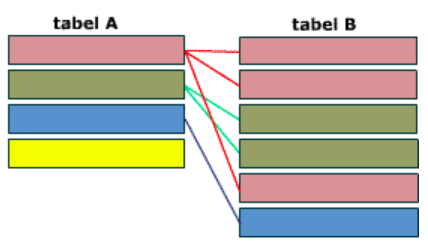


Рисунок 3.3 − Структура связи один-ко-многим

## Обобщенная структура проекта

В общей сложности проект включает в себя 4 папки. Его структура представлена на рисунке 3.4.

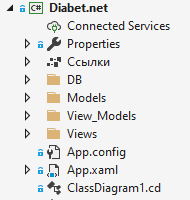


Рисунок 3.4 − Структура проекта

В папке Models хранятся классы, описывающие логическую обработку данных проекта.

Папка View\_Models содержит в себе классы со свойствами и командами, которые связаны механизмом привязки с объектами папки Views.

В папке Views содержатся элементы пользовательского интерфейса, которые пользователь сможет увидеть на экране.

Содержимое папки DB представляет из себя все взаимодействия с базами данных.

Взаимосвязь между классами хорошо видна в диаграмме классов (Приложение Б).

Приложение включает в себя 13 окон и 12 страниц. Связи между ними представлены на рисунке 3.5.

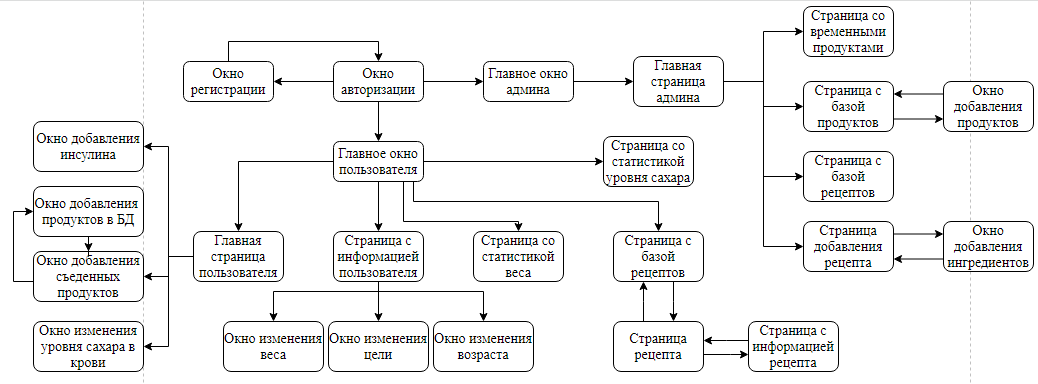


Рис. 3.5 – Схема связи между окнами

# Описание интерфейсов программного средства, его тестирование и анализ полученных данных

## Описание интерфейсов программного средства

Сразу после запуска приложения перед пользователем открывается окно авторизации, в которое требуется ввести логин и пароль. Внешний вид окна представлен на рисунке 4.1.

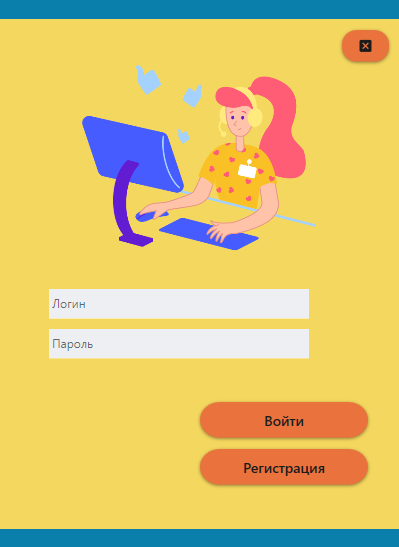


Рисунок 4.1 − Внешний вид окна авторизации

Из окна авторизации пользователи, которых нет в базе данных приложения, могут зарегистрироваться, нажав на кнопку «Регистрация». После того, как это произойдет, откроется окно регистрации новых пользователей, интерфейс которого представлен на рисунке 4.2.

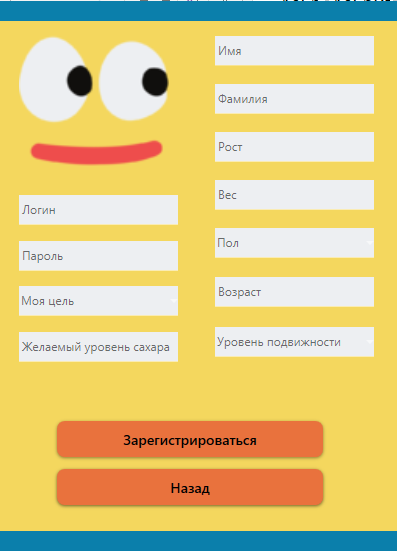


Рисунок 4.2 − Интерфейс окна регистрации

Также в окне регистрации есть функция возврата к окну авторизации, которую реализует элемент управления с надписью «Назад».

### **4.1.1 Вход в качестве пользователя**

После успешной авторизации открывается главное окно приложения, которое встречает пользователя. С главной страницы которого мы можем добавить съеденную еду, контролировать воду, медикаменты и инсулин, вводить данные об уровне сахара в крови, включать/выключать уведомления, а так же перейти на другие страницы приложения. Внешний вид главного окна для пользователя представлен на рисунке 4.3.

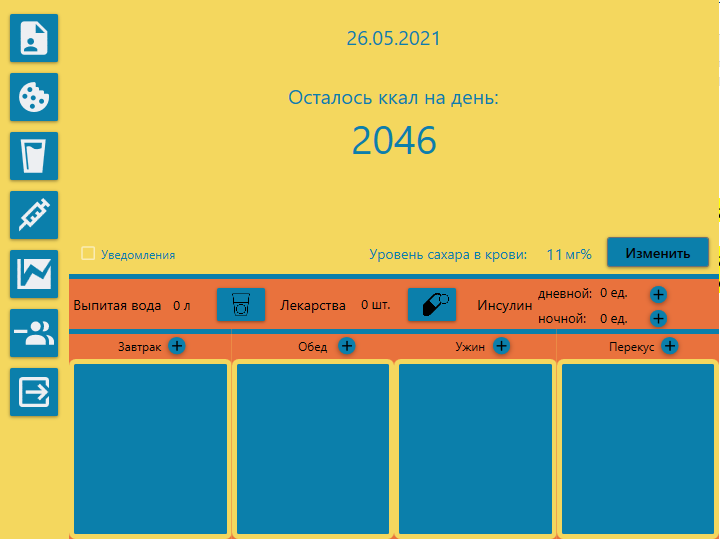


Рисунок 4.3 − Внешний вид главного окна пользователя

При нажатии на кнопки в левой боковой панели открываются различные страницы «Личный кабинет», «Рецепты», «Статистика уровня сахара в крови», «Статистика веса», а так же можно выйти из аккаунта или со всего приложения. На рисунках 4.4 – 4.7 представлен внешний вид этих страницах.

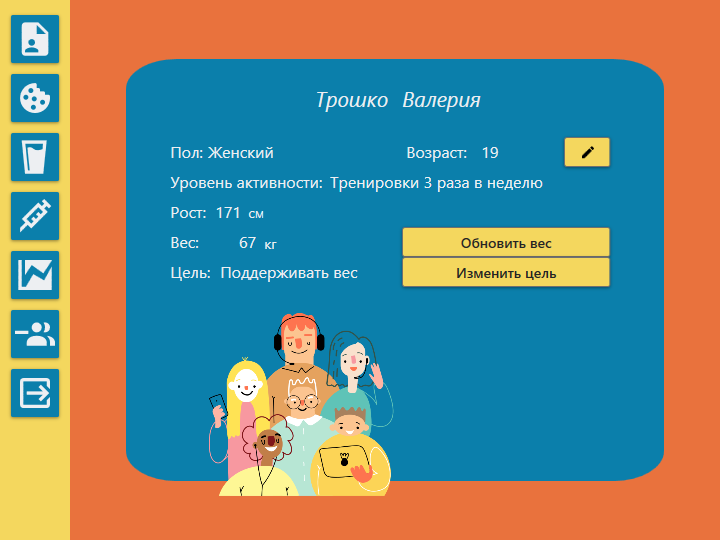


Рисунок 4.4 − Страница «Личный кабинет»

В «Личном кабинете» пользователь может просмотреть информацию о себе, а так же внести изменения веса, возраста, цели.

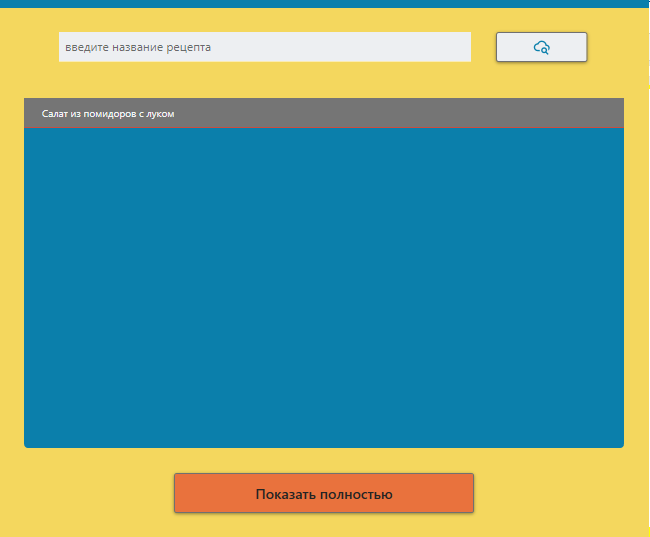


Рисунок 4.5 − Страница «Рецепты»

На странице «Рецепты» пользователь может выбрать и просмотреть различные рецепты, при нажатии кнопки «Показать полностью» мы может ознакомится с рецептом, а точнее с его КБЖУ, ингредиентами и способом приготовления рис. 4.6.

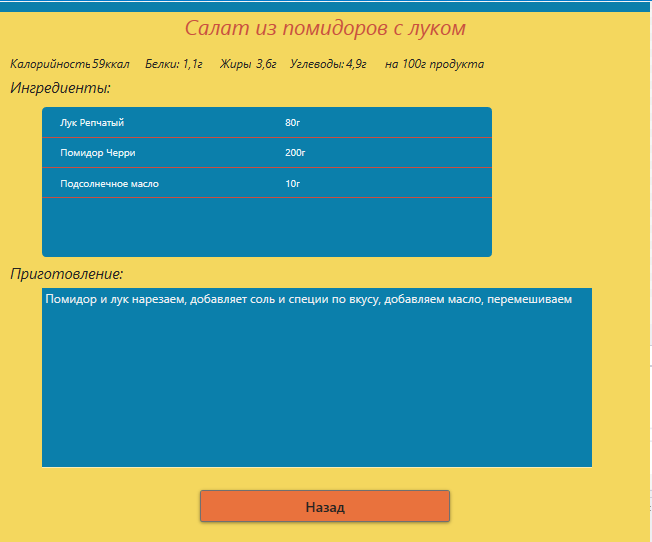


Рисунок 4.6 − Страница «Информация рецепта»

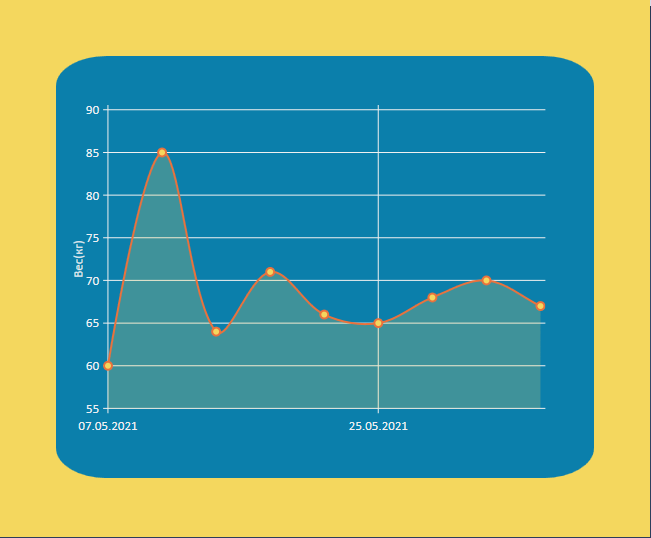


Рисунок 4.7 − Страница «Статистика веса»

Страницы «Статистика веса» и «Статистика уровня сахара в крови» построены по одному принципу.

### **4.1.2 Вход в качестве администратора**

После успешной авторизации открывается главное окно приложения (рис. 4.8), которое встречает администратора.



Рисунок 4.8 − Внешний вид главного окна администратора

На рисунке 4.9 продемонстрирована таблица админа, где находятся продукты, которые ожидают подтверждения.

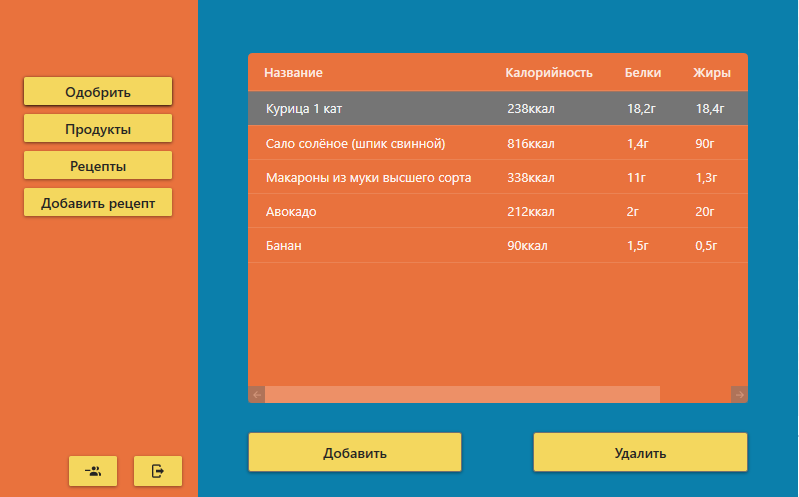


Рисунок 4.9 − Страница продуктов для одобрения

Админ контролирует базу продуктов и базу рецептов (рис. 4.10). Соответственно он может пополнять их.

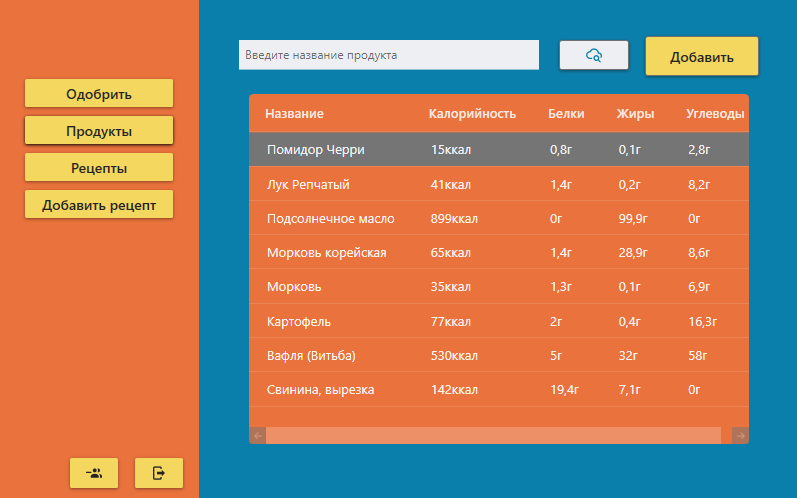


Рисунок 4.10 − Страница базы продуктов

## Тестирование программного средства и анализ полученных данных

Для проверки работоспособности программного средства было проведено его тщательное тестирование, в ходе которого осуществлялась проверка заполняемости базы данных, корректность извлекаемых из базы данных значений, а также реакция программы на вводимые, заведомо неверные, данные. Было проработано множество ситуаций, которые могут возникнуть в ходе работы с программным средством.

Так, при входе осуществляется проверка на действительные логин и пароль. Например, пользователь ввел неверный пароль или логин (результат на рис 4.11):

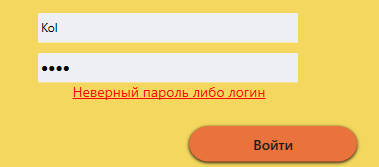


Рисунок 4.11 − Результат входа с неверным логином или паролем

Или вы упустили пустое поле и нажали войти (результат на рис. 4.12):

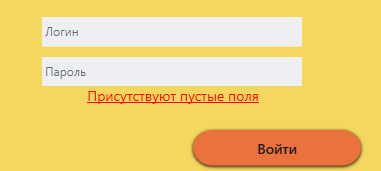


Рисунок 4.12 − Пустые поля при попытке авторизации

При рассмотрении исключительных ситуаций в окне регистрации также была организована проверка вводимых пользователем данных. Так, например, если при регистрации пароль будет меньше 8 символов, то (результат на рис. 4.13):

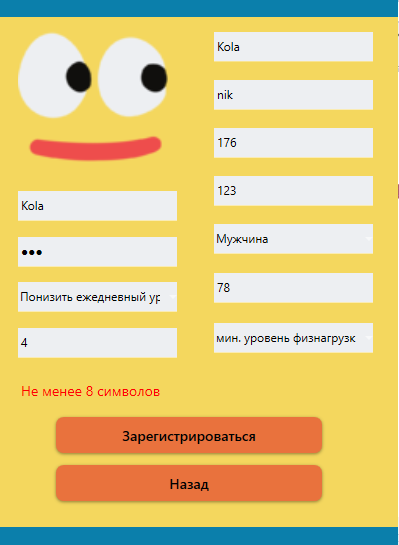


Рисунок 4.13 − Короткий пароль

При попытке изменить уровень сахара, вес, цель, возраст, но поле изменения оставить пустым (результат на рис 4.14):

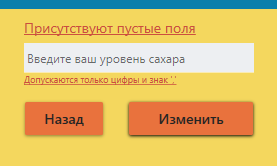


Рисунок 4.14 − Пустые поля при попытке изменить данные

Или при попытке изменения ввести значение идентичное прошлому (результат на рис 4.15):

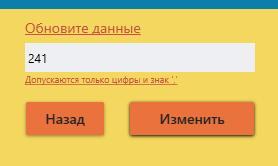


Рисунок 4.15 − Новые данные равные старым

И вот еще пример, где мы хотели добавить продукт, при этом мы не вводили его массу (результат рис. 4.16):

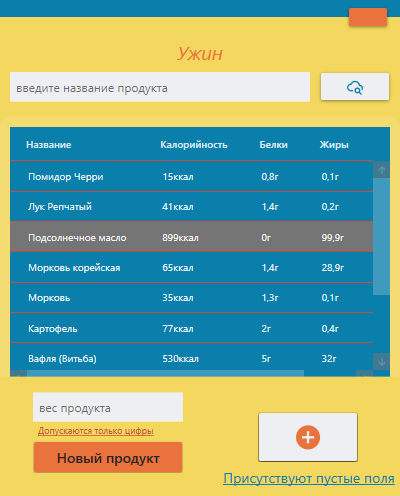


Рисунок 4.16 − Пустые поле веся при добавлении блюда в рацион

# Руководство по установке и использованию

## Установка программного средства

Для работы программного средства необходима первоначальная настройка. Для начала, необходимо наличие установленной на компьютере системы управления базами данных Microsoft SQL Server, а также утилиты SQL Server Management Studio для управления и администрирования компонентов Microsoft SQL Server.

Необходимо войти в SQL Server Management Studio, где в обозревателе объектов раскрыть список текущего подключения и нажать правой кнопкой мыши на вкладке «Базы данных». В контекстном меню требуется выбрать пункт «Присоединить», как на рисунке 5.1.

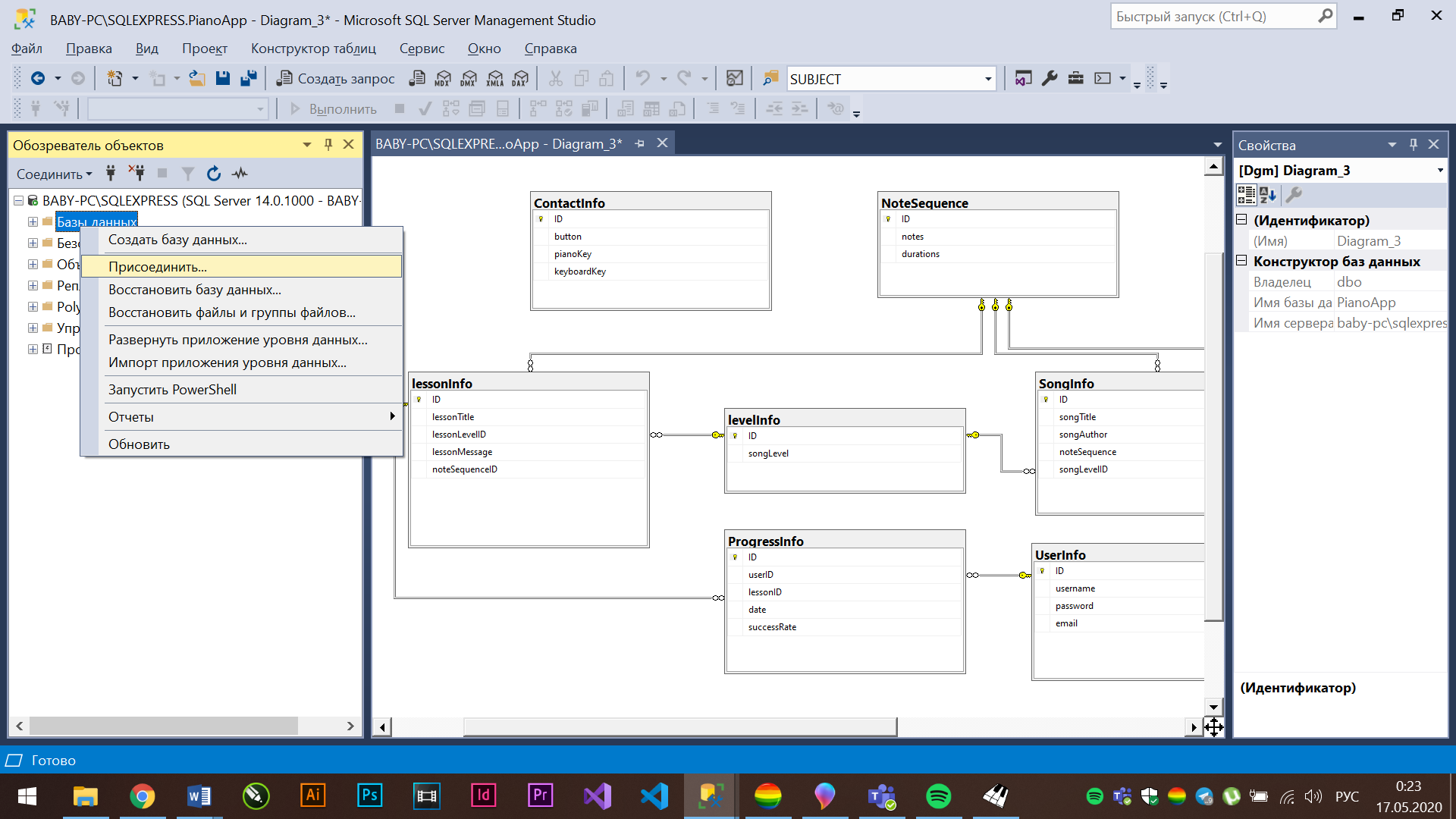


Рисунок 5.1 − Выбор пункта «Присоединить»

В открывшемся окне на панели «Базы данных для присоединения» необходимо нажать кнопку «Добавить» и указать физический путь к базе данных проекта PianoApp. После этого нажать на кнопку «ОК».

После всех проделанных шагов в раскрывающемся списке «Базы данных» обозревателя объектов должна появиться база данных PianoApp, как на рисунке 5.2.

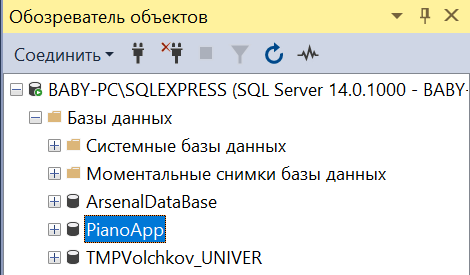


Рисунок 5.2 − Появление базы данных PianoApp

Теперь программное средство готово к запуску.

## Руководство пользователя по работе с программным средством

После успешного присоединения базы данных PianoApp пользователю необходимо запустить файл YourPiano.exe. После открытия окна авторизации новому пользователю требуется перейти к окну регистрации, в котором правильно заполнить все поля. Условия валидности вводимых данных можно прочитать, если навести курсор мыши на название поля.

Так, например, имя пользователя должно быть уникальным, т. е. пользователей с таким именем быть не должно, и содержать от 4 до 15 символов. В состав пароля должны входить латинские буквы верхнего и нижнего регистра, а также цифры, с общей длиной от 4 до 10 символов. Электронный адрес же, по условию регулярного выражения, должен состоять из четырех частей: текстовых данных до символа «@», самого символа «@», и еще двух частей после символа «@», разделенных точкой. Пример правильного заполнения полей показан на рисунке 5.3.

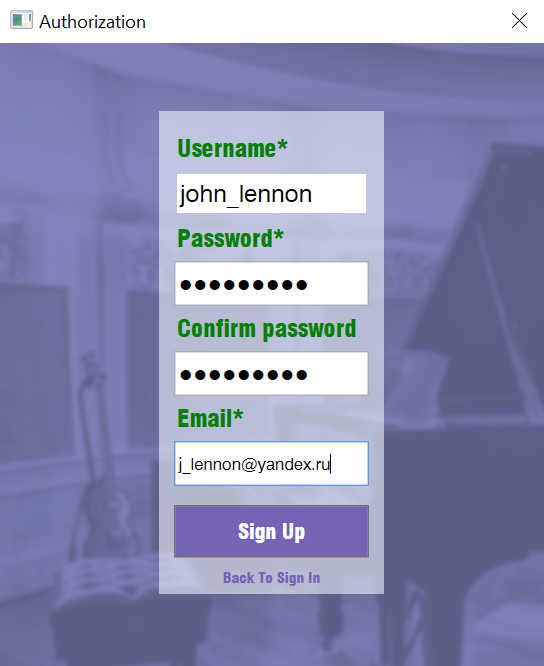


Рисунок 5.3 − Пример правильного заполнения полей окна регистрации

После успешной регистрации, пользователь авторизуется и попадает на главную страницу программного средства.

С главной страницы можно стразу попасть на страницу выбора режима игры, нажав на соответствующую кнопку.

В режиме «Свободной игры» пользователю доступна функция выбора и воспроизведения различных композиций. Для этого необходимо нажать на кнопку «SongBank» в левом верхнем углу. На экране появятся две кнопки «App songs» и «User’s songs», представленные на рисунке 5.4.

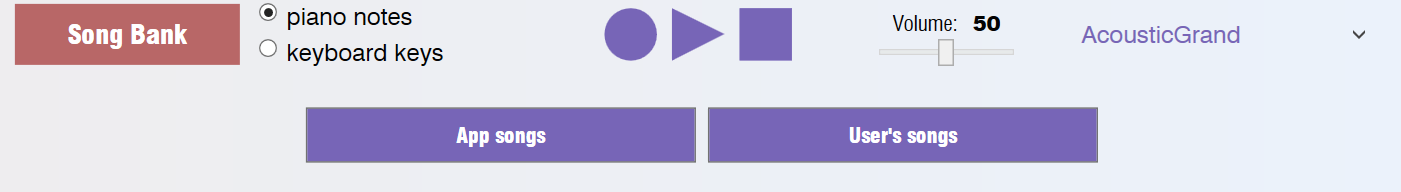


Рисунок 5.4 ­− Отображение кнопок «App songs» и «User’s songs»

Кнопка «App songs» отвечает за отображение таблицы композиций, добавленных самим разработчиком в базу данных PianoApp. Кнопка «User’s songs» отвечает за отображение композиций, записанных пользователями приложения.

После выбора необходимой композиции пользователю открывается панель, на которой отображаются текущая, предыдущая и следующая нота в выбранной композиции. Нажатие на кнопку «Play» начинает воспроизведение композиции, нажатие на кнопку «Stop» останавливает воспроизведение.

Изначально режимом отображения нот является нотная система, т. е. на экране отображаются названия самих нот. Неопытных пианистов это может поставить в тупик, поэтому режим отображения можно сменить, выбрав «keyboard keys» вместо «piano notes» на панели инструментов.

Чтобы осуществить запись собственной композиции пользователю необходимо нажать на кнопку «Record» на панели инструментов. После ее нажатия она приобретет красный цвет, как на рисунке 5.5.



Рисунок 5.5 − Включение функции записи

После того, как пользователь записал желаемую последовательность, ему необходимо повторно нажать на кнопку «Record», чтобы запись закончилась. Затем в появившемся диалоговом окне согласиться с сохранением записанного произведения. В появившемся окне ввести название произведения, как, например, на рисунке 5.6.

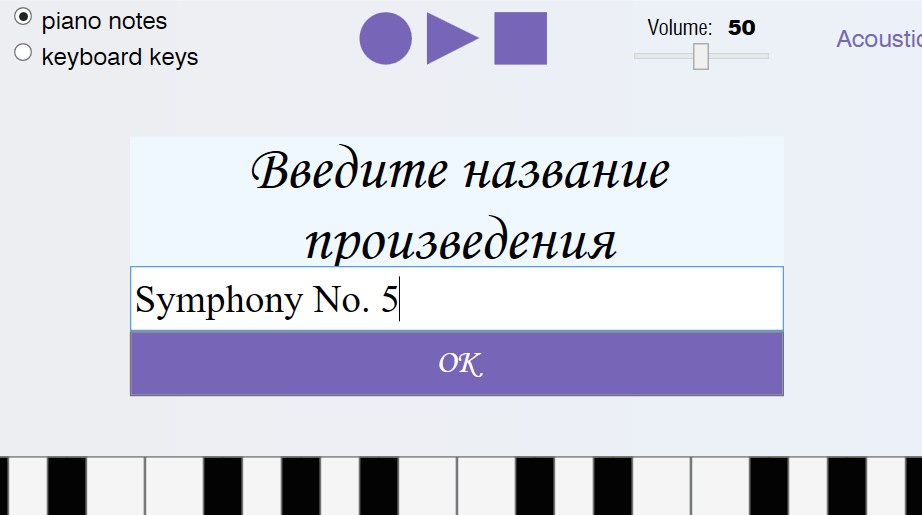


Рисунок 5.6 − Ввод названия сохраняемого произведения

Сразу после ввода названия произведение станет доступным для прослушивания, а таблица с пользовательскими композициями автоматически обновится, где можно будет найти и только что записанное произведение.

Также стоит отметить, что в данном режиме пользователь может регулировать громкость воспроизведения, а также выбирать инструмент, на котором и будет воспроизводиться композиция. Элементы управления, отвечающие за эти функции, показаны на рисунке 5.7.

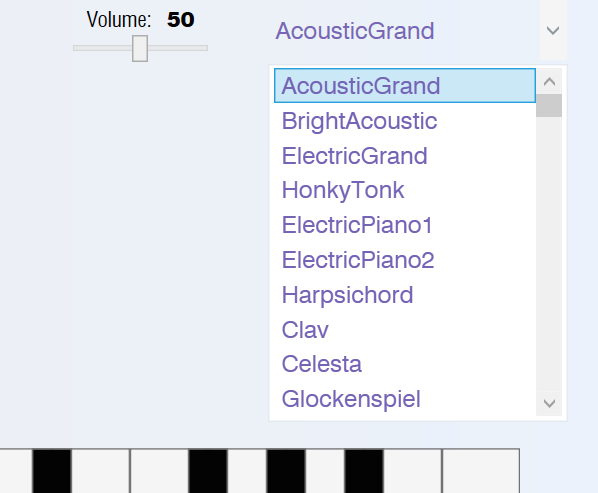


Рисунок 5.7 ­− Элементы управления регулировки громкости и смены инструмента

В режиме «Обучения» пользователю предлагается выбрать урок для его последующего прохождения из таблицы со всеми уроками. После прохождения первого урока станет доступен второй, после прохождения второго третий и т. д.

Главная отличительная особенность режима «Обучения» от режима «Свободной игры» заключается в том, что пользователю больше недоступна функция смены режима отображения нот. Также нет функции ручного регулирования воспроизведения с помощью стрелок. При возникновении трудностей пользователь может обратиться за помощью. Для этого он должен нажать на кнопку «Help!» на панели инструментов. При нажатии появляется «шпаргалка», на которой пользователь сможет узнать название интересующей его ноты. Она представлена на рисунке 5.8.

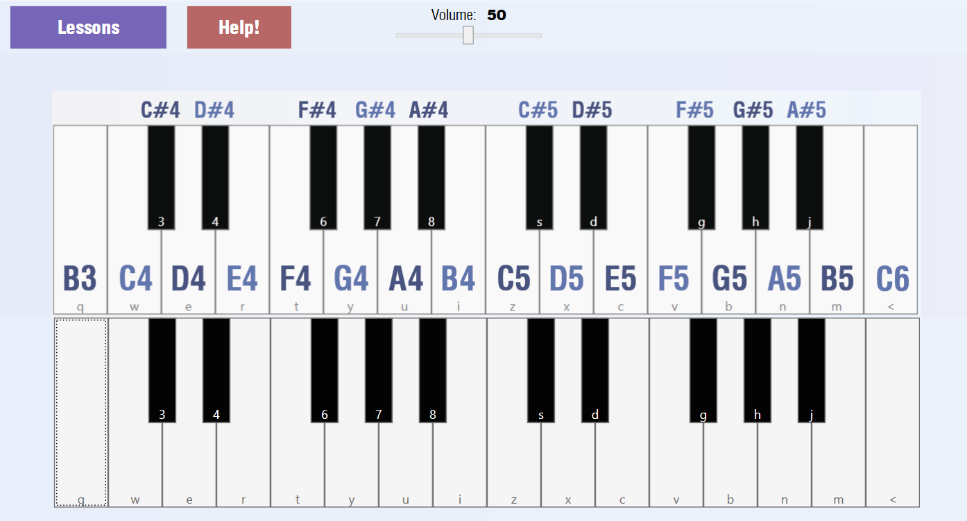


Рисунок 5.8 ­− Отображение функции помощи

Отслеживать свой прогресс пользователю позволяет страница «Прогресс», на которую он может перейти, нажав соответствующую кнопку в главном окне приложения на навигационной панели слева. Наглядное расположение кнопки показано на рисунке 5.9.

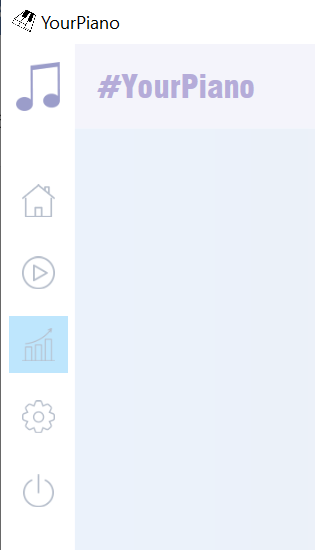


Рисунок 5.9 ­− Кнопка перехода на страницу «Прогресс»

На этой странице пользователь сможет найти всю информацию о текущем количестве пройденных уроков, среднем проценте правильно нажимаемых клавиш фортепиано, проценте пользователей, которых данный пользователь превосходит по количеству пройденных уроков, или, в случае их равенства, среднему проценту успешности прохождения уроков, а также суммарном количестве записанных композиций.

Если же пользователь захочет начать прохождение уроков заново, или вовсе полностью удалить свой профиль, для таких целей существует страница «Настройки», для открытия которой необходимо нажать кнопку, изображенную на рисунке 5.10.

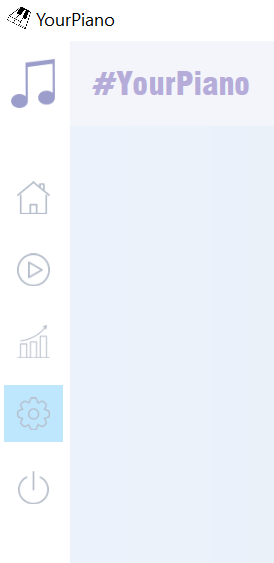


Рисунок 5.10 ­− Кнопка перехода на страницу «Настройки»

Для выхода из приложения предусмотрена кнопка «Выход», расположенная в главном окне приложения на навигационной панели слева в самом низу. Ее расположение продемонстрировано на рисунке 5.11.

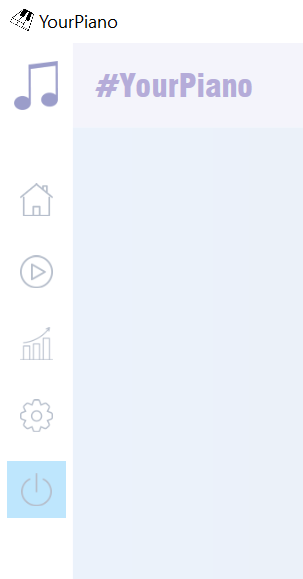


Рисунок 5.11 ­− Кнопка выхода из приложения

# Заключение

В ходе выполнения курсовой работы был разработан проект, являющийся помощником в управлении режимом питания и контролем веса, с уклоном для людей с диагнозом «Сахарный диабет». Во время выполнения данной курсовой работы было изучено немало теоретического материала, а также просмотрено и разобрано большое количество уже готовых решений тех или иных задач.

В ходе выполнения курсовой работы было выполнено следующее:

- разработка базы данных для хранения информации;

- разработка архитектуры приложения;

- разработка функциональных части приложения;

- разработка пользовательского интерфейса;

- написание исходного кода приложения;

- тестирование приложения.

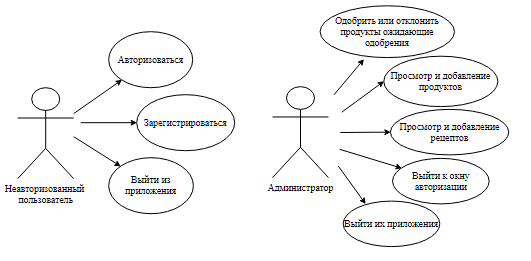
В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает, верно, а требования технического задания выполнены в полном объеме.

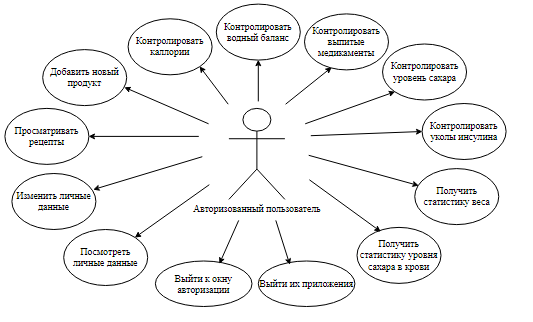
Само приложение – первый опыт в данной области. Естественно, существует ряд приложений, имеющих функционал и дизайн в разы лучше. Подводя итог всей курсовой работе, можно сделать выводы, что в разработке приложений мне предстоит еще долгий путь.

# Список использованных источников

1. Everyone Piano − Forever Free Piano Software [Электронный ресурс]. − Режим доступа: https://www.everyonepiano.com/ Дата доступа: 17.05.2020
2. Dream Computer Piano 2.10 [Электронный ресурс]. − Режим доступа: https://www.malavida.com/en/soft/dream-computer-piano/ Дата доступа: 17.05.2020
3. PC 73 Virtual Piano Keyboard [Электронный ресурс]. − Режим доступа: http://www.chordpulse.com/pc-virtual-piano-keyboard.html Дата доступа: 17.05.2020
4. Garofallo, R. Building Enterprise Applications with Windows Presentation Foundation and the Model View ViewModel Pattern / R. Garofallo − California: O’Reilly Media, − 2011 − 220 p.
5. Информационный сайт: базы данных [Электронный ресурс]. Режим доступа:[inform.sch117nn.edusite.ru/DswMedia/klyaks\_netuchitelyutestyibazyidannyix.htm](http://inform.sch117nn.edusite.ru/DswMedia/klyaks-_netuchitelyutestyibazyidannyix.htm) – Дата доступа: 24.05.2020.
6. Сайт о программировании Metanit: Введение в ADO.NET [Электронный ресурс]– Режим доступа: <https://metanit.com/sharp/adonet/1.1.php> – Дата доступа: 24.05.2021.

**Приложение А**



****

**Приложение Б**

