**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ**

**«ПЕРВЫЙ МОСКОВСКИЙ ОБРАЗОВАТЕЛЬНЫЙ КОМПЛЕКС»**

(ГБПОУ «1-й МОК»)

Допустить к защите

Заведующая отделением среднего профессионального образования факультета «Информационные технологии и управление»

\_\_\_\_\_\_\_\_\_\_\_\_/Пашохонова Е.Е.

 «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г

**ДИПЛОМНЫЙ ПРОЕКТ**по специальности

09.02.04 «Информационные системы (по отраслям)»

(наименование специальности)

Создание информационной системы поддержки самостоятельного обучения профессиональному английскому языку.

**Тема: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Седов Артем Вадимович

группы \_\_\_\_\_

41ИС

(фамилия, имя, отчество)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тузовский Анатолий Федорович

(фамилия, имя, отчество)

**Работа защищена «\_\_»** июня 2021 года **с оценкой**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Секретарь Государственной

экзаменационной комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Москва

2021

Оглавление

[Введение 3](#_Toc69932681)

[1. анализ предметной области 5](#_Toc69932682)

[1.1 Функциональное моделирование 6](#_Toc69932683)

[1.2 Объектно-ориентированное моделирование 9](#_Toc69932684)

[2. описание реализации программного обеспечения информационной системы 17](#_Toc69932685)

[2.1 Описание архитектуры программного обеспечения 17](#_Toc69932686)

[2.2 Создание базы данных 17](#_Toc69932687)

[2.2 Разработка программного кода 20](#_Toc69932688)

# Введение

В последние десятилетия изучение иностранных языков вызывает повышенный интерес. При этом отмечается возрастающая роль, которую играют иностранные языки в осуществлении влияния на сознание и деятельность людей. Также необходимо учитывать, что знание языков может играть важную роль и давать некоторые преимущества в личной и профессиональной коммуникации.

Самый распространённый иностранный язык для обучения является английский язык. Люди, которые идут в институты или техникумы на специальность в сфере IT обязаны знать английский, так как все понятия написаны на английском. В связи с этим, многие люди вынуждены самостоятельно заниматься вопросом изучения. Конечно же, это нелегкий путь, но наличие современных технологий способно во много раз упростить данный процесс.

Данная работа направлена на то, чтобы пользователь при работе с информационной системы мог легко обучатся профессиональному английскому языку. Интерфейс очень простой и пользователя-дружелюбный, который приведет к быстрому и легкому обучению.

Целью дипломного проекта является проектирование и разработка подсистемы обучения английскому языку, которое позволит облегчить изучение и практики английского языка в IT сфере.

Для осуществления цели нужно выполнить следующие задачи:

* провести анализ использования информационных технологий в процессе обучения английскому языку;
* провести анализ предметной области;
* выполнить описание предметной области с использованием моделей структурного и объектно-ориентированного подхода;
* определить требования к функциональным характеристикам будущей подсистемы в UML-моделях;
* выполнить проектирование подсистемы обучения английскому языку;……
  + описание решаемых задач
  + логическая модель данных
  + макет графического интерфейса
* разработать базу данных с использованием СУБД Microsoft SQL Server и SQL Server Management Studio (SSMS);
* разработать программный код подсистемы с помощью среды разработки Visual Studio.
* провести тестирование разработанной подсистемы с целью определения и повышения ее качества.

# анализ предметной области

Информационная система «Rattlesnake» специализируется в профессиональном самообучении английскому языку в IT-сфере пользователя путем предоставления много изучаемого материала, тестов по темам и словаря.

Процесс обучения систематизирован. Пользователю дается материал для изучения, после чего он проходит тестовую часть для отслеживания прогресса и перехода на следующую тему, которая будут уже посложнее.

Сначала пользователь выбирает первую тему и изучает учебный материал. Учебный материал предоставляется пользователю по темам и по уровню сложности. После того как он изучил материал он может перейти на тестирование.

Для каждой учебной темы есть несколько тестов, который пользователь должен проходить чтобы перейти на следующую тему. Он также может проходить эти тесты несколько раз и результат их будут отображаться в профиле пользователя. Результат тестов отображается в процентах.

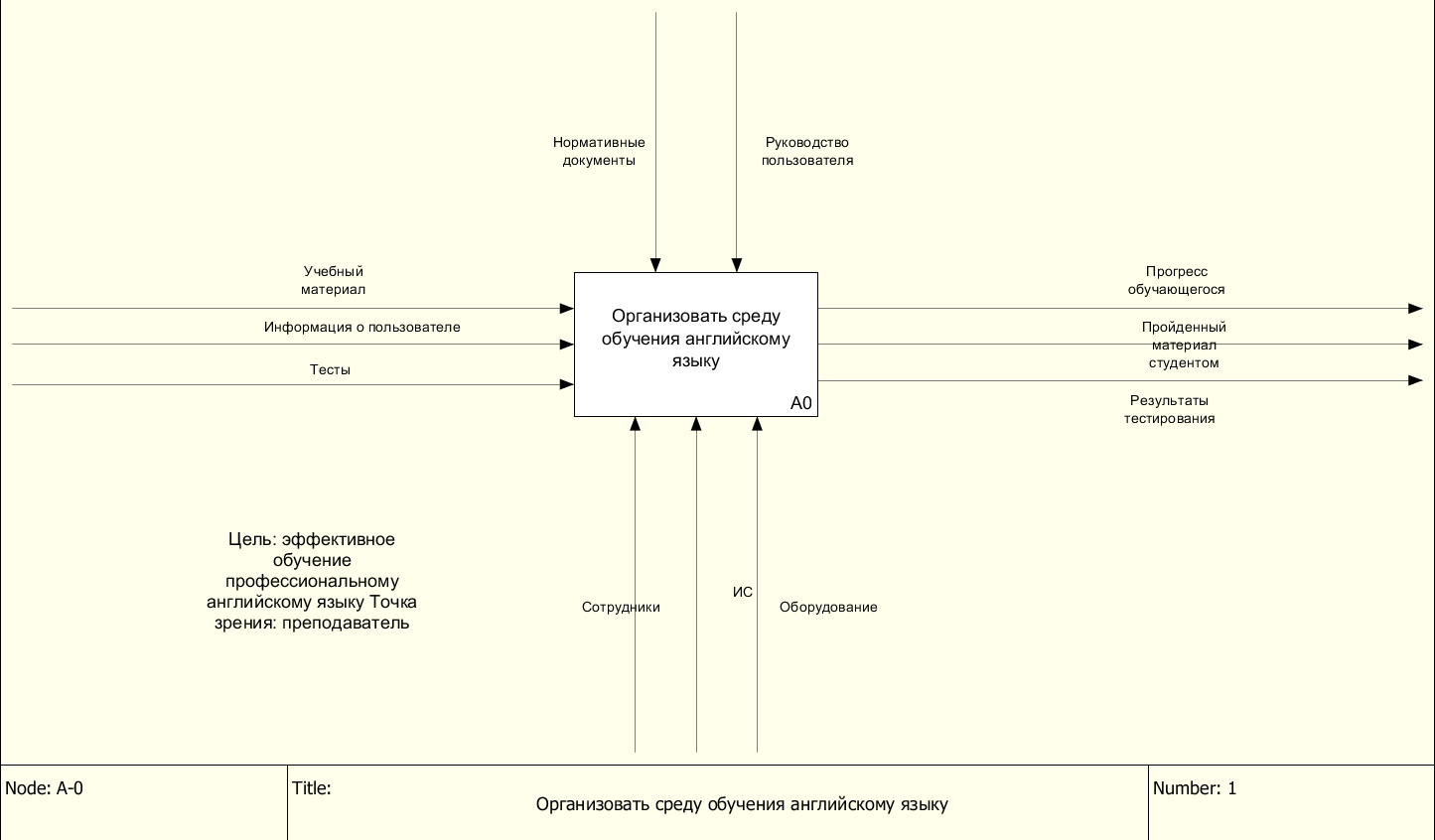
После прохождении всех тестов по теме, пользователь может переходить на следующий тему, изучая учебный материал и также выполняя тестовые задания по ним.

В прогрессе обучения, сложные профессиональные слова из учебных материалов будут добавляться в словарь. Пользователь может просматривать новые слова в словаре, посмотреть перевод на русский или просто повторить.

## Функциональное моделирование

**Контекстная диаграмма**

В диаграмме бизнес-процессом является «Организовать среду обучения английскому языку». Представлен в функциональном блоке, для выполнения которого на входе используется информация о пользователе, учебные материалы и тесты. В качестве управления выступают нормативные документы и руководство пользователя. Инструментами выполнения главной бизнес-функции служат сотрудники, сама информационная система и оборудование (компьютеры). На выходе мы получаем прогресс пользователя, пройденный материал и результат тестирования. (Рисунок 1).



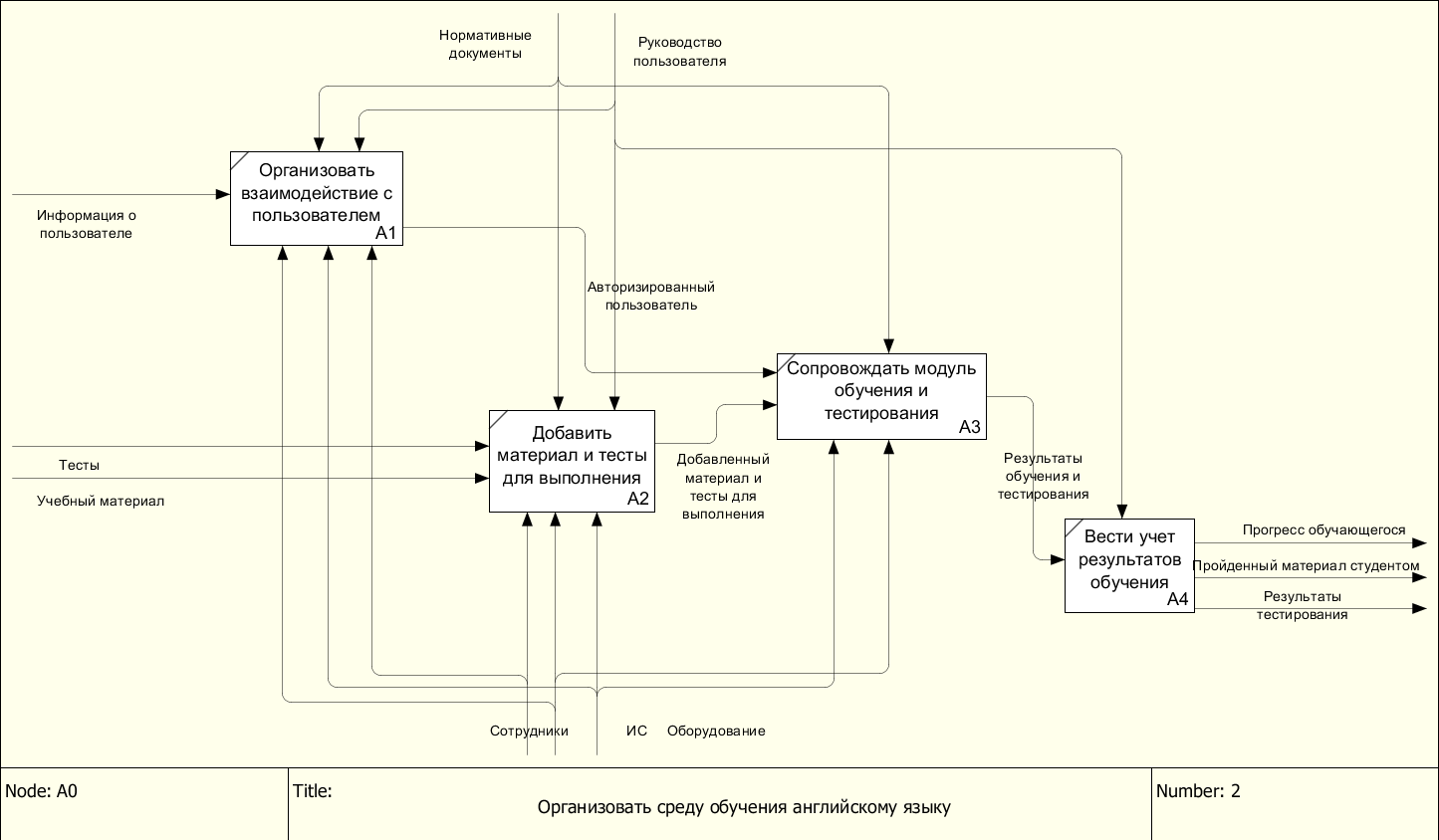
**Рисунок 1 «Контекстная диаграмма»**

**Диаграмма декомпозиция**

Проанализировав деятельность рекламного агентства, и проведя предпроектное исследование, можно выделить три основных бизнес-подпроцесса:

1. Организовать взаимодействие с пользователем;
2. Добавить материал и тесты для выполнения;
3. Сопровождать модуль обучения и тестирования;
4. Вести учет результатов обучения.

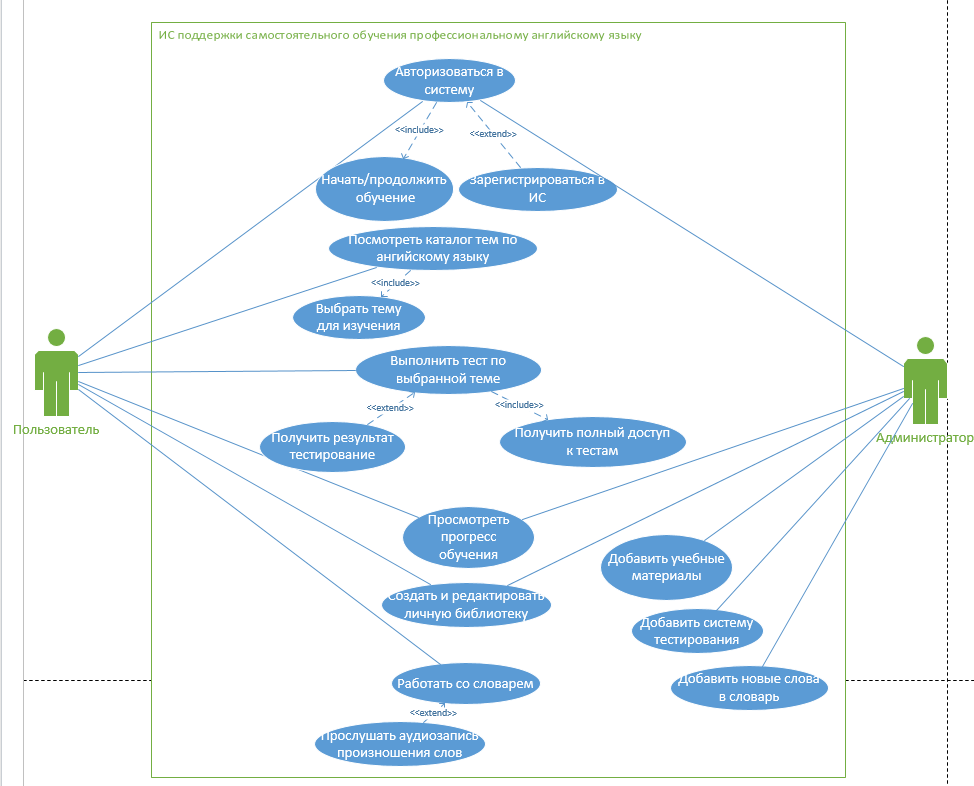
В подпроцесс «Добавить материал и тесты для выполнения» на входе идут учебные материалы и тесты. Это осуществляется при управлении руководством пользователя и нормативных документов и механизмов: сотрудники, ИС и оборудования. На выходе получаем «Добавленный материал и тесты для выполнения», которые входит в третий подпроцесс, «Сопровождать модуль обучения и тестирования». Первый подпроцесс «Организовать взаимодействие с пользователем» также входит в третий подпроцесс. На выходе третьего подпроцесса является «Результаты обучения и тестирования», которые входят в четвертый подпроцесс – «Вести учет результатов обучения». Из четвертого подпроцесса выходит пройденный материал, прогресс и результаты тестирования (Рисунок 2).



**Рисунок 2 «Диаграмма декомпозиции»**

## 1.2 Объектно-ориентированное моделирование

**Диаграмма прецедентов**

Как видно на (Рисунок 3), происходит взаимодействие системы с внешним миром, имеются два основных актера, такие как администратор и пользователь. Пользователь и администратор могут авторизоваться в подсистему. Прецедент «Начать/продолжить обучение» является включением в авторизацию, пользователь сможет продолжить с того место, где он закончил в прошлый раз.

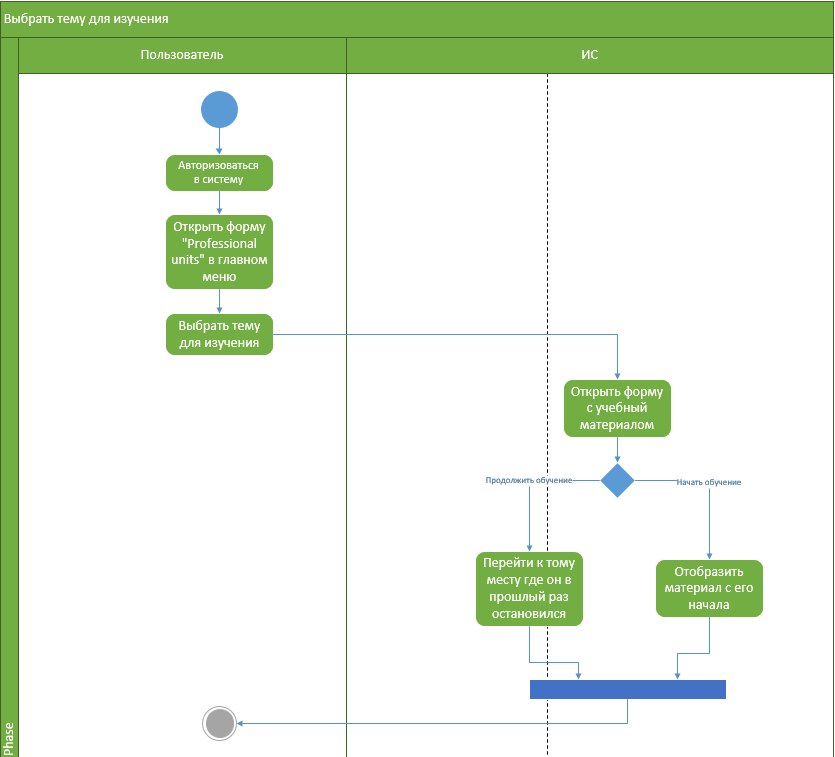
**Рисунок 3 – Диаграмма прецедентов**

Из авторизации расширяется прецедент «Зарегистрироваться в ИС». Далее пользователь может посмотреть каталог тем в это включает выбрать тему для изучения и получение полный доступ к тестам по теме. После этого пользователь может выполнить тест. Так же пользователь может посмотреть прогресс своего обучения в его профиле и работать со словарём.

Администратор может создавать новые учебные материалы и добавлять новые тесты по учебному материалу.

**Диаграммы деятельности**

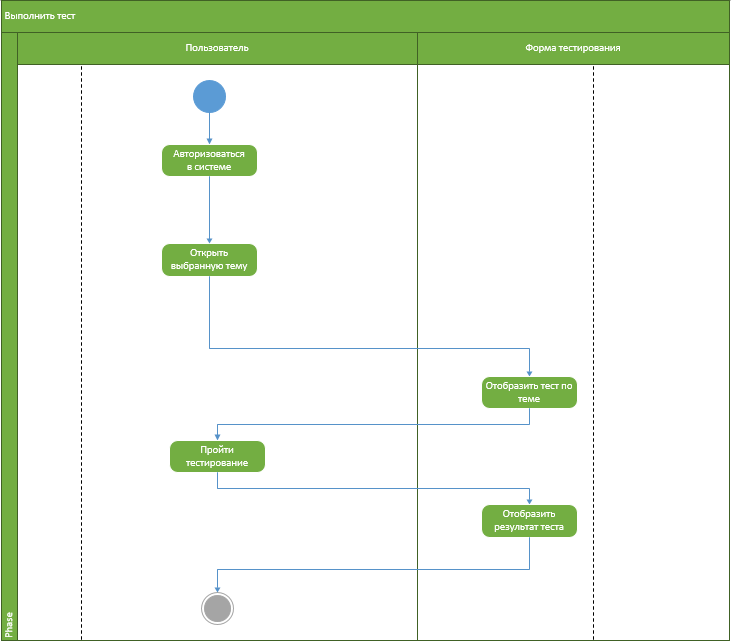
Данная диаграмма деятельности составлена по прецеденту «Выбрать тему для изучения». Существует две дорожки: пользователь и информационная система (ИС). Создается начальный узел на дорожке пользователя, он авторизуется в систему и открывает форму «Professional units» в главном меню. Далее пользователь выбирает тему для изучения, после этого отправляется сообщение дорожки ИС, где происходит действие, которое открывает форму с учебным материалом. Из этого действия идет решение, которое разветвляется на два пути: если пользователь уже начал изучать темы и, если пользователь впервые открыл выбранную тему. Если пользователь уже начал изучать темы, то происходит действие, где он открывает форму на том месте, где он последний раз остановился, а если нет, то происходит действе, где открывается форма с ее начала. Эти два действия входят в соединительный узел, который ведет к конечному узлу на дорожки пользователя (Рисунок 4).



**Рисунок 4 – Диаграмма деятельности по прецеденту**

**«Выбрать тему для изучения»**

Следующая диаграмма деятельности составлена по прецеденту «Выполнить тест». Существует две дорожки: пользователь и информационная система (ИС). Создается начальный узел на дорожке пользователя, он авторизуется в систему и открывает выбранную тему, по которой пользователь хочет пройти тест. Следующие действие — отобразить тест после чего переходит к действие выполнить тест на дорожки пользователь. После выполнение теста отправляется сообщение ИС, где происходит действие, которое отображает результат теста, которое соединяется с конечным узлом на дорожке пользователь (Рисунок 5).

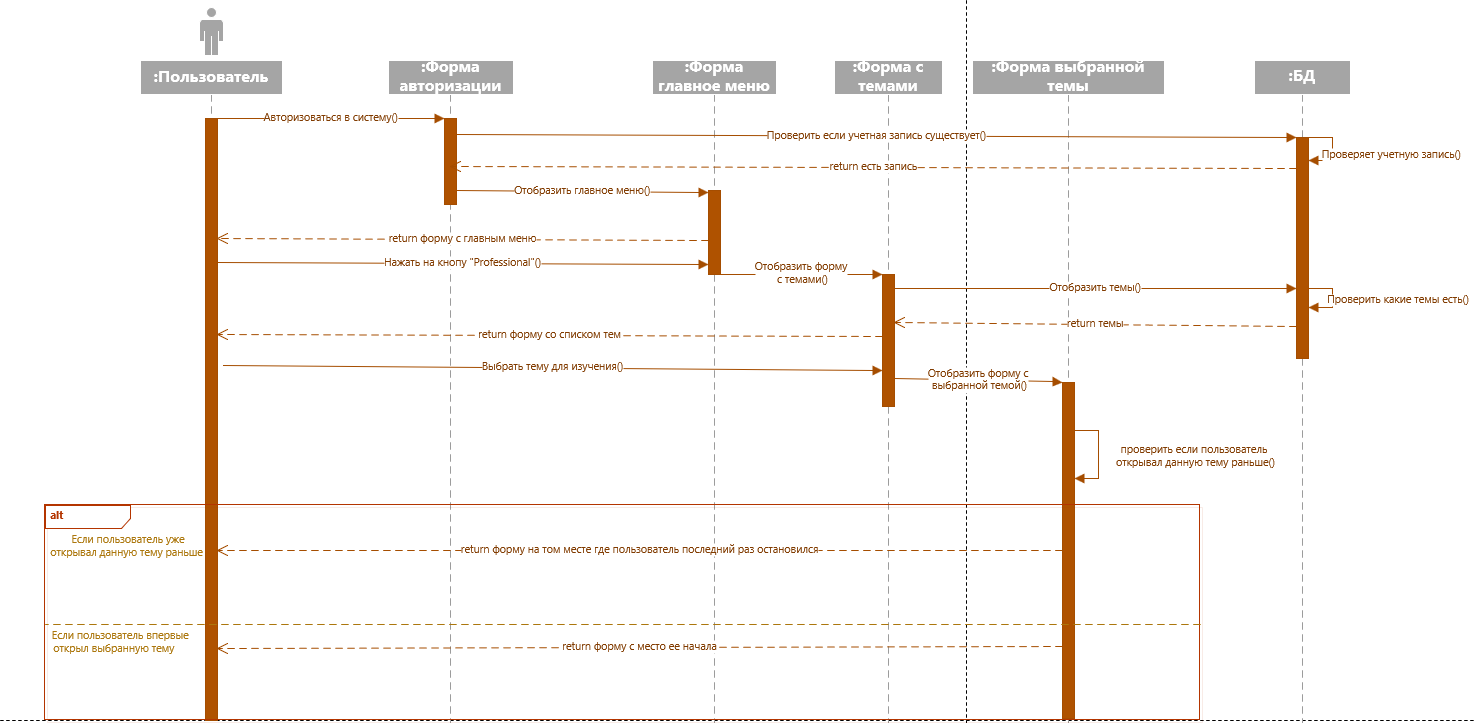


**Рисунок 5 – Диаграмма деятельности по прецеденту**

**«Выполнить тест по выбранной теме»**

**Диаграмма последовательности**

Диаграмму инициирует объект пользователь начинает действие и создается фокус управления, который указывает что происходит в данный момент, в этом случаи он отправляет сообщение форме авторизации: авторизоваться в систему, форма авторизации делает рефлексивное сообщение проверяет если учетная запись существует и создается фокус управления в объекте «:Форма главное меню», которому отправляется сообщение «Отобразить главное меню». Далее «:Форма главное меню» возвращает сообщение с «форма главное меню», после чего пользователь отправляет сообщение «Нажать на кнопку «Professional units»()». После чего объект «:Форма главное меню» отправляет сообщение объекту «:Форма с темами», где создается фокус управления, отобразить форму с темами, что «:Форма с темами» и возвращает пользователю. Далее объект «Пользователь» отправляет сообщение объекту «:Форма с темами», «Выбрать тему для изучения()», после чего объект «:Форма с темами» отправляет сообщение объекту «:Форма выбранной темы», чтобы она отобразила форму с выбранной темой. Объект «:Форма выбранной темы» делает рефлексивное сообщение, проверяет если пользователь открывал данную тему раньше. После чего создается оператор взаимодействия альтернативных фрагментов: фрагмент если пользователь уже открывал данную тему раньше, то возвращает форму на том месте, где пользователь последний раз остановился, и фрагмент если пользователь впервые открыл выбранную тему, то возвращает форму с места ее начала (Рисунок 6).



**Рисунок 6 – Диаграмма последовательности по прецеденту**

**«Выбрать тему для изучения»**

**Диаграмма кооперации**

Понятие кооперации (collaboration) является одним из фундаментальных понятий в языке UML. Оно служит для обозначения множества взаимодействующих с определенной целью объектов в общем контексте моделируемой системы. Цель самой кооперации состоит в том, чтобы специфицировать особенности реализации отдельных наиболее значимых операций в системе. Кооперация определяет структуру поведения системы в терминах взаимодействия участников этой кооперации.

Кооперация может быть представлена на двух уровнях:

* На уровне спецификации - показывает роли классификаторов и роли ассоциаций в рассматриваемом взаимодействии.
* На уровне примеров - указывает экземпляры и связи, образующие отдельные роли в кооперации.

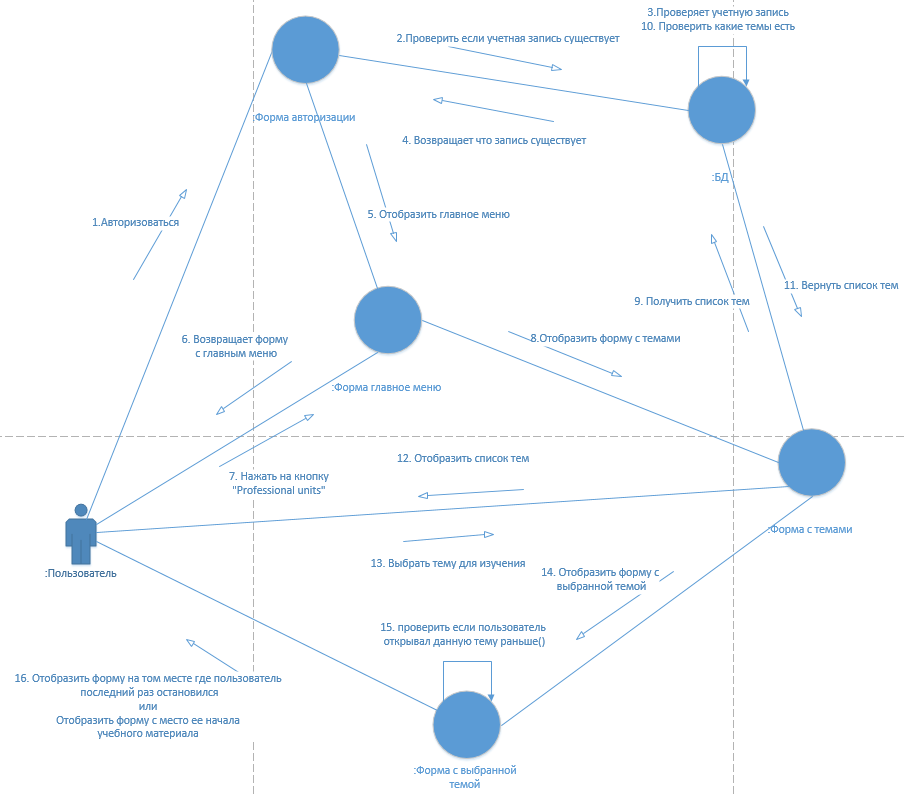
Диаграмма кооперации уровня спецификации показывает роли, которые играют участвующие во взаимодействии элементы. Элементами кооперации на этом уровне являются классы и ассоциации, которые обозначают отдельные роли классификаторов и ассоциации между участниками кооперации.

Диаграмма кооперации уровня примеров представляется совокупностью объектов (экземпляры классов) и связей (экземпляры ассоциаций). При этом связи дополняются стрелками сообщений. На данном уровне показываются только релевантные объекты, т. е. имеющие непосредственное отношение к реализации операции или классификатора.

В кооперации уровня примеров определяются свойства, которые должны иметь экземпляры для того, чтобы участвовать в кооперации. Кроме свойств объектов на диаграмме кооперации также указываются ассоциации, которые должны иметь место между объектами кооперации. При этом вовсе не обязательно изображать все свойства или все ассоциации, поскольку на диаграмме кооперации присутствуют только роли классификаторов, но не сами классификаторы. Таким образом, в то время как классификатор требует полного описания всех своих экземпляров, роль классификатора требует описания только тех свойств и ассоциаций, которые необходимы для участия в отдельной кооперации.

Отсюда вытекает важное следствие. Одна и та же совокупность объектов может участвовать в различных кооперациях. При этом, в зависимости от рассматриваемой кооперации, могут изменяться как свойства отдельных объектов, так и связи между ними. Именно это отличает диаграмму кооперации от диаграммы классов, на которой должны быть указаны все свойства и ассоциации между элементами диаграммы.

Объекты диаграммы: пользователь, форма авторизации, БД, форма главное меню, форма с темами и форма с выбранной темой. Объект «:Пользователь» инициирует диаграмму, он обращается к объекту «Форма авторизации» с сообщением «Авторизоваться». После чего объект «Форма авторизации» отправляет сообщение объекту «БД» чтобы база данных проверила если данные, введенные пользователем, существуют, база данных делает рефлексивное сообщение проверяет существование учетной записи и возвращает сообщение форме авторизации что запись существует. Объект «Форма авторизация» отправляет сообщение объекту «Форма главное меню» чтобы та отобразила форму главного меню, что оно и возвращает объекту «Пользователь». Далее пользователь отправляет сообщение форме с главным меню что он нажимает на кнопку «Professional units» (учебные материалы), после чего форма главного меню отправляет сообщение форме с темами чтобы он отобразил форму с темами. Форма с темами отправляет сообщение базе данных чтобы он отобразил список тем, он проверяет какие темы есть и отправляет их форме с темами после чего этот объект отображает пользователю. Пользователь выбирает тему для изучения на форме с темами после чего форма с темами открывает форму с выбранной темой, которая проверяет на каком месте пользователь остановился и отображает пользователю (Рисунок 7).



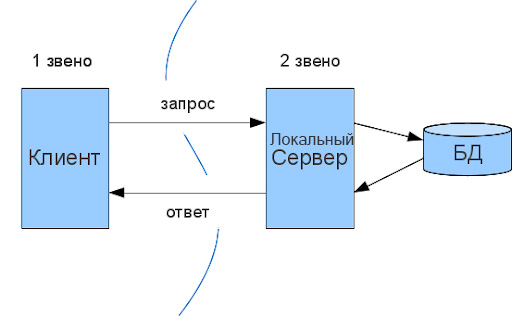
**Рисунок 7 – Диаграмма кооперации по прецеденту**

**«Выбрать тему для изучения»**

# 2. описание реализации программного обеспечения информационной системы

## 2.1 Описание архитектуры программного обеспечения

Программное обеспечение разрабатываемой информационной системы было реализовано в виде клиент-серверного приложения, схема которого показана на рисунке (Номер).



**Рисунок Номер – Клиент-серверная архитектура**

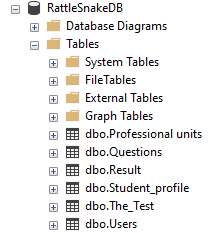
На компьютере каждого пользователя должна быть установлена, разработанная в данной дипломной работе, клиентская программа «Rattlesnake». Данная программа взаимодействует с сервером, установленным в локальной сети, который работает с базой данных «RattleSnakeDB».

## 2.2 Создание базы данных

В качестве системы управления базами данных используется Microsoft SQL Server. MS SQL – это система управления реляционными базами данных (РСУБД). Данная система управления базами данных позволяет хранить большие объемы данных и выполнять различные манипуляции с ними (сохранение, выборка, изменение и удаление и т.п.). Она может использоваться для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

Для выполнения работы с MS SQL используется специальная программа SQL Server Management Studio (SSMS), которая позволяет с помощью графического интерфейса удобно выполнять конфигурирование, управление и администрирование всеми компонентами MS SQL Server. SSMS включает редактор SQL скриптов и графическую программу, которая работает с объектами и настройками сервера. Главным инструментом SSMS является компонент Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также полностью управлять ими.

На основе диаграммы Entity Relationship с помощью программы SSMS была создана база данных «RattleSnakeDB». В состав этой базы данных были включены 6 взаимосвязанных таблиц:



**Рисунок Номер – Скриншот из MS SQL базы данных.(поменять)**

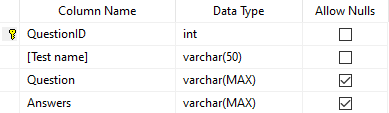
В таблицах базы данных были заданы названия столбцов, тип данных, который может храниться в них и возможность не задавать им значения (тип NULL):

1. Таблица по учетными данными «Professional units»(Рисунок – номер).

**Рисунок Номер – Таблица «Professional units»**

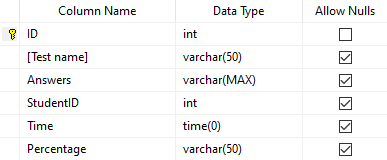


1. Таблица по учетными данными «Questions»(Рисунок – номер).



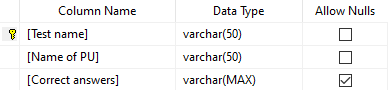
**Рисунок Номер – Таблица «Questions»**

1. Таблица по учетными данными «Result»(Рисунок – номер).



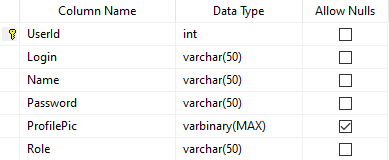
**Рисунок Номер – Таблица «Result»**

1. Таблица по учетными данными «The\_Test»(Рисунок – номер).



**Рисунок Номер – Таблица «The\_Test»**

1. Таблица по учетными данными «Users»(Рисунок – номер).



**Рисунок Номер – Таблица «Users»**

C помощью диаграммы базы данных между таблицами были заданы взаимосвязи, которые показаны на рисунке номер.

## 2.2 Разработка программного кода

Клиентская часть программного обеспечения была разработана на языке C# с использованием среды разработки Microsoft Visual Studio 2019. Программа имеет графический интерфейс, разработанный с использованием технологии Windows Forms, и взаимодействует с MS SQL сервером с использованием технологии Entity Frameworks ADO.NET Code first from database.

В программе разработано Номер форм.

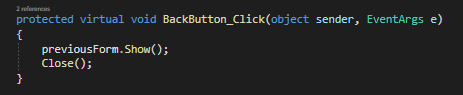
**Форма «basic Form»**

Это форма является базовой формой, которая наследуется во всех остальных формах. У нее стоят все свойства, которые наследуется в других формах, это цвет текста, цвет фона, цвет фона панели и шрифт текста.

**Рисунок Номер – Базовая форма «basicForm»**

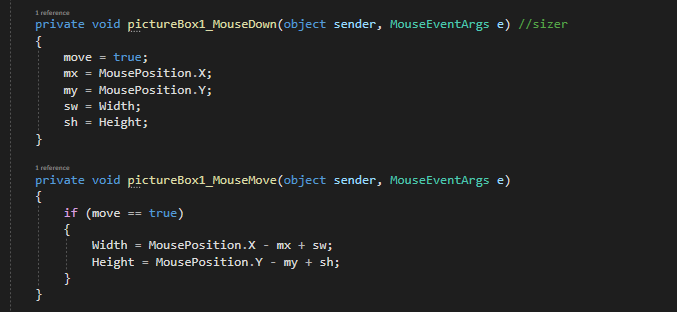


PictureBox в левом верхнем углу это иконка приложения, при нажатии на которую переходит в меню в зависимости от роли пользователя. Так же есть lable1 текст, которого вручную заполняется на других формах, но расположение и цвет текста, и шрифт переноситься в другие формы. В нижнем левом углу есть кнопка назад, которая закрывает данную форму и открывает предыдущую. Событие представлено на рисунке номер.



**Рисунок Номер – События кнопки назад «BackButton»**

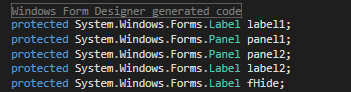
Так же в правом нижнем углу есть PictureBox изменение размера формы (sizer), при нажатии на него задается переменным sw (Ширина формы после нажатие на sizer), sh (Высота формы после нажатие на sizer), mx (X координат формы после нажатие на sizer), my (Y координат формы после нажатие на sizer) и move (Переменная проверяющая если движется мышка). Потом при движении мышки срабатывает событие pictureBox1\_MouseMove, где проверяется если переменная move равняется true то задается ширине формы (Width) позиция мышки в данном моменте из которого вычитается значение переменной mx и прибавляется значение sw. Это заставляет форму расширятся. Код событий представлен на рисунке номер.



**Рисунок Номер – События pictureBox1\_MouseDown и pictureBox1\_MouseMove**

Для того, чтобы производные формы, созданные на основе данной базовой формы, могли изменять свойства ЭУ (например, задавать свойство Text для текстовых меток или делать кнопки управлением формы) у этих элементов нужно изменить режим доступа с private (закрытый – доступный только в базовом классе) на protected (защищенный – доступный и в базовом и в производных классах).

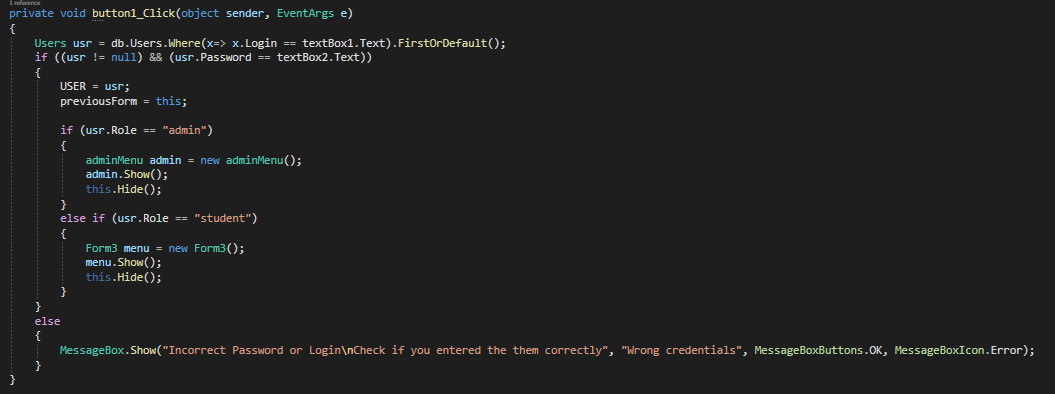
И в нижней части файла у всех ЭУ заменим режимы доступа с private на режимы доступа protected (Рисунок номер):



**Рисунок номер– Designer базовой формы**

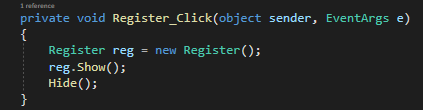
**Форма «Login»**

Данная форма представляет из себя форму авторизации в систему. На ней расположены текстовые поля для ввода логина и пароля и рядом расположены подписи полей, определяющие что нужно вводить. Так же в нижнем правом углу существует кнопка «Login», при нажатии которой совершается событие button1\_Click(), в котором создается локальный класс User и ему присваивается пользователя которой потом ищется в базе данных с помощью модели Entity framework используя событие Where в скобках, которых я задаю значение textBox1.Text(текст поля заполнения логина), потому что это ключевое поле таблицы. Дальше программа проверяет если этот пользователь существует в базе данных и, если текст в textBox2 (поля для ввода пароля) совпадает с паролем из базы данных.Если проверка проходит, то переменной previousForm, которая является типом Form присваивается значение формы Login и переменной USER (тип данных User). Дальше идет проверка на если роль найденного пользователя равняется роль админа, то открывается форма меню админа и скрывается данная форма, а если роль равняется роли студента, то открывается форма меню студента и скрывается данная форма. Код формы можно увидеть на приведенном ниже рисунке (Рисунок номер).



**Рисунок номер– Designer базовой формы**

В форме логина еще есть кнопка регистрации, которая открывает форму регистрации (Рисунок номер).



**Рисунок номер– Событие кнопки «Register»**

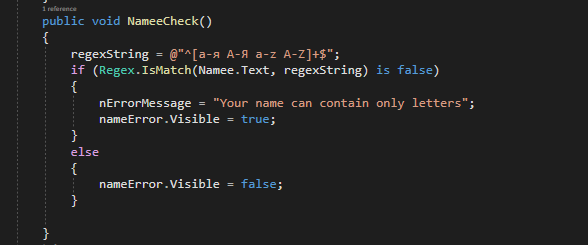
**Форма «Register»**

В форме регистрации есть поля для заполнения имени, логина, пароля и проверка на совпадение паролей.

Для каждого из полей существуют проверки на правильность заполнения события проверки правильность ввода данных с помощью Regex. Используя namespace System.Text.RegularExpressions.

Создается строковая переменная regexstring, в которой записываются какие символы проверять надо. Далее проверяется условие, в котором используется библиотека Regex и событие IsMatch в ней, в скобках которой прописывается какой текст нужно проверить и переменная regexString (символы которых мы проверяем наличие). Если мы не находим проверяемые символы, то в nameError (label, в котором есть индикация на то, что ошибка в заполнении текстового поля есть) меняется свойство Visible на false, значит, что этого label не будет на экране, означающие то, что ошибок при заполнении полей нет. А если обнаруживается символы, то в свойстве Visible nameError задается значение true это значит, что он появляется на форме и при наведении на него мышки будет выдаваться в чем ошибка заполнения, которая присваивается toolTip1 (Рисунок номер).

**Форма «Menu»**

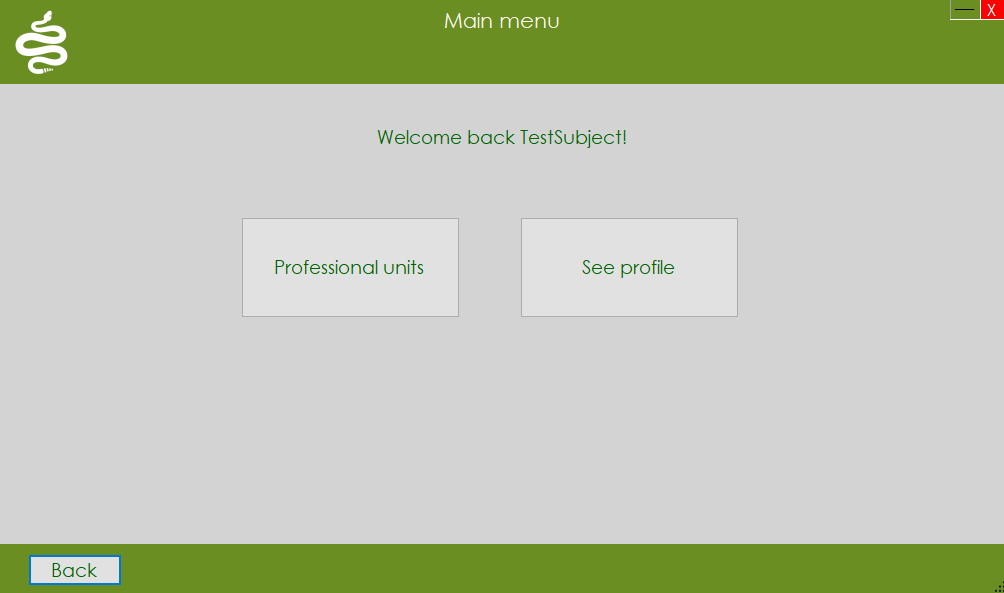


**Рисунок номер – Одна из проверок: событие проверки имени**

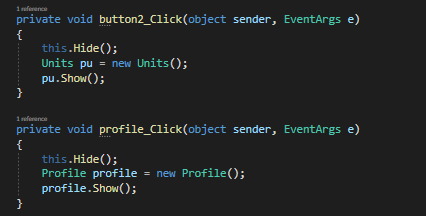
Данная форма — это главное меню для пользователей с ролью «student».

На ней есть кнопки: «Professional units» и «See profile». Кнопка «Professional units» скрывает форму «Menu» и открывает форму «Units» там, где находится учебный материал для изучения, а кнопка «See profile» открывает форму «Profile» где можно посмотреть результаты пройденных тестов и можно посмотреть информацию пользователя и если нужно, то поменять её.

Так же если пользователь с ролью «student» только что авторизовался в систему, это первая форма, которую он видит и появляется надпись, которая приветствует его (Рисунок номер).



**Рисунок номер – Форма «Menu»(меню студента)**

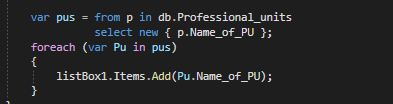


**Рисунок номер – Код событий кнопок «Professional units» и «See profile»**

События кнопок «Professional units» и «See profile» представлены на рисунке номер.

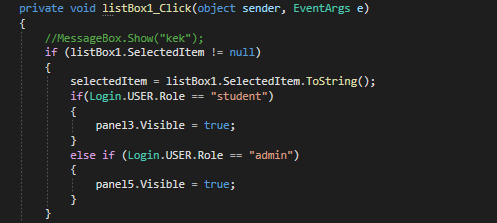
**Форма «Units»**

Форма «Units» предоставляет пользователю список всех существующих учебных материалов, которые берутся из базы данных и добавляется в lisBox, добавленный в дизайнере. Для нахождения всех названий учебных материалов используется LinQ запрос (Рисунок номер).



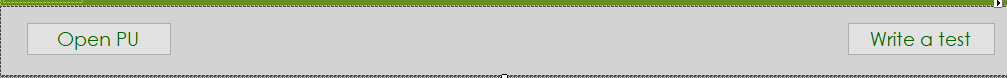
**Рисунок номер – Запрос на выгрузку из базы данных названий учебных материалов**

Эта форма используется для пользователей с ролью «student» и «admin», только отображаются разные вещи при нажатии один раз на учебный материал. Используется оператор условий if, который проверяет роль пользователя и отображает ему панель с доступными ему функциями (Рисунок номер).



**Рисунок номер – Код событий при выборе учебного материала**

Панель panel3 для студентов содержит в себе кнопки «Open PU», который открывает выбранный учебный материал и кнопка «Write a test», которая открывает форму с тестом по выбранному материалу (Рисунок номер1).



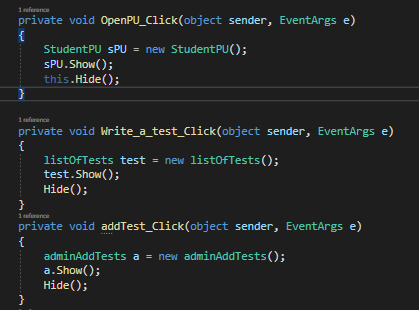
**Рисунок номер1 – Панель panel3(для студентов)**

Панель panel5 для администратора содержит в себе кнопку «Add a Test», которая открывает форму «adminAddTests» (Рисунок номер2).



**Рисунок номер2 – Панель panel5(для администратора)**

События кнопок «Open PU», «Write a test» и «Add a Test» представлены на рисунке номер.



**Рисунок номер2 – События кнопок «Open PU», «Write a test» и «Add a Test»**