**федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ» (КНИТУ-КАИ)**

**ОТДЕЛЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ДИПЛОМНЫЙ ПРОЕКТ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Тема: Разработка приложения для изучения русского языка, как иностранного, для испаноговорящих .** | | | | |
| **Руководитель**  Преподаватель |  |  |  | Гимадеев И.Ф. |
| *(должность)* |  | *(подпись)* |  | *(И.О. Фамилия)* |
| **Обучающийся**  4432 |  |  |  | Валова П.А. |
| *(группа)* |  | *(подпись)* |  | *(И.О. Фамилия)* |
| **Специальность** 09.02.07 Информационные системы и программирование . | | | | |
| (шифр и наименование специальности) | | | | |

Казань 2022 г.

**АННОТАЦИЯ**

Данный документ представляет собой пояснительную записку к выпускной квалификационной работе по направлению обучения 09.02.07 «Информационные системы и программирование» на тему «Разработка приложения для изучения русского языка, как иностранного, для испаноговорящих».

Пояснительная записка состоит из: количество страниц – \_\_; количество приложений – \_\_; количество иллюстраций – \_\_; количество таблиц – \_\_; кол-во использованных источников – \_\_.

Целю дипломного проекта является обучение испаноговорящих русскому языку, как иностранному, с помощью теоретических и практических занятий.

Данная информационная система предусмотрена для образовательных учреждений, в которых осуществляется подготовка к поступлению в университет.

Результат – автоматизированная информационная система “Calanthe” для изучения русского языка, как иностранного, для испаноговорящих.

Ключевые слова: desktop приложение, обучение, Windows Presentation Foundation, Microsoft SQL Server, Visual Studio.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc104661419)

[РАЗДЕЛ 1. ПОСТАНОВКА ЗАДАЧИ 6](#_Toc104661420)

[1.1 Цели и задачи информационной системы 6](#_Toc104661421)

[1.2 Описание предметной области 6](#_Toc104661422)

[РАЗДЕЛ 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 7](#_Toc104661423)

[2.1 Выбор средств реализации 7](#_Toc104661424)

[2.1.1 Язык программирования 7](#_Toc104661425)

[2.1.2 Среда разработки 8](#_Toc104661426)

[2.1.3 Платформа 9](#_Toc104661427)

[2.1.4 Система управления базами данных 10](#_Toc104661428)

[2.2 Выявление требований к программному обеспечению 10](#_Toc104661429)

[2.3 Проектирование диаграммы состояний 14](#_Toc104661430)

[2.4 Проектирование диаграммы деятельности 15](#_Toc104661431)

[2.5 Проектирование диаграммы классов 16](#_Toc104661432)

[2.6 Проектирование диаграммы последовательностей 17](#_Toc104661433)

[2.7 Проектирование диаграммы сущность-связь 20](#_Toc104661434)

[2.9 Проектирование базы данных 22](#_Toc104661435)

[2.10 Разработка пользовательского интерфейса 22](#_Toc104661436)

[РАЗДЕЛ 3. ПОЛЬЗОВАТЕЛЬСКАЯ ДОКУМЕНТАЦИЯ 29](#_Toc104661437)

[3.1 Руководство пользователя 29](#_Toc104661438)

[3.2 Руководство программиста 29](#_Toc104661439)

[3.3 Руководство по установке ПО 29](#_Toc104661440)

[ЗАКЛЮЧЕНИЕ 30](#_Toc104661441)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31](#_Toc104661442)

[ПРИЛОЖЕНИЕ 1. Листинг программы 32](#_Toc104661443)

[ПРИЛОЖЕНИЕ 2. Список сокращений 33](#_Toc104661444)

[ПРИЛОЖЕНИЕ 3. Презентация 34](#_Toc104661445)

# ВВЕДЕНИЕ

Число иностранных студентов увеличивается с каждым годом, но, к сожалению, обучения на подготовительных курсах, которые предоставляют образовательные учреждения, не достаточно.

Во время обучения русскому языку иностранцы встречаются с такими трудностями, как: старые обучающие материалы, не имеющие визуальных решений; плохая коммуникация с преподавателями русского языка из-за лингвистического барьера. Как следствие, иностранные студенты поступают в образовательные учреждения, но не получают информацию в полной объёме, из-за чего, чаще всего, у них появляется множество академических задолженностей или они понимают, что не справляются с данным объемом учебного материала, и покидают страну.

Именно для решения проблем, описанных выше, было принято решение создать информационную систему “Calanthe”, позволяющую иностранцам, говорящим на испанском языке, ознакомиться с базой русского языка и научиться активно применять полученные знания в жизни.

Для решения поставленной цели было решено реализовать следующие модули:

1. Регистрации;
2. Авторизации;
3. Просмотра теоретических занятий;
4. Просмотра практических занятий;
5. Редактирования профиля;
6. Просмотра статистики:
7. Ведения словаря.

Разделы дипломного проекта, их название и описание:

1. Раздел 1 “Постановка задачи”: этот раздел содержит в себе формулировку решаемой задачи.
2. Раздел 2 “Проектирование информационной системы”: в этой главе определяется обобщенная архитектура программного обеспечения (пакета программ, системы или модуля). Описывается назначение выделенных блоков и связи между ними.
3. Раздел 3 “Пользовательская документация”: в данный раздел входят следующие руководства: руководство по установке программного обеспечения, руководство пользователя, руководство программиста.

# РАЗДЕЛ 1. ПОСТАНОВКА ЗАДАЧИ

* 1. **Цели и задачи информационной системы**
  2. **Описание предметной области**

# РАЗДЕЛ 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

**2.1 Выбор средств реализации**

### **2.1.1 Язык программирования**

C# - это язык программирования, который Microsoft сделал, взяв лучшее из языков C и C++, и продолжил добавлять к нему функциональные возможности, заимствуя из других языков, таких как java. Он ориентировал его для всей своей платформы .NET (как Framework, так и Core), и со временем адаптировал средства создания кода, которые имели еще один из его самых популярных языков, Visual Basic, что сделало его таким же универсальным и простым в освоении.

Как мы уже говорили ранее, C# это язык, предназначенный для использования на .NET, целью которой является создание приложений простым способом. Таким образом, этот язык используется для разработки приложений на этой платформе.

Рассмотрим особенности этого ЯП:

1. Простота: C# не содержит элементы, которые не нужны в. NET;
2. Современность: Этот ЯП автоматически и интуитивно включает в свой язык элементы, которые необходимы для разработки приложения;
3. Безопасность: встроенный механизм для обеспечения правильного доступа к типам данных, что позволяет избежать ошибок, которые трудно обнаружить;
4. Расширяемость: возможность добавлять основные типы данных, операторы и модификаторы;
5. Совместимость: C# поддерживает синтаксис, очень похожий на C, C++, Java и многие другие языки программирования, чтобы облегчить работу программиста;
6. Эффективный: несмотря на ограничения, которые C# имеет по всему коду, вы можете пропустить эти ограничения, используя объекты с помощью указателей.

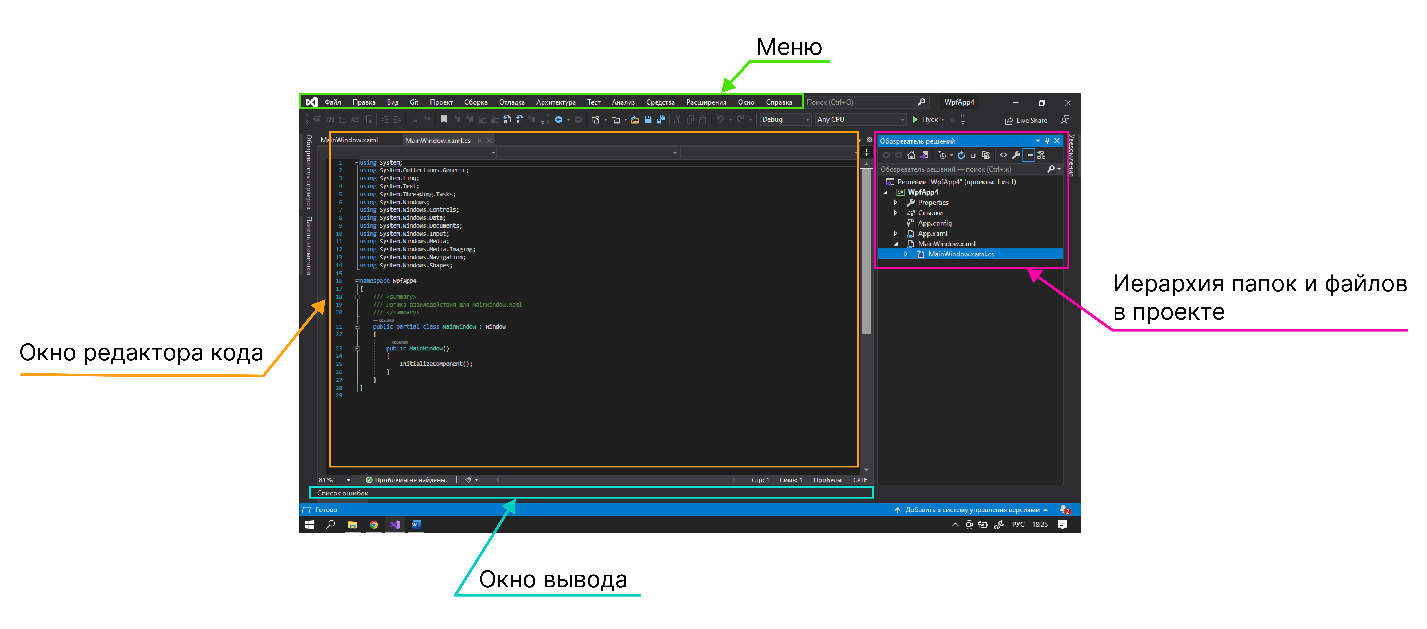
Для работы с C# рекомендуется использовать Microsoft Visual Studio, поскольку она была разработана специально для этого и поддерживает нагрузку и работу с тем же языком.

### **2.1.2 Среда разработки**

Visual Studio – это интегрированная среда разработки для просмотра и редактирования любого типа кода после отладки, сборки и публикации Android, iOS, Windows, веб-приложений и облачных приложений. VS cозданна компанией Microsoft и доступна для операционных систем Windows, Linux и macOS, и в то же время он совместим с несколькими языками программирования, такими как C++, C#, Visual Basic .NET, F#, Java, Python, Ruby и PHP, а также что среды веб-разработки, такие как ASP.NET.

VS - Это многофункциональная программа, которую можно использовать для многих аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, предоставляемых большинством IDE, Visual Studio включает в себя компиляторы, инструменты завершения кода, графические дизайнеры и многие другие функции, упрощающие процесс разработки программного обеспечения.

Краткое описание пользовательского интерфейса Visual Studio показано на рисунке 1.

****Рисунок 1 – Пользовательский интерфейс Visual Studio

### **2.1.3 Платформа**

WPF - это сокращение от Windows Presetation Foundation. Это часть платформы разработки. NET. Но не просто часть, она самая большая из всех. Он охватывает многие концепции от пользовательских интерфейсов с XAML до шаблонов проектирования, таких как MVVM.

С точки зрения программирования, это серия сборок и инструментов .NET Framework. Он предназначен для предоставления API (Интерфейс прикладного программирования) для создания богатых и сложных пользовательских интерфейсов для Windows. Он поддерживается от Windows XP до последней версии Windows, версии 10.

А так же WPF включает в себя новые функции, которые делают его предшественника, Windows Forms, устаревшим. Например, поддержка 3D, расширенная типография и PDF-подобные документы и многое другое.

Рассмотрим наиболее важные функции WPF:

1. Декларативный графический интерфейс: WPF позволяет создавать пользовательские интерфейсы с использованием языка разметки под названием XAML;
2. Динамический дизайн;
3. Векторная графика: графики в WPF основаны на векторах. В отличие от графики на основе растровых изображений, векторную графику можно масштабировать без деформаций;
4. Шаблоны: мы можем создавать повторно используемые элементы для графического интерфейса;
5. Стили: стили в WPF очень похожи на каскадные таблицы стилей, CSS, для HTML. В этом случае стили более мощные, чем CSS. Мы можем обрабатывать любые графические функции, которые вы можете себе представить, поля, пространства, цвета, размеры и многое другое.

### **2.1.4 Система управления базами данных**

Microsoft SQL Server - это система управления реляционными базами данных (СУБД), которая поддерживает широкий спектр приложений для обработки транзакций, бизнес-аналитики и аналитики в корпоративных вычислительных средах. Microsoft SQL Server является одной из трех ведущих технологий баз данных на рынке.

Как и другие технологии СУБД, SQL Server в основном построен вокруг структуры таблиц на основе строк, которая соединяет связанные элементы данных в разных таблицах друг с другом, избегая необходимости избыточного хранения данных в нескольких местах в базе данных. Реляционная модель также обеспечивает ссылочную целостность и другие ограничения целостности для поддержания точности данных.

Основным компонентом Microsoft SQL Server является механизм базы данных SQL Server, который контролирует хранение, обработку и безопасность данных. Включает реляционный механизм, который обрабатывает все команды и запросы, и механизм хранения, который управляет файлами базы данных, таблицы, страницы, индексы, буферов данных и транзакций. Механизм базы данных также создает и выполняет хранимые процедуры, триггеры, представления и другие объекты базы данных.

**2.2 Выявление требований к программному обеспечению**

Для начала выявим все пользовательские требования для данного программного продукта (Таб. 2.1).

Таблица 2.1 - Пользовательские требования

|  |  |
| --- | --- |
| **№** | **Пользовательское требование** |
| 1 | Возможность регистрации пользователя |
| 2 | Возможность авторизации пользователя |
| 3 | Возможность восстановления пароля |
| 4 | Возможность просмотра личных данных |
| 5 | Возможность просмотра статистики |
| 6 | Возможность редактирования личных данных |
| 7 | Возможность просмотра теоретических занятий |
| 8 | Возможность прохождения практических занятий |
| 9 | Возможность просмотра словаря |
| 10 | Возможность редактирования словаря |

Далее необходимо выполнить сокращение этих требований:

Шаг 1: Все требования, представленные выше, относятся к первоначальной цели разработки проекта, поэтому сокращение происходить не должно;

Шаг 2: Теперь следует распределить их по категориям (Таб. 2.2);

Таблица 2.2 - Распределение требований по категориям

|  |  |
| --- | --- |
| **Категория** | **Список требований** |
| «должно быть включено» – если эти требования не будут включены, не будет достигнута цель | Возможность регистрации пользователя |
| Возможность авторизации пользователя |
| Возможность просмотра теоретических занятий |
| Возможность прохождения практических занятий |
| Возможность просмотра словаря |
| Возможность редактирования словаря |
| «желательно включить» – если эти требования не будут включены, цели не будут реализованы в полном объеме | Возможность просмотра личных данных |
| Возможность редактирования личных данных |
| «хорошо бы включить» – это полезные требования, но они не способствуют достижению первоначальных целей | Возможность восстановления пароля |
| Возможность просмотра статистики |
| «нужно отклонить» – это требования, которые не соответствуют первоначальным целям | - |

Шаг 3: Приведем список требований в соответствие с возможностями: никакие из требований не нуждаются в удалении;

Шаг 4: Требования категории «должно быть включено» исключаться не будут, так как план проекта остался неизменным;

Шаг 5: В таблице 2.3 представлен перечень оставшихся пользовательских требований, которая будет служить основой для контроля изменений. Данный перечень войдет в спецификацию требований к разрабатываемому ПО.

Таблица 2.3 – Пользовательские требования

|  |  |
| --- | --- |
| **№** | **Пользовательское требование** |
| 1 | Возможность регистрации пользователя |
| 2 | Возможность авторизации пользователя |
| 3 | Возможность восстановления пароля |
| 4 | Возможность просмотра личных данных |
| 5 | Возможность просматривать статистику |
| 6 | Возможность редактирования личных данных |
| 7 | Возможность просмотра теоретических занятий |
| 8 | Возможность прохождения практических занятий |
| 9 | Возможность просмотра словаря |
| 10 | Возможность редактирования словаря |

Шаг 6: Отклоненные требования необходимо сохранить для будущих обзоров как нереализованные потребности.

Теперь мы должны выделить акторов, взаимодействующих с системой. В данной системе будет только два актора – гость и обучающийся.

Описание акторов предоставлено в Таблице 2.4.

Таблица 2.4 – Описание акторов

|  |  |
| --- | --- |
| **Акторы** | **Краткое описание** |
| Гость | Неопределенный пользователь системы, желающий воспользоваться её возможностями |
| Обучающийся | Физическое лицо, осваивающее образовательную программу |

Далее необходимо описать варианты использования, выявленные выше. Они указаны в Таблице 2.5.

Таблица 2.5 – Выявленные варианты использования

|  |  |  |
| --- | --- | --- |
| **Основной актор** | **Наименование** | **Формулировка** |
| Гость | Авторизация | Пользователь с помощью почты и пароля может войти в систему |
| Гость | Регистрация | Пользователь с помощью логина, почты и пароля может зарегистрироваться в системе |
| Обучающийся | Восстановления пароля | Пользователь может восстановить пароль, введя электронную почту, которую он указывал при регистрации |
| Обучающийся | Просмотр личных данных | Пользователь может просматривать свои личные данные в личном кабинете |
| Обучающийся | Редактирование личных данных | Пользователь может просматривать свои личные данные в личном кабинете |
| Обучающийся | Просмотр статистики | Пользователь может просматривать свою статистику |
| Обучающийся | Возможность просмотра теоретических занятий | Пользователь может просматривать теоретические занятия |
| Обучающийся | Возможность прохождения практических занятий | Пользователь имеет возможность проходить практические занятия |
| Обучающийся | Возможность просмотра словаря | Пользователь просматривать свой словарь |
| Обучающийся | Возможность редактирования словаря | Пользователь может осуществлять ведение словаря |

Исходя из выявленных вариантов использования для акторов на рисунке 2.1 была представлена диаграмма вариантов использования.

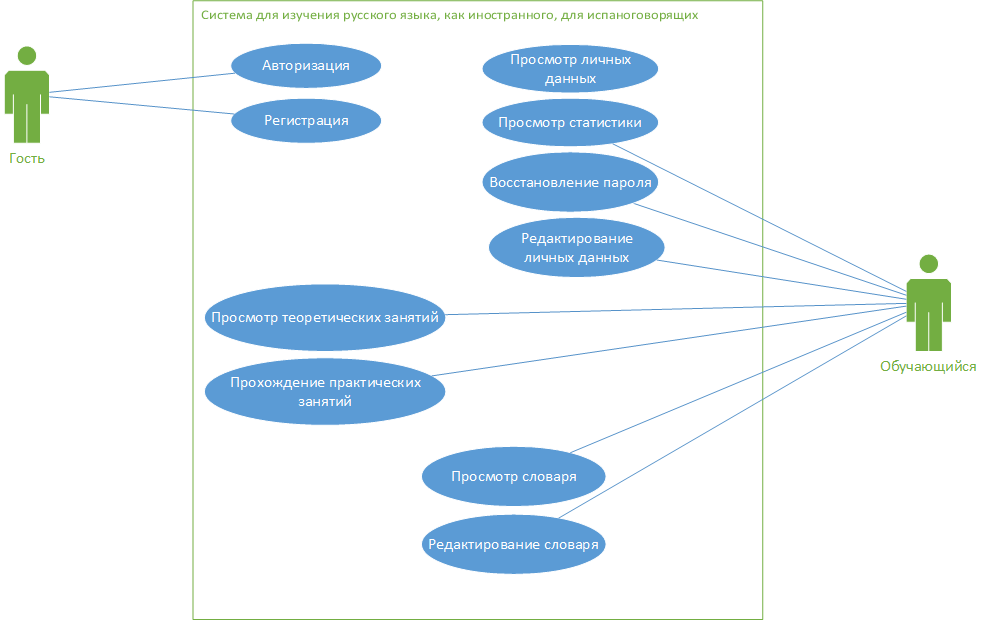


Рисунок 2.1 – Диаграмма вариантов использования

**2.3 Проектирование диаграммы состояний**

Данная диаграмма необходима для того, чтобы описать, каким образом работают объекты, а так же она показывает, как объект переходит из одного состояния в другое.

Построим диаграмму состояния для актора «Обучающийся» (Рис.2.2).

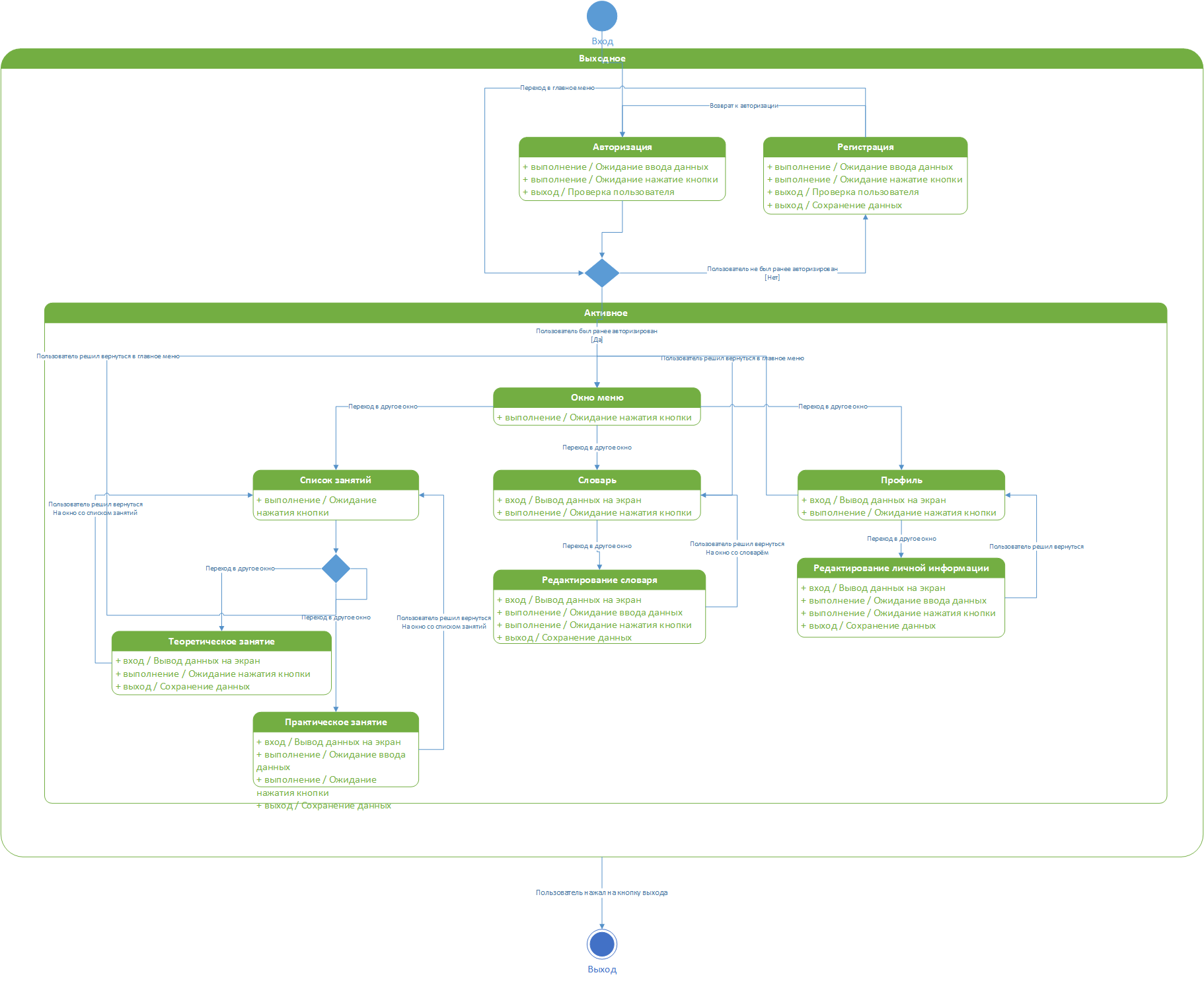


Рисунок 2.2 – Диаграмма состояний для актора «Обучающийся»

**2.4 Проектирование диаграммы деятельности**

Диаграмма деятельности содержит в себе представление рабочих процессов поэтапных действий и действий с поддержкой выбора, итерации и параллелизма.

Данная диаграмма показана на Рисунке 2.3.

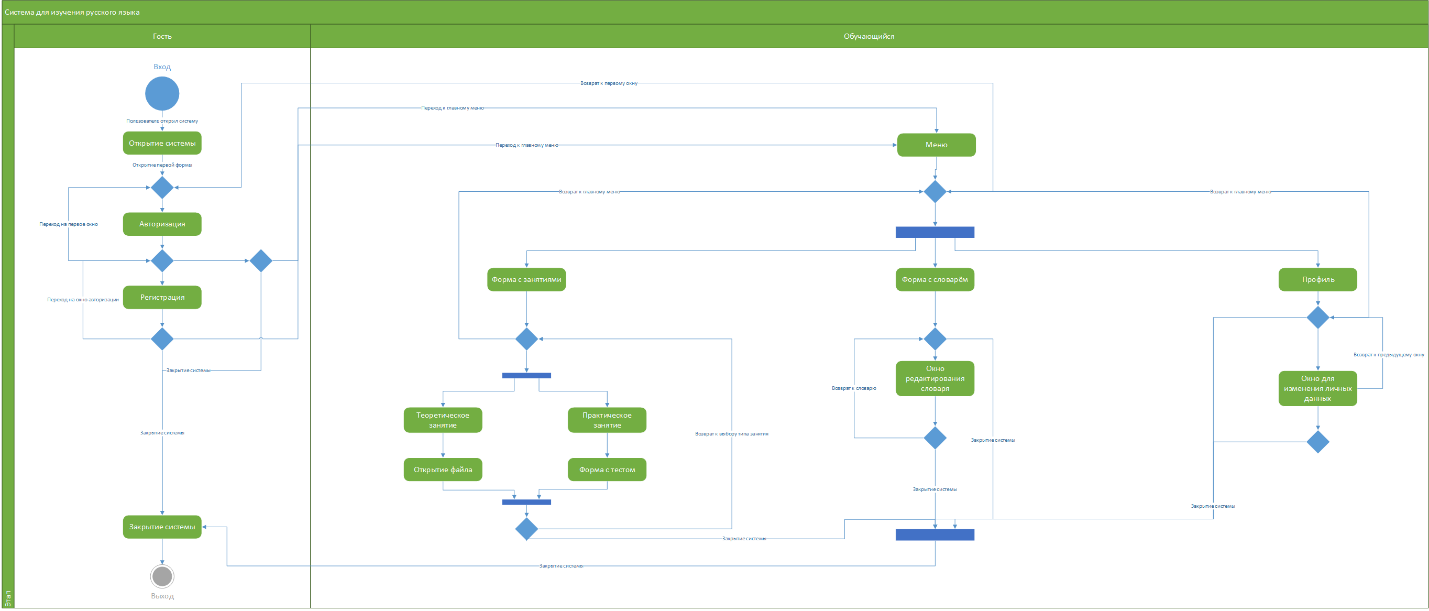


Рисунок 2.3 – Диаграмма деятельности

## **2.5 Проектирование диаграммы классов**

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования.

Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений.

Создадим диаграмму классов (Рис. 2.4):

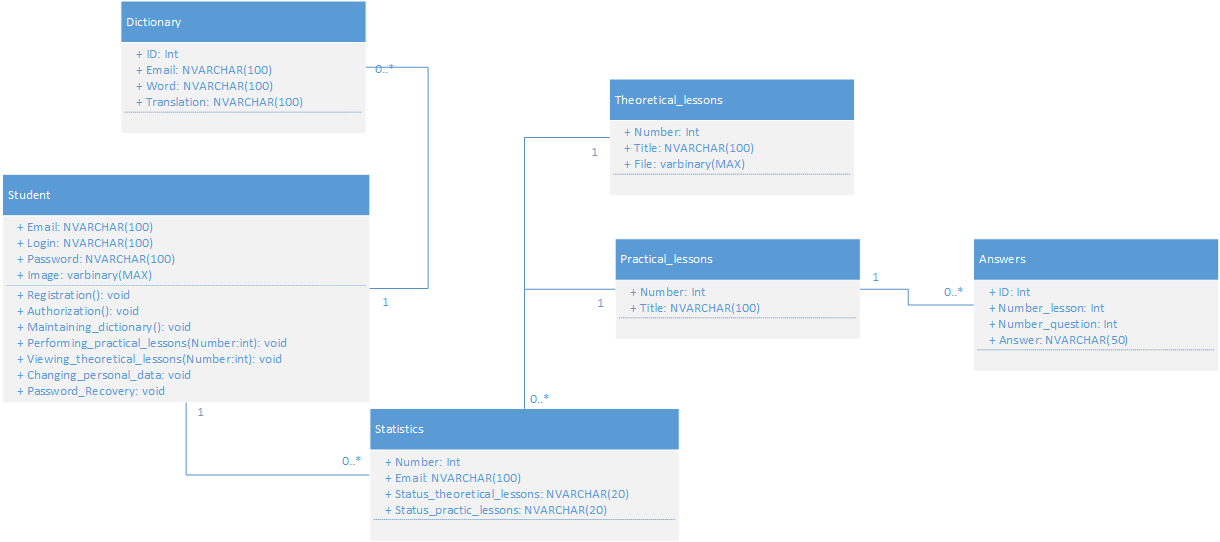


Рисунок 2.4 – Диаграмма классов

## **2.6 Проектирование диаграммы последовательностей**

Эта диаграмма описывает отношения объектов в различных условиях. Условия взаимодействия задаются сценарием, полученным на этапе разработки диаграмм вариантов использования.

Диаграмма последовательности содержит в себе: линии жизни, бары активации, сообщения.

Для данной ИС были созданы диаграммы последовательностей для следующих прецедентов: «Авторизация» (Рис. 2.5), «Регистрация» (Рис. 2.6), «Просмотр занятий» (Рис. 2.7), «Восстановление пароля» (Рис. 2.8), «Редактирование личных данных» (Рис. 2.9).

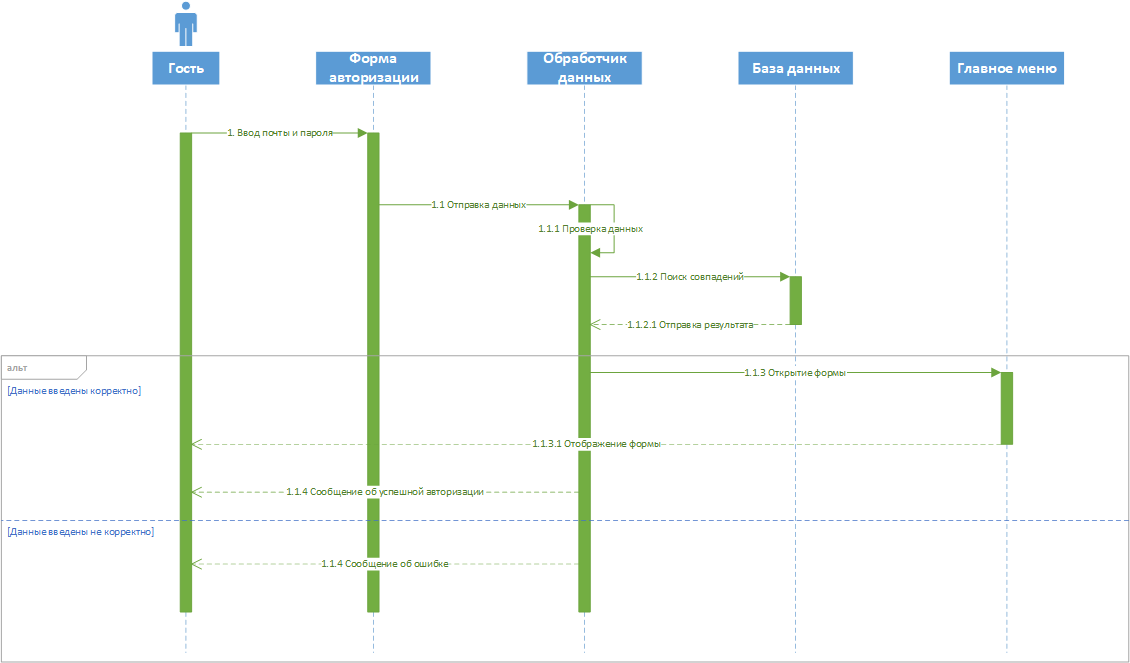


Рисунок 2.5 – Диаграмма последовательности для прецедента «Авторизация»

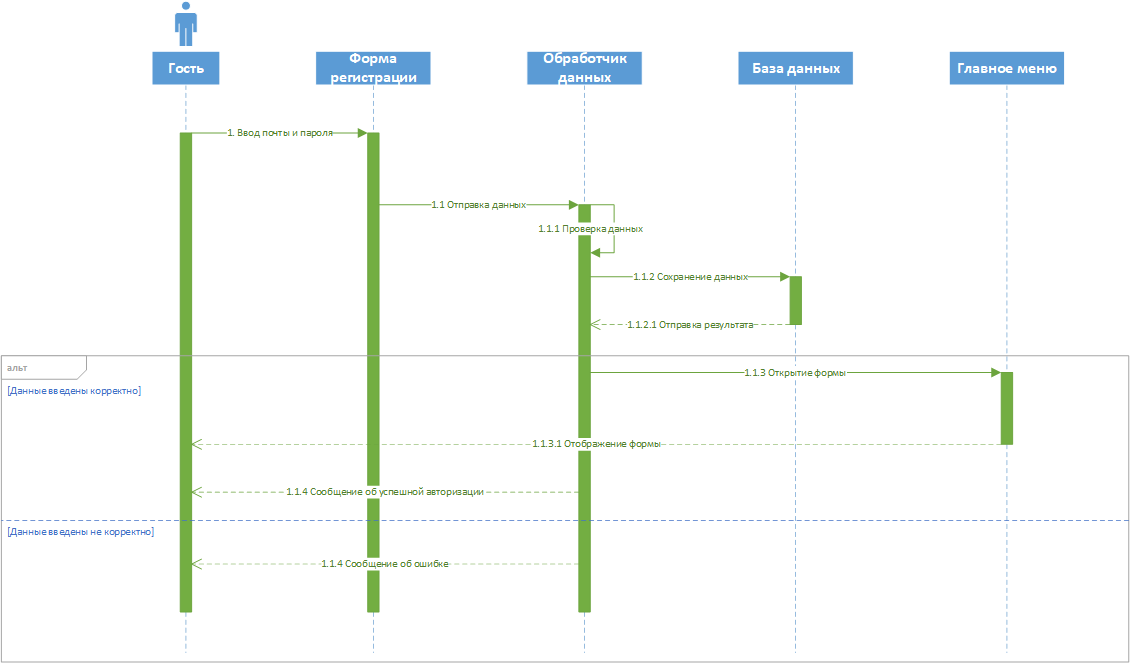


Рисунок 2.6 – Диаграмма последовательности для прецедента «Регистрация»

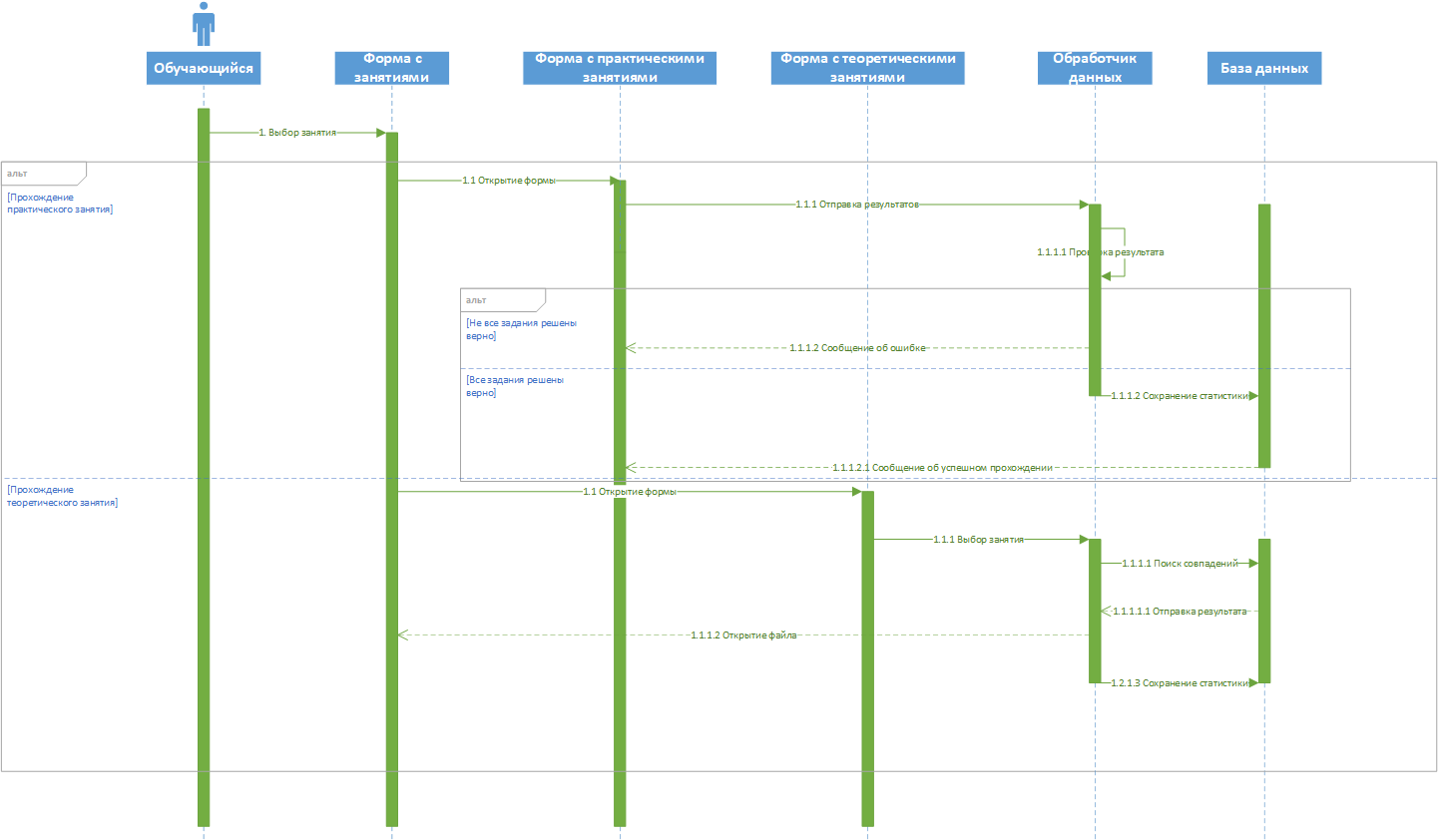


Рисунок 2.7 – Диаграмма последовательности для прецедента «Просмотр занятий»

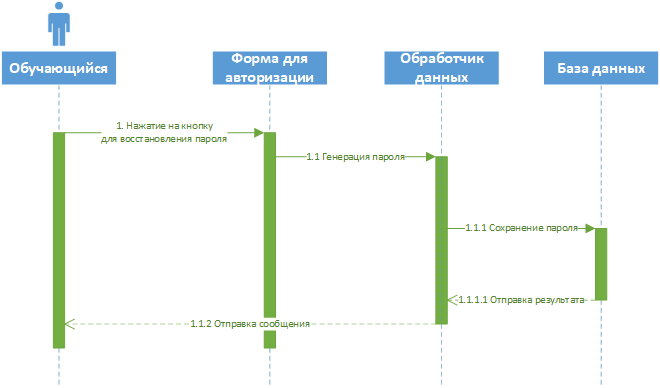


Рисунок 2.8 – Диаграмма последовательности для прецедента «Восстановление пароля»

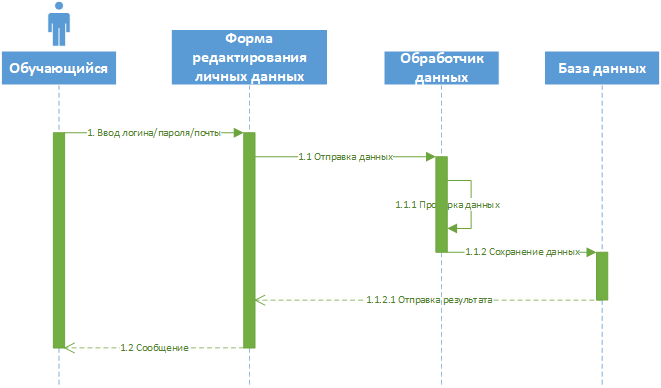


Рисунок 2.9 – Диаграмма последовательности для прецедента «Изменение личных данных»

## **2.7 Проектирование диаграммы сущность-связь**

Диаграмма сущность-связь предназначена для моделей данных и обеспечивает стандартный способ определения данных и отношений между ними.

Для нашей ИС данная диаграмма будет выглядеть так (Рис. 2.10):

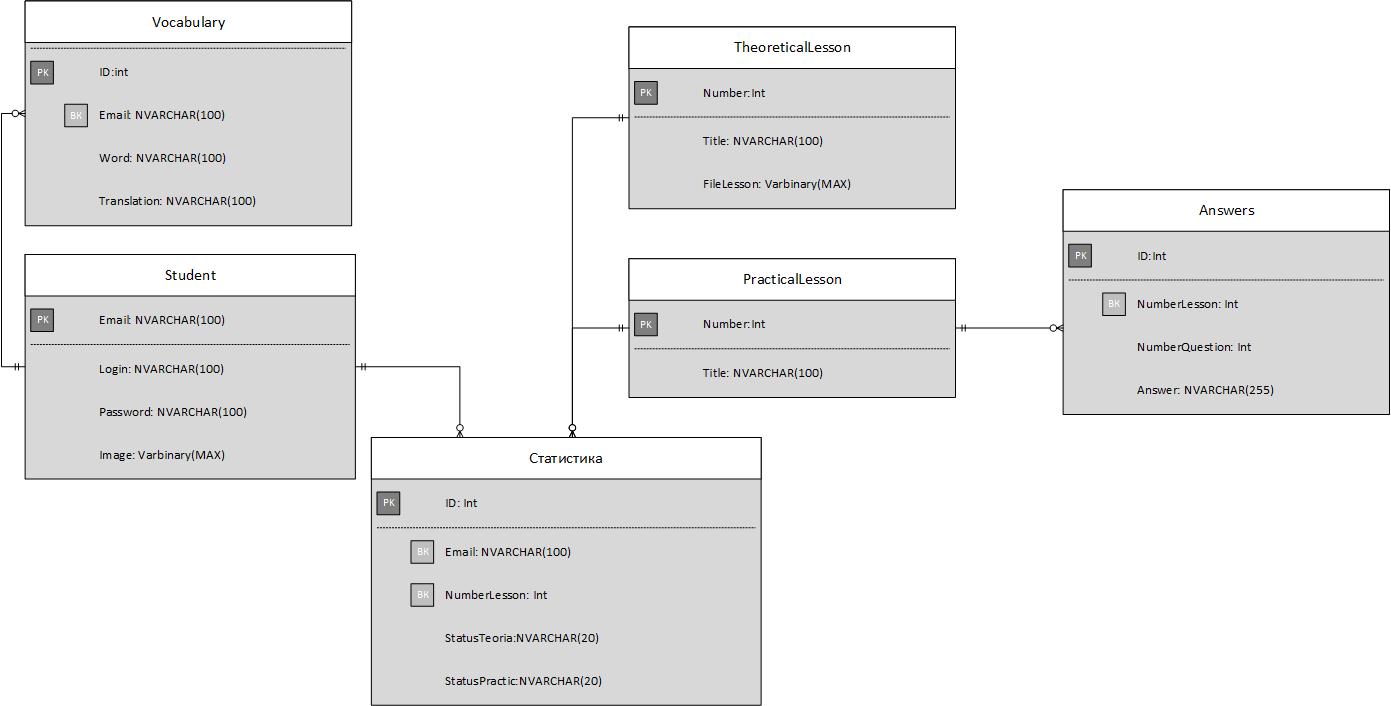


Рисунок 2.10 – Диаграмма сущность-связь

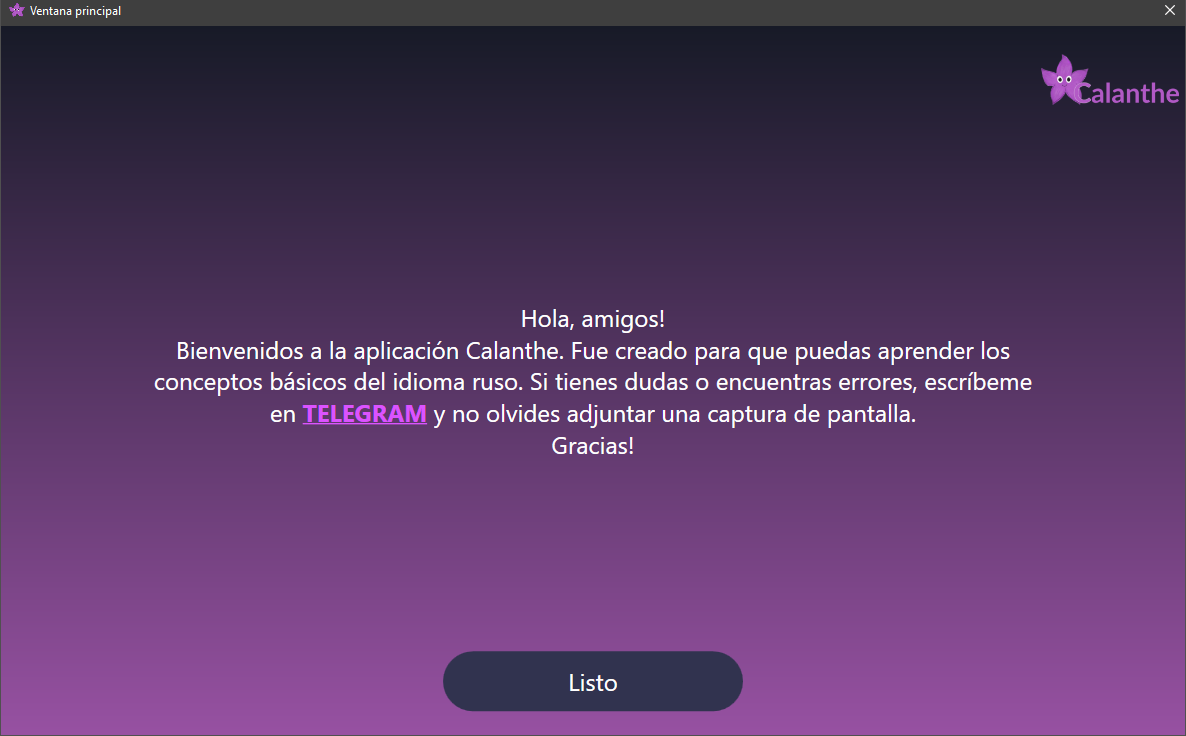
А так же, опираясь на диаграмму сущность-связь, показанную на рисунке 2.10, составим словарь данных (Рис. 2.11):

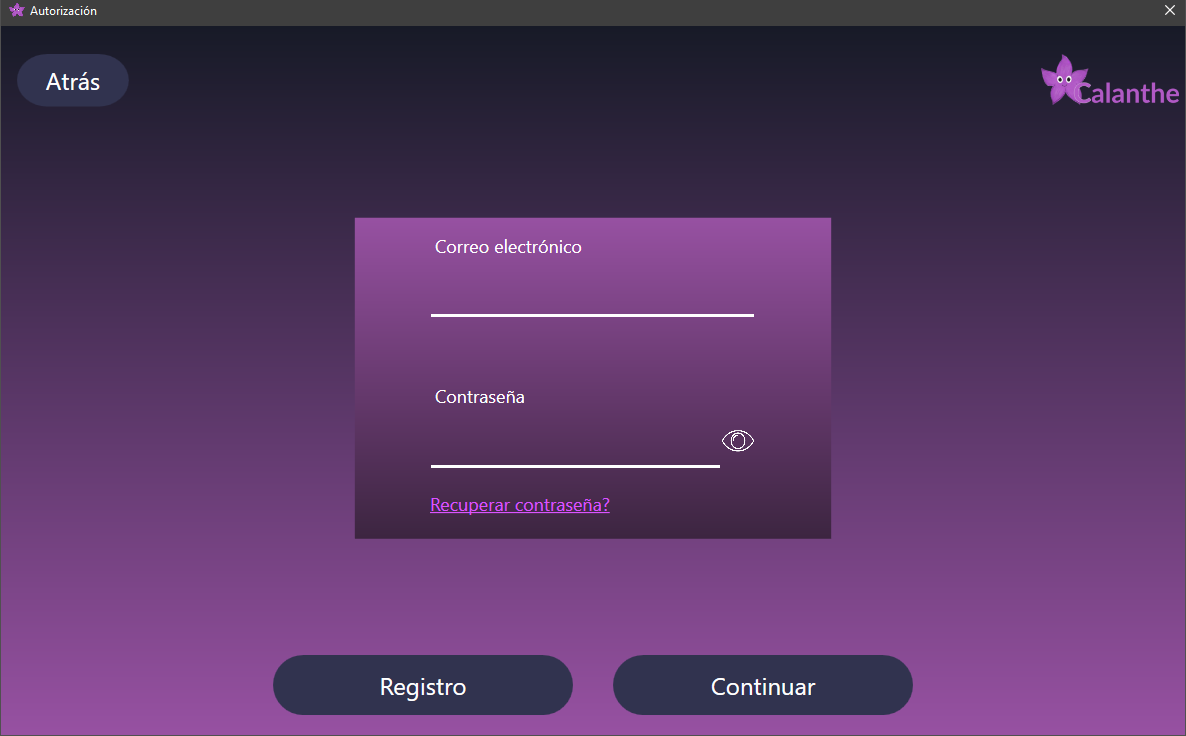


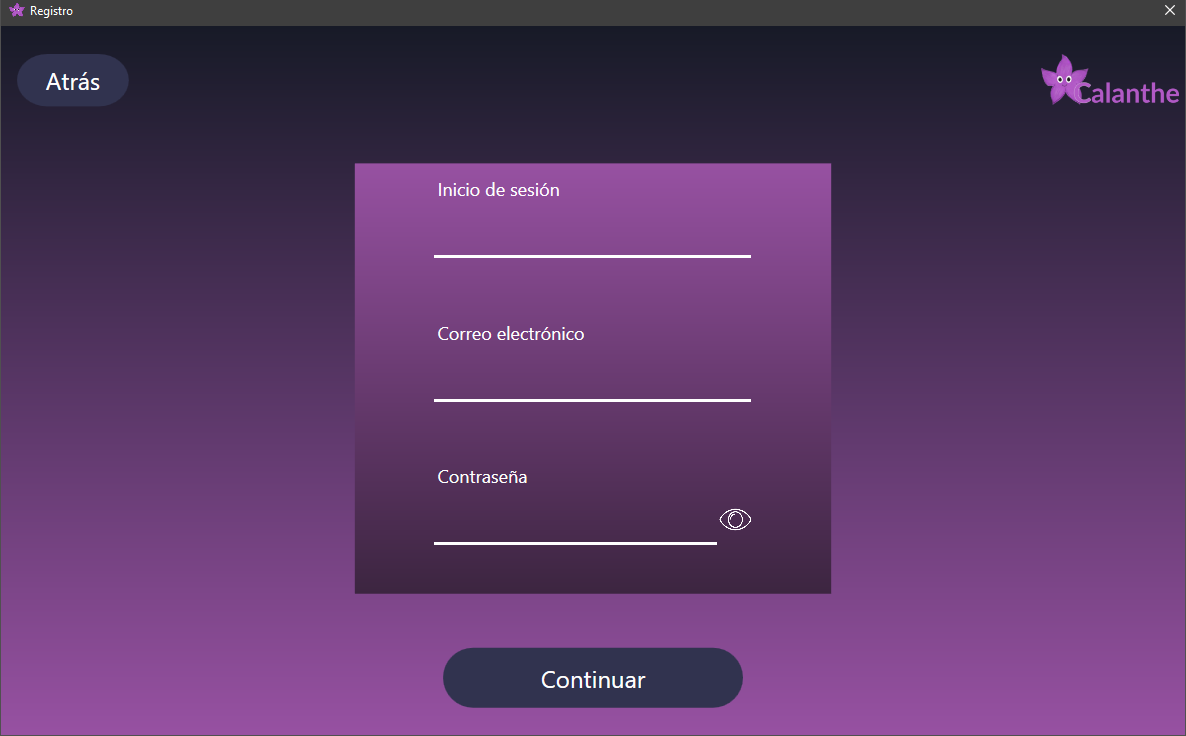
Рисунок 2.11 – Словарь данных

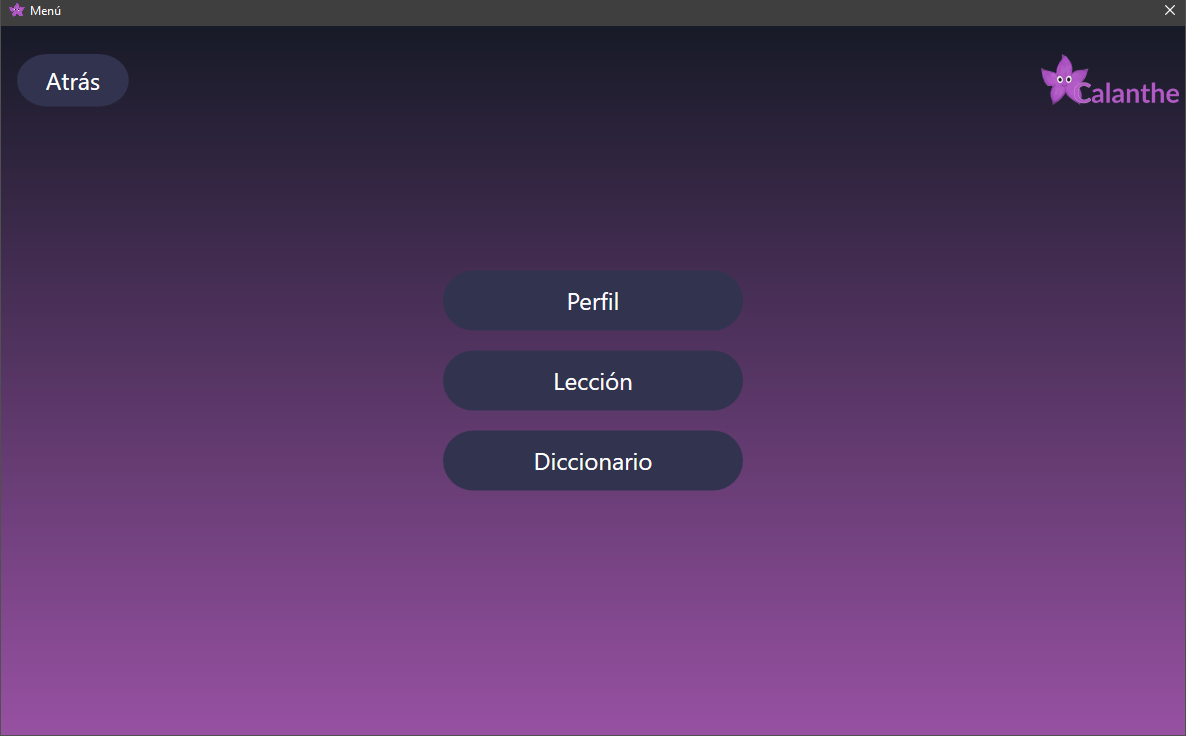
## **2.9 Проектирование базы данных**

## **2.10 Разработка пользовательского интерфейса**



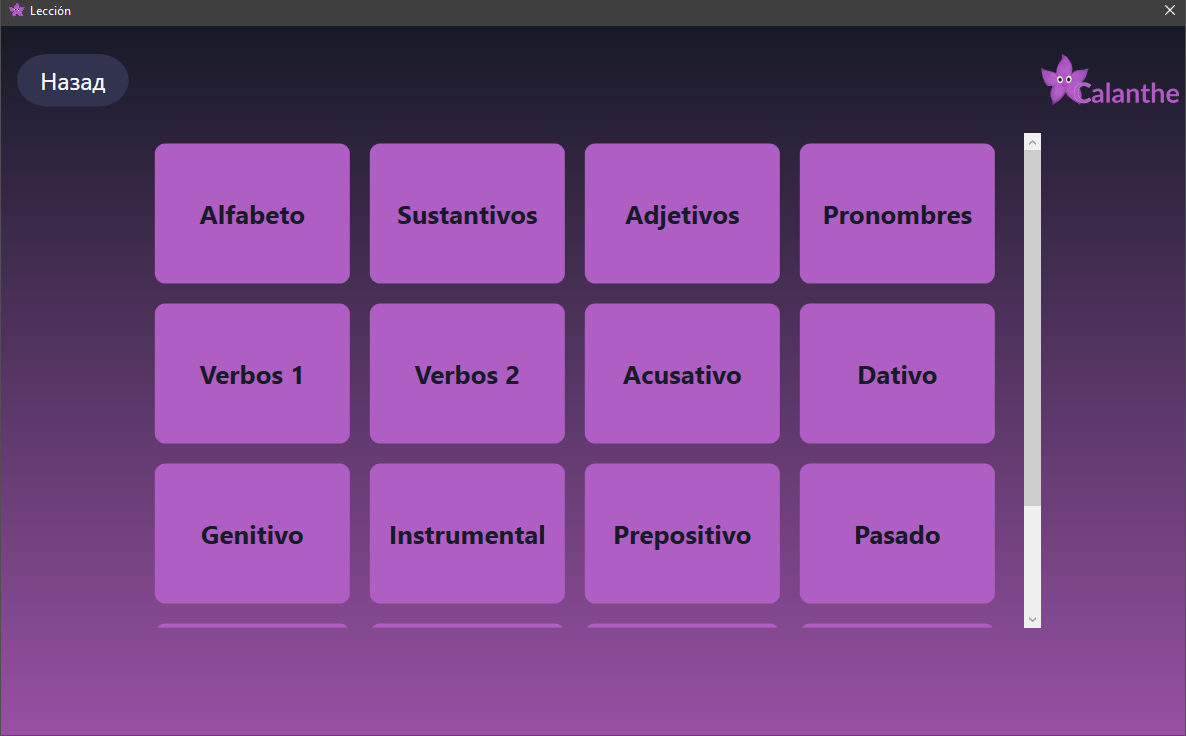


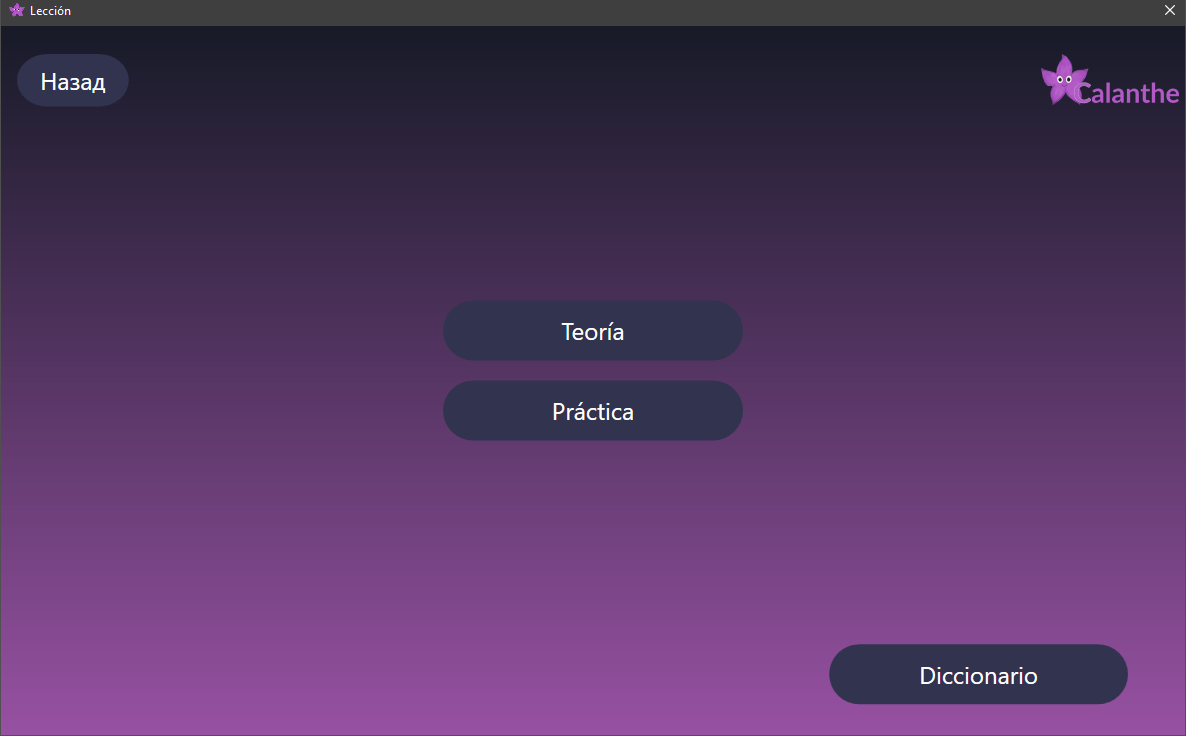


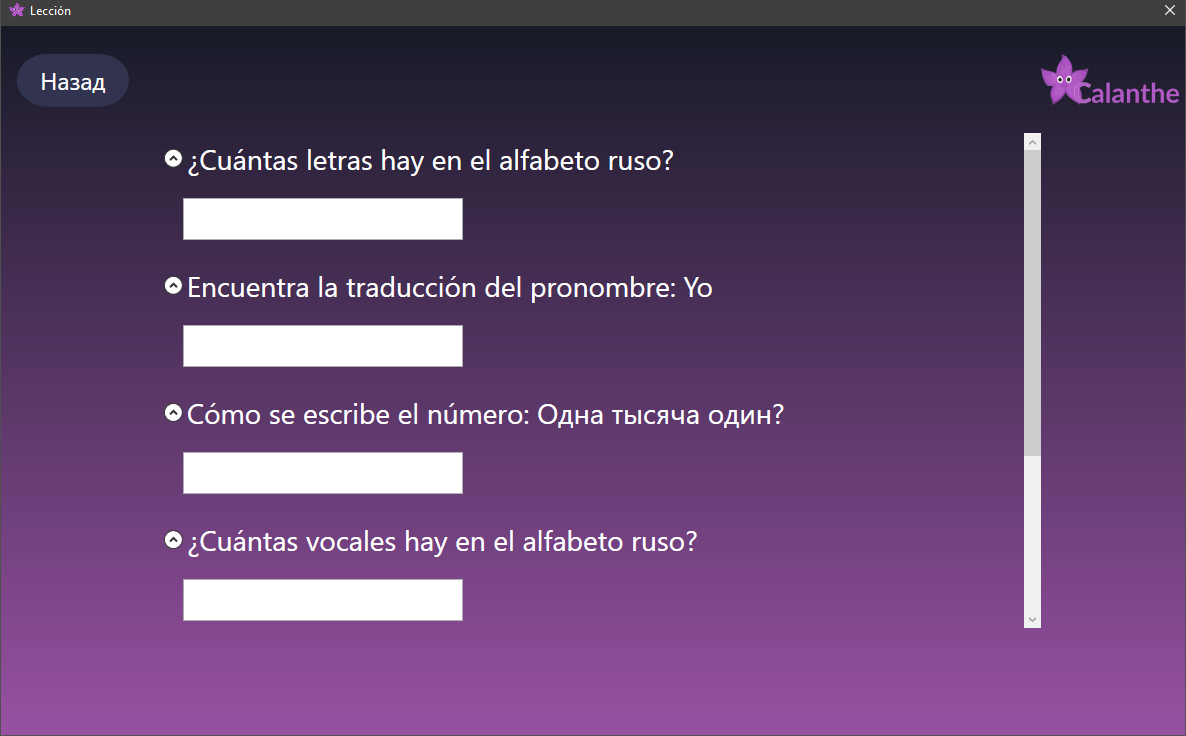


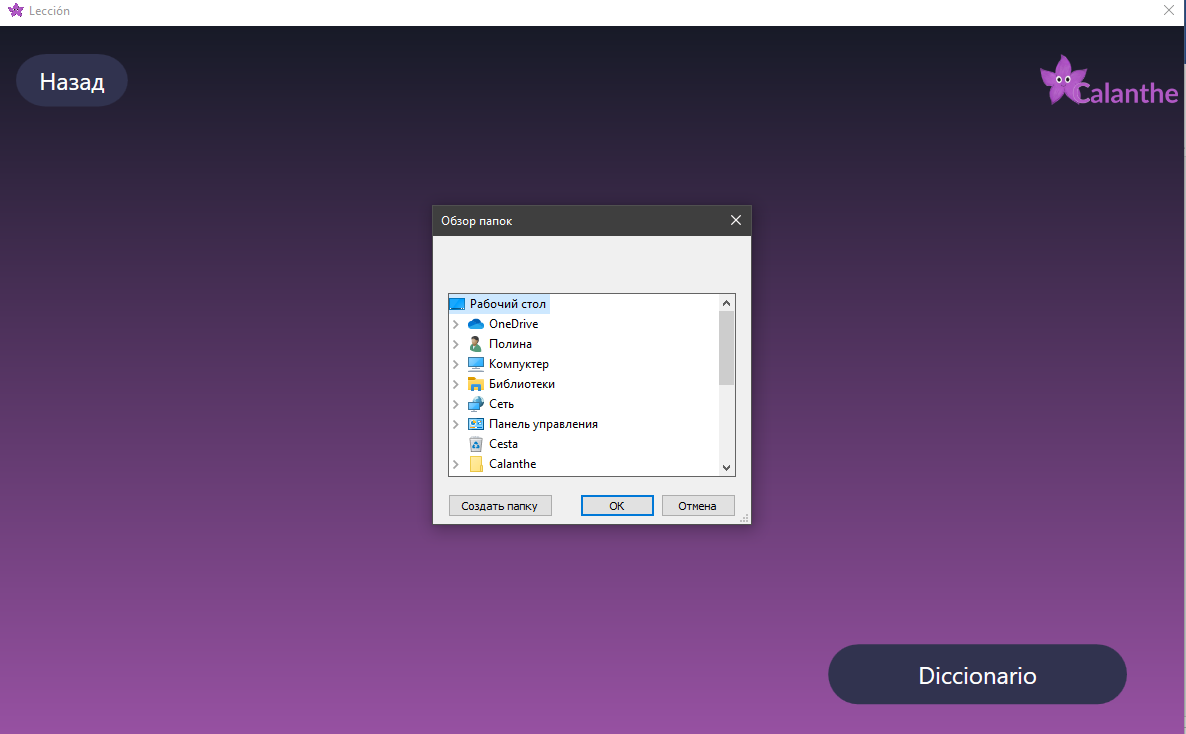


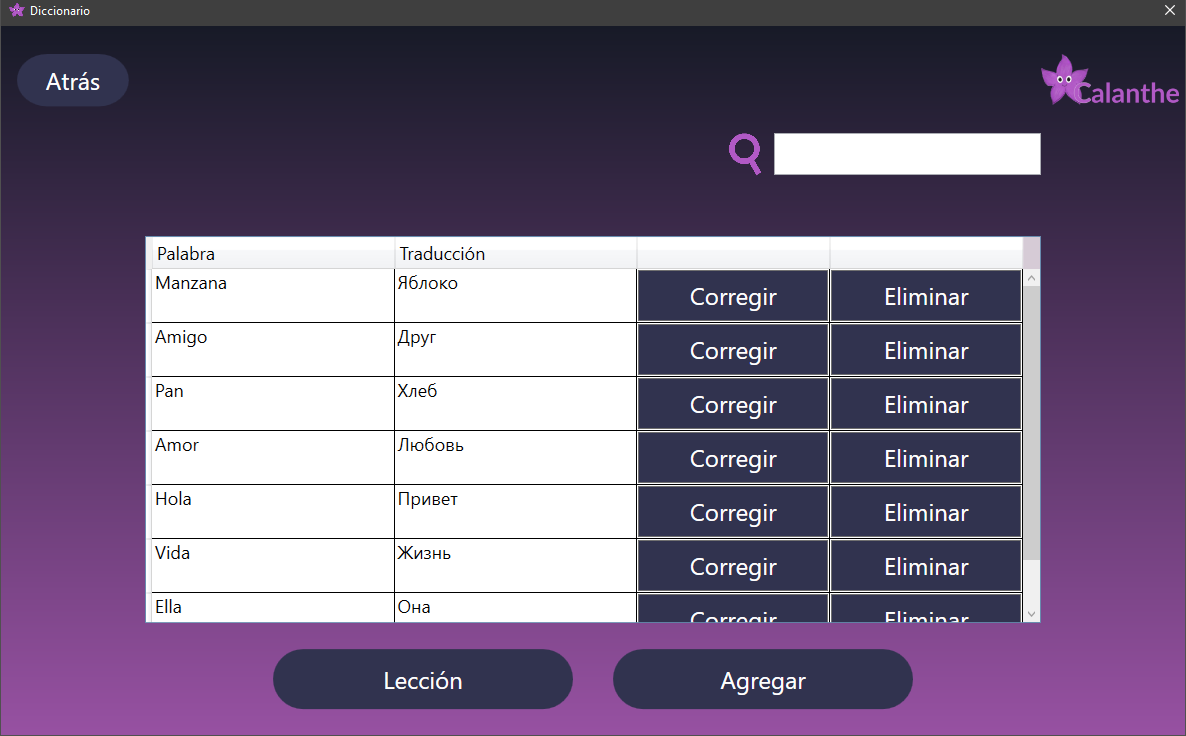


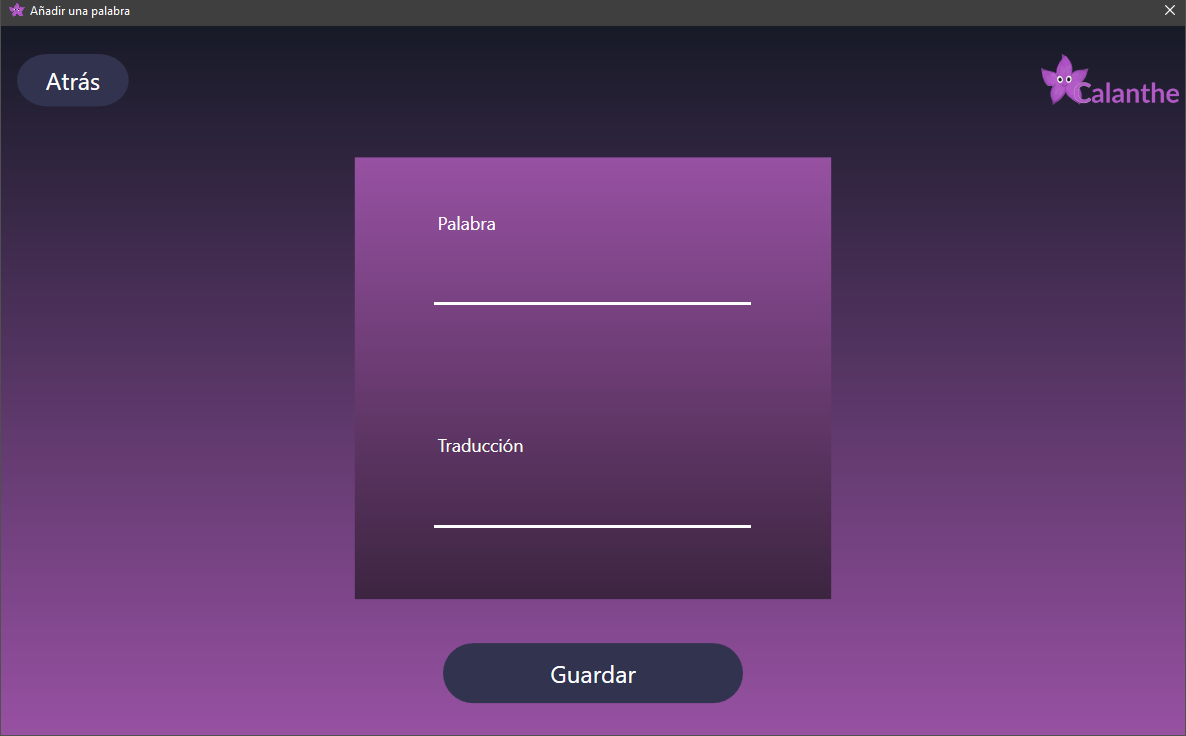


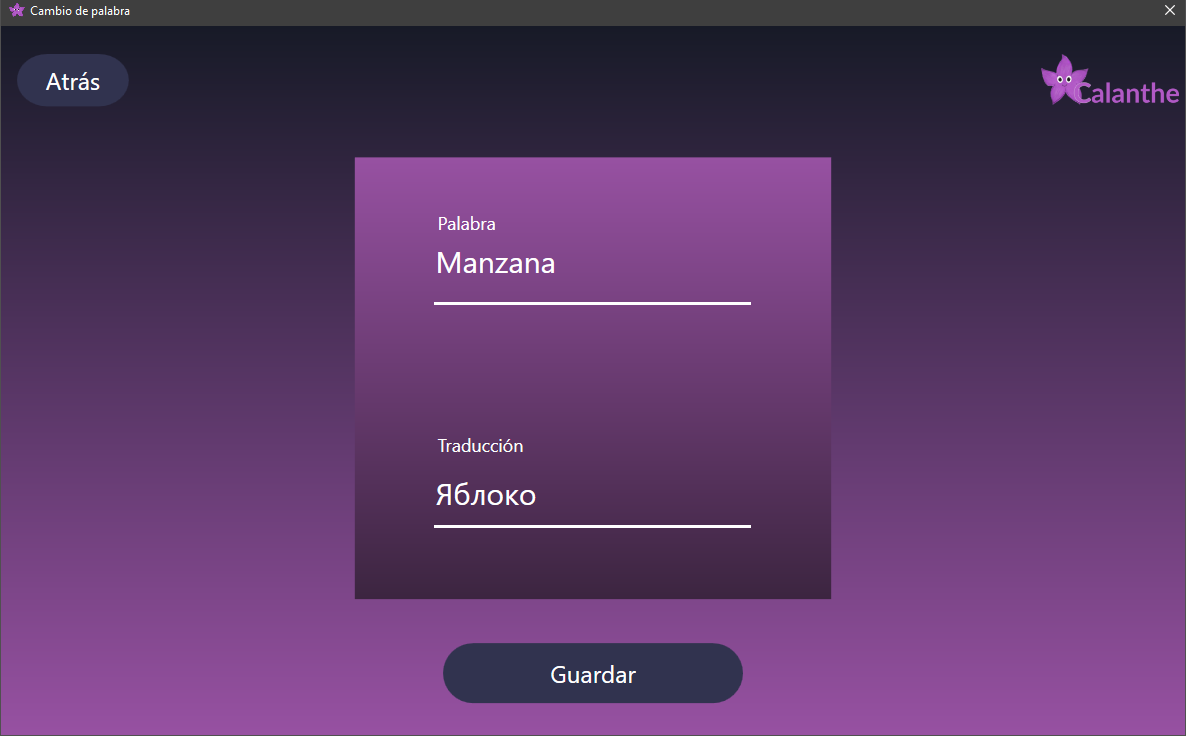


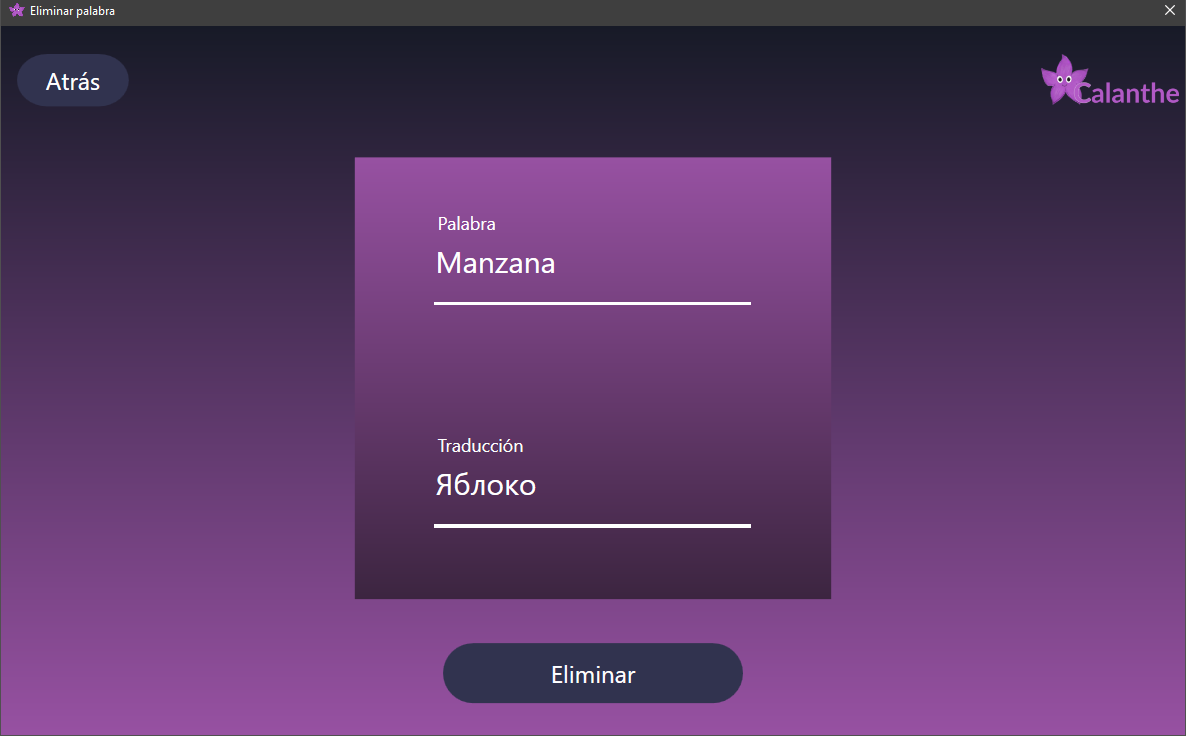












# РАЗДЕЛ 3. ПОЛЬЗОВАТЕЛЬСКАЯ ДОКУМЕНТАЦИЯ

**3.1 Руководство пользователя**

**3.2 Руководство программиста**

**3.3 Руководство по установке ПО**

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1..Новые возможности C# 9.0 // Microsoft URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/whats-.. (дата обращения: 12.05.2022).]

2..Основа WPF // Professor web URL: https://professorweb.ru/my/WPF/base\_WPF/level1/base\_W.. (дата обращения: 10.04.2022).

3..Полное руководство по WPF // Учебник по WPF URL: https://www.wpf-tutorial.com/ (дата обращения: 23.04.2022).

4. Руководство по классическим приложениям (WPF .NET) // Microsoft URL: https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/o.. (дата обращения: 11.05.2022).

5. Что нового в Visual Studio 2022? // Cpab URL: https://cpab.ru/chto-novogo-v-visual-studio-2022-clou.. (дата обращения: 18.04.2022).

6. Язык программирования C#: краткая история, возможности и перспективы // timeweb>cloud URL: https://timeweb.com/ru/community/articles/chto-takoe-.. (дата обращения: 05.03.2022).

7. Entity Framework: как быстрее написать код для работы с базой данных // Skillbox URL: https://skillbox.ru/media/code/entity\_framework/ (дата обращения: 01.05.2022).

8..Microsoft Visual Studio // Microsoft URL: https://visualstudio.microsoft.com/ru/ (дата обращения: 12.02.2022).

9..Visual..Studio..2022…//…ХАБР…URL:..https://habr.com/ru/company /microsoft/blog/553442/ (дата обращения: 25.01.2022).

# ПРИЛОЖЕНИЕ 1. Листинг программы

# ПРИЛОЖЕНИЕ 2. Список сокращений

БД - база данных

ГОСТ - государственный стандарт

ИС - информационная система

ПО - программное обеспечение

СУБД - система управления базами данных

ЯП – язык программирования

C# - C Sharp

MVVM - model-View-ViewModel

SQL - structured Query Language – Язык структурированных запросов

VS - visual studio – среда разработки

WPF – windows Presentation Foundation

# ПРИЛОЖЕНИЕ 3. Презентация