**Национальный исследовательский ядерный университет «МИФИ»**

|  |  |
| --- | --- |
|  | **Институт интеллектуальных и кибернетических систем**  **Кафедра «Компьютерные системы и технологии»** |

Пояснительная записка

к проекту по курсу «Программирование

сетевых приложений»

на тему «Веб-приложение для контекстуального изучения новых слов иностранного языка»

Студент гр. Б20-513 Логвиненко И. А. /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

Руководитель Овчаренко Е. С. / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

Москва, 2022г.

АННОТАЦИЯ

Пояснительная записка 26 с., 17 рис., 4 табл., 5 ист.

Объектом разработки является приложение для контекстуального изучения слов иностранного (английского) языка.

Целью работы является разработка веб-приложения, с помощью которого можно увеличивать словарный запас, при этом изучая контекстуальное значение слов.

В рамках анализа предметной области были рассмотрены существующие решения для контекстуального изучения иностранных слов, использующие метод интервального повторения.

Приложение реализовано на языке программирования C# с использованием фреймворка ASP.NET. Для работы с базой данных была выбрана СУБД PostgreSQL.

Содержание

[Введение 4](#_Toc154851856)

[1 Теоретическая часть 5](#_Toc154851857)

[1.1 Анализ существующих решений 5](#_Toc154851858)

[1.1.1 Anki 5](#_Toc154851859)

[1.1.2 ClozeMaster 6](#_Toc154851860)

[1.2 Постановка задачи и назначение разработки 7](#_Toc154851861)

[1.3 Схема приложения 8](#_Toc154851862)

[2 Практическая часть 10](#_Toc154851863)

[2.1 Используемые технологии 10](#_Toc154851864)

[2.2 Описание таблиц базы данных 11](#_Toc154851865)

[2.3 Уровень доступа к данным 12](#_Toc154851866)

[2.4 Уровень бизнес-логики 15](#_Toc154851867)

[2.5 Аутентификация пользователей 15](#_Toc154851868)

[2.6 Конечные точки 16](#_Toc154851869)

[2.6.1 Контроллер аутентификации 16](#_Toc154851870)

[2.6.2 Контроллер задач 17](#_Toc154851871)

[2.6.3 Контроллер персональной страницы 17](#_Toc154851872)

[2.6.4 Контроллер пользовательских приложения и задач 18](#_Toc154851873)

[2.7 Пользовательский интерфейс 18](#_Toc154851874)

[3 Ручное тестирование 24](#_Toc154851875)

[Заключение 25](#_Toc154851876)

[Список используемых источников 26](#_Toc154851877)

# Введение

Знание иностранных языков играет важную роль в повседневной и профессиональной жизни людей, а возможности, которые открываются при достаточном владении языком, делают его неотъемлемой частью личного и профессионального развития. Одной из ключевых проблем, с которой сталкиваются изучающие иностранный язык, является развитие и обогащение словарного запаса, а запоминание новых слов и их правильное использование в реальных ситуациях – критические важные навыки при изучение нового языка.

Чтение как навык является эффективным способом не только расширить словарный запас, но и погрузиться в языковую среду, понять лексические и грамматические средства и оттенки значения слов в контексте [1]. Изучение новых слов в контексте позволяет лучше их запоминать, усваивать грамматические условия их использования и развивать навыки понимания текста. Это помогает обеспечить более естественное и органичное владение языком.

Ознакомившись с веб-сервисами и приложениями, предоставляющими функционал для изучения слов иностранного языка в контексте, можно было заметить, что многие из них устарели (в плане предлагаемого для изучения материала и/или интерфейса взаимодействия с пользователем) и не обеспечивают все необходимые для комфортного и плодотворного обучения функции, а некоторые сервисы больше недоступны на территории РФ. В связи с этим было принято решение разработать веб-приложение, которое станет удобной и доступной альтернативой и эффективным инструментом для продуктивного изучения слов иностранного языка в контексте.

# 1 Теоретическая часть

## 1.1 Анализ существующих решений

Ниже рассмотрены приложения и веб-сервисы для изучения слов иностранного языка, их преимущества, особенности и ограничения.

### 1.1.1 Anki

Anki [2] представляет собой приложение, разработанное для систематизации и упрощения процесса запоминания информации. Основным назначением приложения является создание и использование флэш-карточек.

Anki основана на методе интервальных повторений (Spaced Repetition System, SRS), который научно обоснован и помогает эффективнее запоминать информацию. Суть метода заключается в повторении материала с определенными интервалами времени, что способствует укреплению памяти и улучшению запоминания.

Anki позволяет пользователям создавать собственные карточки собственных типов. Также пользователи имеют возможность использовать готовые колоды карточек, созданные другими пользователями, в различных областях знаний.

Применение алгоритмов повторения, таких как SM-2 (SuperMemo-2), обеспечивает определение оптимальных интервалов повторения для каждой карточки, учитывая степень успешности ответа и степень затруднения запоминания. Это позволяет подстроить систему повторений под индивидуальные потребности пользователя.

Каждая карточка в Anki имеет ассоциированный с ней интервал повторения, который определяется исходя из времени, прошедшего с момента последнего повторения и степени успешности ответа. При успешном повторении карточки интервал увеличивается, в то время как неуспешные ответы приводят к уменьшению интервала, что позволяет сосредоточиться на сложных для запоминания моментах.

Среди достоинств Anki можно отметить высокий уровень персонализации и адаптивности процесса обучения, а также возможность создания собственных типов карточек. Однако недостатком может быть высокий порог входа в начале использования приложения из-за большого количества настроек, довольно недружелюбный пользовательский интерфейс, а также неавтоматизированный процесс синхронизации данных между устройствами.

### 1.1.2 ClozeMaster

Clozemaster [3] представляет собой веб-приложение, разработанное для помощи в изучении иностранных языков путем использования методики заполнения пропусков (cloze deletion). Основной целью приложения является тренировка словарного запаса, грамматики и понимания контекста на основе заполнения пропусков в предложениях на иностранном языке.

Метод заполнения пропусков реализуется через предоставление пользователю предложений на изучаемом языке, где одно из слов вычеркнуто, и пользователю предлагается выбрать правильное слово для заполнения пропуска.

Среди преимуществ Clozemaster следует отметить его эффективную методику тренировки словарного запаса и грамматики через контекстуальное запоминание, а также возможность выбора уровня сложности и тематики, что позволяет персонализировать обучение в соответствии с индивидуальными потребностями и интересами пользователей.

Clozemaster также предоставляет аудио-произношение предложений, что способствует развитию навыков восприятия на слух и улучшению произношения. Пользователи также могут отслеживать свой прогресс и результаты обучения с помощью предоставляемой статистики.

Однако, стоит учитывать, что приложение имеет ограниченную степень интерактивности по сравнению с другими образовательными платформами, и может не предоставлять глубокой теоретической базы, что может осложнить более глубокое понимание грамматических конструкций.

Также, одним из существенных недостатков Clozemaster является тот факт, что данный веб-сервис недоступен в России. Кроме того, пользователи из России не могут приобрести про-подписку, открывающую доступ к большей части функционала.

## 1.2 Постановка задачи и назначение разработки

Исходя из анализа существующих решений, был сформулированы функциональные требования приложения.

Пользователю будет предложено дополнить необходимым словом предложение на иностранном (английском) языке, в котором искомое слово пропущено. При необходимости можно ознакомиться с переводом всего предложения на основной (русский) язык. Такое задание с пропуском называется карточкой.

После ввода ответа и проверки его корректности пользователь отмечает, насколько трудной для него показалась данная карточка, в зависимости от чего рассчитывается интервал, используя метод интервальных повторений, через который это предложение покажется пользователю снова. Параметры, используемые при расчете интервалов, могут быть изменены пользователем, чтобы сделать процесс тренировок более комфортным и подходящим для конкретного обучающегося.

Предложение, понравившееся пользователю или вызвавшее его интерес, может быть добавлено в избранное. И наоборот, предложение, с которым пользователь не хочет работать, может быть отмечено как нежелательное, и в будущем не будет использоваться в процессе обучения.

Пользователь может ознакомиться со статистикой своей активности в виде графов, включающей количество отработанных и добавленных карточек по дням, максимальный период тренировок без перерывов и текущий период без перерывов.

Исходя из полученных результатов были выделены следующие требования к приложению, которые позволяют совместить две сферы, объединив преимущества каждой:

Веб приложение должно предоставлять следующие возможности:

* регистрация / авторизация пользователей;
* редактирование учетной записи пользователя;
* основное упражнения – дополнить пропущенным словом предложение;
* создание пользовательских карточек;
* маркировка карточек (в избранное или нежелательное).

## 1.3 Схема приложения

Для разработки веб-приложения была выбрана трехзвенная архитектура, состоящая из трех основных уровней: клиентский, серверный и уровнем базы данных. Схема взаимодействия между уровнями приложения представлена на рисунке 1.

Изображение выглядит как диаграмма, зарисовка, дизайн

Автоматически созданное описание

Рисунок 1 – Трехзвенная архитектура

Клиентский уровень представляет собой интерфейс, с которым взаимодействует пользователь. Здесь располагается визуальное представление веб-приложения, отвечающие за отображение контента и пользовательское взаимодействие.

Серверный уровень отвечает за бизнес-логику веб-приложения. Здесь размещается серверная часть приложения, которая обрабатывает запросы от клиента, выполняет логику, работает с базой данных и формирует ответы для клиента.

Уровень сервера базы данных представляет собой часть серверной структуры, отвечающую за взаимодействие с базой данных. Этот уровень включает в себя логику, отвечающую за выполнение запросов к базе данных, передачу данных между сервером и базой данных, а также обработку данных, возвращаемых из базы данных в ответ на запросы приложения.

# 2 Практическая часть

## 2.1 Используемые технологии

Для разработки веб-приложения был выбран язык программирования C#, так как он обладает богатым набором инструментов и библиотек для создания веб-приложений.

В качестве фреймворка был выбран ASP.NET [4], поскольку он обладает обширным функционалом для разработки веб-приложений различного уровня сложности. Фреймворк поддерживает паттерн MVC (Model-View-Controller), что позволяет создавать приложения с четким разделением логики, представления и управления.

Кроме того, он обеспечивает простую интеграцию с базами данных, включая поддержку Entity Framework Core (EF Core), который позволяет работать с данными в виде объектов и упрощает маппинг объектов на схему базы данных. EF Core автоматически отслеживает изменения в объектах приложения, что упрощает процесс работы с данными и реализацию операций CRUD (создание, чтение, обновление, удаление). Также EF Core предоставляет механизм миграций, который позволяет автоматически обновлять схему базы данных при изменении моделей данных в приложении. В качестве базы данных используется PostgreSQL [5], как одна из самых распространённых и надежных СУБД с открытым исходным кодом.

Для разработки пользовательского интерфейса было принято использовать HTML, CSS, JavaScript и свободный набор инструментов Bootstrap - популярный фреймворк для создания адаптивных и кросс-браузерных веб-сайтов. Он предоставляет готовые компоненты, шаблоны, сетки и стили, что значительно ускоряет разработку пользовательского интерфейса и обеспечивает единообразный и современный внешний вид приложения.

## 2.2 Описание таблиц базы данных

Чтобы организовать хранение информации о пользователях и заданиях, которые они выполняют, была спроектирована база данных, структура которой представлена на рисунке 2.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 2 – Структура базы данных

Сущность User – в ней хранятся id пользователя и вся необходимая для работы приложения информация о пользователе.

Сущность Sentence – в ней хранятся id предложения, предложение, пропущенное слово, переводы и уровень сложности (или категория).

Сущность Task – в ней хранится id пользователя, id предложения, время первого, последнего и следующего повторения.

Сущность PersonalFactors – в ней хранятся персональные пользовательские значения факторов для метода интервальных повторений.

Сущность PersonalSettings – хранит персональные настройки пользователя.

Сущность PersonalStatistics – в ней хранится id пользователя, дата, в которую собрана статистика, и количество пройденных, повторенных и добавленных карточек.

Сущность UserSentence – хранит дополнительную информацию о предложении, добавленном пользователю.

Сущность Folder – хранит информацию о созданной пользователем папке.

## 2.3 Уровень доступа к данным

Для обеспечения взаимодействия с базами данных через объектно-реляционное отображение используется Entity Framework Core.

Был определен один из основных классов при работе с Entity Framework Core – ApplicationDbContext (рис. 3), производный от встроенного DbContext, хранящий коллекции сущностей как тип данных DbSet. Регистрация этого класса как сервиса на начальном этапе запуска приложения обеспечивает доступ сервера к базе данных.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – Взаимодействие ApplicationDbContext и базы данных

Для упрощения работы с данными и обеспечения легкой поддержки и масштабируемости приложения при проектировании уровня доступа к данным использовались паттерны UnitOfWork и (Generic) Repository (рисунок 4), которые предназначены для создания слоя абстракции между BLL (слой бизнес-логики) и DAL (слой доступа к данным). Данные паттерны снижают прямую зависимость между кодом приложения и конкретной базой данных, что позволяет легко изменять источники данных без необходимости изменения других частей приложения.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 4 – Паттерн UnitOfWork в приложение ASP.NET

Паттерн общего репозитория (Generic Repository) направлен на сокращение избыточности путем реализации общих операций с данными в едином, общем хранилище, а не в отдельных хранилищах для каждого типа сущностей. Общий репозиторий начинается с общего интерфейса, определяющего такие общие операции, как добавление, удаление, поиск и обновление. Эти операции определяются в общем виде и применимы к любому типу сущностей. Общий интерфейс затем имплементируется в конкретном классе, который обрабатывает взаимодействие с базой данных. Для работы с конкретным типом сущностей определяется интерфейс с особенными для этого типа операциями, который затем имплементируется собственным классом-репозиторием. На рисунке 5 представлен схема этого паттерна на примере репозитория для хранения предложений.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 5 – Паттерн общего репозитория (Generic Repository)

Всего было определено 5 интерфейсов репозиториев:

* ISentenceRepository – для хранения предложений;
* IUserRepository – для хранения пользователей;
* ITaskRepository – для задач, выполняемых пользователями;
* IStatisticsRepositrory – для статистики активности пользователей;
* IFavoriteSentenceRepository – для управления отмеченных пользователем предложений.

## 2.4 Уровень бизнес-логики

Уровень бизнес-логики представляет собой часть приложения, которая содержит правила, процессы и операции, напрямую связанные с основными задачами приложения. Здесь происходит обработка и управление данными, которые нужны для функционирования самого сервиса.

Можно выделить 5 основных сервиса:

* сервис аутентификации и авторизации пользователей – решает задачи регистрации и входа в систему (подробнее в разделе 2.6);
* сервис задания заполнения пропусков – формирует задачу для пользователя и обрабатывает его ответ;
* сервис просмотра выполненных задач и пользовательских приложений – предоставляет возможность ознакомиться с информацией о текущих задачах, просмотреть добавленные предложения, а также добавить собственные;
* сервис пользователя – обновление персональных данных, настройки тренировок и персонализация параметров метода интервального повторения;
* сервис метода интервального повторения – расчет интервала для пройденной пользователем карточки, основываясь на результате выполнения.

## 2.5 Аутентификация пользователей

В веб-приложении предусмотрена аутентификация и авторизация пользователей, используя JWT (JSON Web Token).

Аутентификация с использованием JWT представляет собой метод аутентификации пользователя посредством передачи подписанного токена в HTTP-заголовке или в теле запроса.

После успешной аутентификации по логину и паролю пользователя, сервер выдает JWT токен, содержащий информацию о пользователе и JWT-claims (заявки). В данном случае токен содержит id пользователя, его имя, почту и роль.

При дальнейшем взаимодействии с сервером клиент передает JWT токен в каждом HTTP-запросе. Сервер получает доступ к JWT токену из каждого запроса и проверяет его на подлинность, целостность и срок действия. Если токен корректен и действителен, процессы аутентификации и авторизации считаются пройденными успешно.

Получить доступ к приложению без аутентификации не получится.

Для реализации аутентификации по JWT был создан контроллер AuthenticationController, который позволяет пользователю регистрироваться в системе, выполнять вход в аккаунт и выходить из него. Подробнее про контроллер в разделе 2.6.1.

## 2.6 Конечные точки

Конечные точки (endpoints) в веб-приложении представляют собой URL-адреса, по которым клиенты обращаются к веб-приложению для выполнения определенных действий или получения данных. Они обеспечивают точки доступа к функциональности приложения и определяют способы взаимодействия с приложением.

### 2.6.1 Контроллер аутентификации

Корневой URL для всех запросов контроллера AuthenticationController будет содержать путь /auth.

Информация о методах (действиях, actions) контроллера аутентификации приведена в таблице 1.

*Таблица 1 – Конечные точки контроллера аутентификации*

|  |  |
| --- | --- |
| URL запроса | Функция конечной точки |
| /auth/register | Регистрация нового пользователя |
| /auth/login | Вход в систему |
| /auth/logout | Выход из системы |
| /auth/generatetoken | Генерация токена пользователя |

### 2.6.2 Контроллер задач

Корневой URL для всех запросов контроллера TaskController будет содержать путь /tasks.

Информация о методах контроллера задач приведена в таблице 2.

*Таблица 2 – Конечные точки контроллера задач*

|  |  |
| --- | --- |
| URL запроса | Функция конечной точки |
| /task/ | Стартовая страница со статистикой пользователя |
| /task/standardtraining | Стандартная тренировка с базовыми встроенными предложениями |
| /task/сustomtraining | Тренировка с пользовательскими предложениями |

### 2.6.3 Контроллер персональной страницы

Корневой URL для всех запросов контроллера PersonalPageController будет содержать путь /user.

Информация о методах контроллера задач приведена в таблице 3.

*Таблица 3 – Конечные точки контроллера персональной страницы пользователя*

|  |  |
| --- | --- |
| URL запроса | Функция конечной точки |
| /user/account | Персональная страница пользователя |
| /user/settings | Общие настройки |
| /user/update | Изменение персональных данных пользователя |
| /user/factors | Изменение персональных показателей для метода интервальных повторений |
| /user/personalsettings | Изменение персональных настроек тренировок пользователя |

### 2.6.4 Контроллер пользовательских приложения и задач

Корневой URL для всех запросов контроллера UserSentencesController будет содержать путь /sentences.

Информация о методах контроллера задач приведена в таблице 3.

*Таблица 4 – Конечные точки контроллера пользовательских приложения и задач*

|  |  |
| --- | --- |
| URL запроса | Функция конечной точки |
| /sentences/tasks | Задачи (карточки), над которыми работает пользователь |
| /sentences/personal | Персональные предложения пользователя |
| /sentences/add | Создание персонального предложения пользователя |
| /sentences/delete | Удаление персонального предложения пользователя |
| /sentences/edit | Изменение персонального предложения пользователя |

## 2.7 Пользовательский интерфейс

При входе на сайт пользователя встречает экран, в котором он может войти в существующий аккаунт или зарегистрировать новый (рисунок 6). На рисунках 7 и 8 показаны интерфейс для входа в аккаунт и регистрации соответственно.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеРисунок 6 – Начальная страница

|  |  |
| --- | --- |
| Изображение выглядит как текст, снимок экрана, Шрифт, число  Автоматически созданное описание | Изображение выглядит как текст, снимок экрана, Шрифт, число  Автоматически созданное описание |
| Рисунок 7 – Вход в аккаунт | Рисунок 8 – Регистрация |

Если аутентификация не удалась, высветится сообщение об этом (рисунок 9).

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 9 – Ошибка при неудачной попытке входа

В случае успешной аутентификации пользователя ждёт экран главного меню (рисунок 10). На нем отражается количество пройденных за текущий день новых предложение, количество предложений для повтора, доступные пользователю тренировки, а также график со статистикой за последние две недели, на котором отображаются количество созданных и пройденных карточек.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 10 – Главная страница

На рисунках 11, 12 и 13 представлены графические интерфейсы обновления персональных данных, настройки параметров тренировок и персонализация факторов для расчета интервалов повторения соответственно.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 11 – Персональные данные

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 12 – Настройки тренировок

**Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание**

Рисунок 13 – Настройки интервального повторения

На рисунке 14 показан пользовательский интерфейс задания с пропуском слов в предложении с включенной функцией выделения неправильного ответа. На рисунке 15 – интерфейс для оценки пользователем сложности выполненного задания.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 14 – Пример пользовательского интерфейса задания

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 15 – Оценка сложности задания

На рисунке 16 продемонстрирован интерфейс для просмотра задач (предложений, над которыми работал пользователь) пользователя и его собственных предложений.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 16 – Задачи пользователя

На рисунке 17 показана форма для добавления предложений.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 17 – Форма создания пользовательского предложения

# 3 Ручное тестирование

При разработке приложения было проведено ручное тестирование с помощью инструментов Postman и Swagger.

Postman позволяет легко создавать различные типы HTTP-запросов и отправлять их на сервер. С помощью этого инструмента удобно проводить ручное тестирование, получать ответы от сервера, формировать заголовки запросов.

Ещё одним инструментом для ручного тестирования является Swagger, который позволяет создать подробную документацию, описывая доступные конечные точки, структуру запросов и ответов. С его помощью было проведено тестирование методов основных контроллеров.

# Заключение

В рамках работы над этим проектом было разработано веб-приложение, которое реализует следующие функции:

* регистрация и вход в аккаунт пользователя;
* редактирование учетной записи пользователя;
* выполнение основного упражнения – дополнить пропущенным словом предложение;
* создание пользовательских карточек;
* учет пользовательских карточек и текущих задач пользователя;
* маркировка карточек (в избранное или нежелательное).

Проведено ручное тестирование приложения.

Возможное дальнейшее развитие сервиса связано с улучшением пользовательского интерфейса и расширением функционала.

# Список используемых источников

1. 5 reasons Reading is the Fastest Way to Learn a Language [Электронный ресурс] – URL: <https://lingtuitive.com/blog/5-reasons-reading-is-the-fastest-way-to-learn-a-language> (дата обращения: 15.09.2023)
2. Anki Web [Электронный ресурс] – 2023 – URL: https://ankiweb.net/about (дата обращения: 24.12.2023)
3. ClozeMaster [Электронный ресурс] – 2023 – URL: https://www.clozemaster.com/ (дата обращения: 24.12.2023)
4. ASP.NET [Электронный ресурс]. 2023. URL: https://dotnet.microsoft.com/en-us/apps/aspnet (дата обращения: 17.12.2023)
5. The PostgreSQL Global Development Group, PostgreSQL [Электронный ресурс]. 1996-2023 URL: https://www.postgresql.org/ (дата обращения: 17.12.2023)