

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ИЗУЧЕНИЯ АНГЛИЙСКОЙ
ЛЕКСИКИ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Уколовой Марии Владимировны

Научный руководитель
доцент, к. ф.-м. н.

И. А. Батраева

Заведующий кафедрой
к. ф.-м. н.

С. В. Миронов

Саратов 2019

ВВЕДЕНИЕ

В настоящее время множество людей стремится изучать иностранные языки, и одним из ключевых аспектов в их освоении является постоянное пополнение словарного запаса словами изучаемого языка, а также их повторение и использование.

Целью бакалаврской работы является разработка веб-приложения по заказу кафедры английского языка и межкультурной коммуникации факультета иностранных языков и лингводидактики Саратовского национального исследовательского государственного университета имени Н. Г. Чернышевского, которое стало бы многофункциональным инструментом для изучения иностранной лексики. Для студентов, использующих его, оно предоставит удобный способ хранения всей изучаемой лексики в виде единого структурированного словаря, а для их преподавателей — гибкую функциональность для управления словарями студентов, отправки им заданий по изучаемой лексике и отслеживания прогресса обучающийся.

Для достижения поставленной цели необходимо решить следующие задачи:

- изучить технологии для разработки веб-приложения;
- проанализировать требования и описать предметную область, ее сущности и отношения, возможные между ними;
- проанализировать существующие решения для поставленной задачи;
- спроектировать и реализовать веб-приложение.

Бакалаврская работа состоит из введения, 3-х разделов, заключения, списка использованных источников и 6-и приложений. Общий объем работы — 55 страниц, из них 40 страниц — основного текста. Список использованных источников — 20 наименований.

Первый раздел «Анализ требований к приложению и существующих решений» содержит анализ и подробный разбор требований, предъявленных заказчиком к приложению, а также анализ 3-х существующих приложений для изучения иностранной лексики, их преимущества и недостатки. Описаны причины, по которым приложения не удовлетворяют потребности заказчика.

Во втором разделе «Средства реализации приложения» представлен обзор технологий и инструментов, которые были использованы при реализации веб-приложения и приводится описание схемы базы данных приложения.

Третий раздел «Описание полученного продукта» содержит описание полученного приложения и демонстрирует примеры его работы, а также описывает некоторые детали его реализации.

1 Краткое содержание работы

1.1 Анализ требований к приложению и существующих решений

В данном разделе описываются требования к разрабатываемому приложению и проводится их анализ. Также рассмотрены некоторые из существующих приложений для решения данной задачи, их достоинства, недостатки и причины, по которым они не способны удовлетворить требованиям заказчика.

1.1.1 Сущность «Пользователь»

Ключевой сущностью приложения является пользователь, который использует его для достижения определенных целей. Для студентов это изучение новой иностранной лексики, представленной для каждого из них в виде личного словаря, и ее усвоение путем прохождения тренировочных заданий, а для их преподавателей — контроль за этим процессом в максимально простой и удобной форме.

Основными атрибутами пользователя в системе являются:

- логин пользователя;
- имя пользователя;
- пароль;
- дата регистрации пользователя в системе.

Также в связи с вышеописанной логикой работы приложения, было решено ввести разделение пользователей на студентов и преподавателей как один из атрибутов данной сущности. У пользователей с разными ролями существенно различается доступная функциональность и имеются отличия в интерфейсе приложения. Для пользователя с ролью студента в системе сохраняется дополнительная информация, которая включает в себя:

- сведения о том, какой пользователь с ролью преподавателя является его учителем;
- доступность словаря для данного студента в настоящий момент времени.

Рассмотрим далее, какие отношения возможны между различными типами пользователей. Здесь следует отметить, что каждый студент закреплен за одним преподавателем, который отвечает за наполнение его словаря и отправку этому студенту тренировочных заданий и тестов. Наличие у студента более одного преподавателя не предусматривается, однако у преподавателя может быть любое число студентов. Соответственно, для пользователя с ролью пре-

подавателя в отношении его учеников имеется следующая функциональность:

- просмотр полного списка его студентов, где отображается основная информация о каждом из студентов;
- просмотр отчета о любом его студенте;
- возможность открыть или закрыть функцию словаря для любого из его студентов.

1.1.2 Сущность «Группа»

Одним из основных требований к системе является возможность гибкого объединения студентов в группы для изучения лексики и назначения им тестов и тренировочных заданий. Поэтому следующей рассмотренной сущностью будет сущность группы.

Основными атрибутами группы являются:

- название группы;
- создатель;
- дата создания группы.

Группа создается одним из пользователей с ролью преподавателя и может содержать любое количество студентов. Также каждый из студентов может принадлежать как к одной, так и к нескольким группам (например, к группе своего курса в университете и группе отстающих, которым преподаватель желает добавить базовую лексику для изучения). Отправка слов, а также тренировочных заданий и тестов осуществляется не лично для каждого студента, а для группы сразу.

Таким образом, между пользователем с ролью преподавателя и сущностью группы возможны следующие отношения:

- создание и редактирование группы с добавлением и удалением студентов;
- открытие или закрытие доступа к словарю для студентов группы;
- добавление лексической единицы для изучения группам студентов;
- добавление группам студентов задания или теста для проверки изученного.

Между пользователем с ролью студента и группой существует только одно отношение — вхождение студента в группу. Также в системе сохраняется дата присоединения студента к группе.

1.1.3 Сущности «Лексическая единица» и «Словарь»

Другой ключевой сущностью системы является лексическая единица. Лексическая единица в приложении может быть представлена словом, устойчивым выражением слов или идиомой на иностранном языке, которые преподаватель считает необходимым отправить своим студентам для изучения. Возможностью добавления и описания такой единицы в системе обладает только пользователь с ролью преподавателя. Пользователи с ролью студента не обладают таким правом ввиду недостатка их компетентности, а также ввиду необходимости сделать описание лексических единиц единообразным для всех студентов внутри группы. Ограничение на количество добавляемых единиц преподавателями отсутствует.

В результате анализа потребностей пользователей для лексической единицы в системе были выделены следующие атрибуты:

- сама лексическая единица;
- определение (приводится на иностранном языке);
- перевод лексической единицы;
- часть речи (словосочетание, идиома);
- тема, к которой преподаватель относит лексическую единицу;
- уровень сложности лексической единицы;
- пример использования лексической единицы в предложении;
- автор лексической единицы в системе;
- группы студентов, которым единица пойдет на изучение.

Группы лексических единиц, созданных конкретным пользователем с ролью преподавателя, образуют его словарь. При описании лексической единицы данный пользователь сразу же указывает, каким группам он желает отправить ее на изучение. Таким образом, словарь для студентов формируется на основании их принадлежности к тем или иным группам.

Рассмотрим теперь, какие отношения возможны между преподавателем, который использует приложение, и его словарем. Эти отношения могут быть выражены в следующих функциях приложения, доступных преподавателю:

- просмотр словаря, где для каждой лексической единицы указаны группы студентов, которым она была отправлена на изучение;
- редактирование уже добавленных лексических единиц и создание новых;
- фильтрация при просмотре словаря в случае, если преподаватель желает

увидеть лексические единицы, относящиеся к определенным темам или отправленные на изучение конкретным группам его студентов.

Для пользователей-студентов, которым доступна функция просмотра словаря, он призван служить удобным инструментом для хранения всей иностранной лексики, которую нужно отчитать преподавателю. Редактирование лексических единиц, которыми пополняется словарь студента, самим студентом запрещено по тем же причинам, что и самостоятельное создание лексических единиц в системе. Следует также отметить, что словарь строится таким образом, чтобы студенту приходилось минимально задействовать дополнительные источники для запоминания новых лексических единиц.

Тем не менее, словарь студенту доступен не всегда. Одной из ключевых особенностей приложения является реализация функции, которая позволит преподавателям временно закрыть доступ к словарю либо для конкретных студентов, либо для их групп. Такая функция особенно полезна, если преподаватель назначил студентам тест и желает проверить, насколько хорошо они усвоили лексику, которая в него включена, без возможности заглянуть в словарь.

1.1.4 Сущность «Тест»

Как уже было упомянуто ранее, контроль за усвоением студентами иностранной лексики осуществляется преподавателем посредством отправки им тренировочных заданий и тестов. Для этого в системе предусмотрена сущность теста. Одним из основных требований к разработке системы является автоматическая генерация и проверка таких заданий на основе имеющегося у преподавателя словаря с указанием некоторого числа первичных настроек.

Для сущности теста в системе были выделены следующие атрибуты:

- название задания или теста;
- автор теста в системе;
- тип теста;
- тема лексических единиц, включенных в тест (необязательно);
- уровень лексических единиц, включенных в тест (необязательно);
- дата создания и отправки теста студентам;
- группы студентов, которым назначается тест.

С учетом имеющейся конфигурации словаря (см. пункт 1.1.3) преподаватель имеет возможность создать задание одного из трех типов:

1. перевести лексическую единицу на иностранный язык;
2. привести слово, словосочетание или идиому по их определению;
3. вставить пропущенную лексическую единицу в предложение.

Тип теста пользователь с ролью преподавателя выбирает при его создании. Также при желании можно выбрать одну из тем, которые встречаются у него в словаре, или уровень лексических единиц для включения в тест. Таким образом, имеется возможность создания общих тестов для проверки всего пройденного материала или тестов более узкой направленности для того, чтобы проверить, как студенты справляются с лексикой из определенного раздела.

Ещё одной первичной настройкой теста является указание групп, которым он будет отправлен. После того как все желаемые настройки будут выбраны, преподавателю предлагается список слов, подходящих заданным критериям. Если тема и уровень не были указаны, отображаются все слова, которые когда-либо были отправлены студентам из заданных групп на изучение, в противном случае — происходит дополнительная фильтрация. Выбрав все слова, которые он желает включить в тест, преподаватель завершает его создание.

Генерация теста и его отправка выбранным группам происходит автоматически. Минимальное количество вопросов в тесте составляет 5. Редактирование теста после его создания и отправки не допускается.

Создание заданий является лишь одной функцией, доступной преподавателю в отношении сущности теста. Рассмотрим остальные:

- просмотр списка всех созданных им заданий (включая их настройки и группы, которым задание было отправлено);
- просмотр содержимого каждого задания (в том виде, в котором оно отображается для студентов, только с ответами и без возможности прохождения теста);
- просмотр отчетов о выполнении студентами заданий.

Согласно требованиям, для наиболее оптимального анализа успехов обучающихся приложение должно предоставлять несколько видов отчетов по заданию:

- сводная таблица с результатами теста для всех студентов, которым данный тест был назначен;
- таблица с баллами, полученными каждым студентом по каждому вопросу

(в таблице также приводится, сколько баллов в сумме получил каждый студент и сколько студентов справилось с каждым вопросом);

- подробный отчет о прохождении теста конкретным студентом, где преподаватель может увидеть ответы учащегося на каждый вопрос.

Когда студент выполняет тест, системой запоминаются все его ответы и время прохождения задания. Студент может:

- просмотреть весь список назначенных тестов, для каждого из которых указывается, пройден ли тест (в таком случае также отображаются сумма баллов за тест и полученный процент);
- пройти какой-либо из еще не пройденных тестов;
- получить подробный отчет о пройденном тесте, где для каждого вопроса он увидит свой ответ, а также ответ, предложенный в качестве правильного, и совершил ли он в этом вопросе ошибку.

1.1.5 Анализ существующих решений

Помимо анализа требований к разрабатываемому приложению, был также проведен анализ нескольких существующих приложений, которые позволяли бы своим пользователям вести собственные словари иностранной лексики и предоставляли бы функционал на их основе. В качестве таких приложений выступили приложение для изучения слов онлайн w2mem, сервис для изучающих английский язык Myefe и приложение для изучения иностранных языков WordSteps. Анализ выявил, что ни одно из рассмотренных приложений не удовлетворяет требованиям заказчика по ряду причин. Во-первых, большинство приложений, пусть и предлагает создание неограниченного числа словарей, не содержит достаточного количества информации по каждому слову внутри него. Многие из них ограничиваются лишь переводом, транскрипцией и комментарием, тогда как заказчик желал бы иметь возможность указать также определение лексической единицы, ее тему, уровень сложности, часть речи и пример ее использования в предложениях.

Во-вторых, каждое из рассмотренных приложений имеет функционал прохождения заданий по изучаемой лексике, однако ни в одном из них информация о пройденных тестах не сохраняется. Нет возможности не только сделать вывод о собственном прогрессе на основе общей статистики упражнений, но и посмотреть, где у пользователя чаще всего возникают ошибки, а какие слова даются легко.

В-третьих, мало приложений предлагают разделение пользователей на студентов и преподавателей с тем, чтобы последние могли контролировать процесс усвоения их учениками материала. Из рассмотренных это было единственное приложение WordSteps, однако и оно не предлагает данный функционал бесплатно, а требует приобретения платной подписки. Несмотря на то что функционал преподавателя в данном приложении действительно богат, заказчику хотелось бы иметь бесплатный аналог такого приложения.

Из всего вышесказанного было решено, что при разработке собственного приложения стоит сделать акцент на структуре словаря пользователей. Также, вместо того чтобы предлагать пользователю множество словарей, он будет один, однако его можно будет отфильтровать по теме или по группам студентов. Преподавателю будет предоставлен полный контроль над созданием тестов, а также в системе будет сохраняться информация о каждой попытке студентов пройти задание и их ответы на все вопросы теста.

1.2 Средства реализации приложения

В данном разделе рассмотрены некоторые технологии, инструменты и другие решения, которые были использованы для реализации данного веб-приложения. Отметим, что при их выборе учитывались небольшие сроки реализации проекта, популярность тех или иных средств разработки, а также необходимость ознакомления с данными технологиями и их изучения перед началом реализации проекта.

Поскольку на сегодняшний день существует большое количество фреймворков, которые позволяют значительно повысить скорость и удобство разработки и оптимизировать затраты на создание приложения, было принято решение использовать их для достижения целей проекта. Поскольку реализацию проекта планировалось осуществить на языке программирования Python, выбор производился между двумя наиболее популярными веб-фреймворками для этого языка — Django и Flask. В результате обзора для написания серверной части приложения по ряду причин был выбран Flask [1].

К преимуществам данного фреймворка можно отнести следующее:

1. Flask обеспечивает простоту, гибкость и полный контроль над проектом. Он позволяет разработчику самостоятельно решать, как реализовывать те или иные аспекты приложения.
2. Flask поставляется вместе с шаблонизатором Jinja2, системой маршрути-

зации Werkzeug, встроенным веб-сервером и многочисленными инструментами отладки.

3. Flask поддерживает серию пакетов-расширений (названия обычно начинаются с Flask-), предоставляющих богатый функционал, и позволяет при желании использовать широкий набор внешних библиотек и дополнений.
4. Flask снабжен хорошей документацией.
5. В целом, данный фреймворк позволяет получить больше опыта и возможностей для обучения.

В качестве СУБД была выбрана SQLite [2]. Данная СУБД отличается своей быстротой и мощностью благодаря технологиям обслуживающих библиотек, а также предоставляет довольно широкий набор инструментов для работы. К другим достоинствам относятся популярность и надежность, полностью свободная лицензия, безопасность, простота использования и отсутствие необходимости настройки сервера СУБД, очень экономичная архитектура. Данная СУБД является «встраиваемой», что означает, что SQLite не использует парадигму клиент-сервер и движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а предоставляет библиотеку, с которой программа компонуется и движок становится составной частью программы. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле.

Поскольку во фреймворке Flask отсутствует встроенная ORM, которая позволила бы создать некую удобную абстракцию над базой данных приложения, было решено использовать расширение Flask-SQLAlchemy [3], которое предоставляет обертку для проекта SQLAlchemy [4] — программной библиотеки на языке Python для работы с реляционными СУБД с применением технологии ORM. Flask-SQLAlchemy удобен в использовании в простых приложениях и легко расширяется для сложных.

Помимо этого, при реализации приложения были задействованы и другие расширения данного фреймворка, такие как:

- Flask-Login [5], позволяющее осуществлять менеджмент сессий пользователей в приложениях, авторизацию и выход из приложения, ограничение доступа к представлениям для неавторизованных пользователей;
- Flask-Bcrypt [6] для безопасного хранения паролей пользователей;

- Flask-WTF [7], которое представляет собой обертку пакета WTForms для обработки веб-форм;
- Flask-Bootstrap [8] для стилизации и форматирования веб-страниц.

Bootstrap — это CSS-фреймворк, который изначально создавался для внутреннего использования компанией «Twitter», но в итоге был опубликован в открытый доступ и стал хорошим набором инструментов для front-end разработки под названием «Bootstrap». К преимуществам фреймворка Bootstrap можно отнести высокую скорость разработки макетов страниц сайта, кроссбраузерность и адаптивность сайта, а также легкость в использовании.

WTForms [9] представляет собой довольно удобную библиотеку для создания и обработки форм, которая имеет ряд встроенных классов для их описания, а также для описания их полей.

1.3 Описание полученного продукта

В данном разделе бакалаврской работы дается описание полученного программного продукта, демонстрируются примеры его использования, а также приводятся некоторые детали его реализации. Описана функциональность четырех основных разделов, которые были заявлены в требованиях, а также авторизация и регистрация в приложении.

Рассмотрены формы авторизации и регистрации и приветственная страница приложения для двух видов пользователей.

В описании раздела «Студенты», доступного преподавателю, представлен его интерфейс и рассмотрена его функциональность, которая заключается в просмотре списка обучающихся и отчетов по ним.

Аналогично описан раздел «Группы», где преподаватель может просматривать списки своих групп, добавлять или удалять группы, а также редактировать их по своему желанию.

Для раздела «Словарь» приводится описание функциональности, доступной как преподавателям, так и студентам. Рассмотрен интерфейс словаря пользователя, а также формы создания и редактирования лексических единиц, которые принадлежат к числу основных функций учителей в системе.

В описании раздела «Тесты» рассмотрены функциональность студентов, позволяющая просматривать список заданий, выполнять их и получать по ним отчеты. Для преподавателей рассматриваются функции создания нескольких видов тестов и их просмотра, а также получения нескольких видов отчетности.

ЗАКЛЮЧЕНИЕ

В рамках данной работы была поставлена и достигнута задача создания веб-приложения для изучения иностранной лексики. Для обеспечения качества приложения были тщательно проанализированы требования, а также рассмотрены существующие решения поставленной задачи, их достоинства и недостатки. Были спроектированы модель данных предметной области, на основе которой была создана база данных, и общая архитектура приложения, а также спроектированы и реализованы основные модули веб-приложения. В качестве одной из задач был изучен фреймворк Flask для языка программирования Python и некоторые его расширения, такие как Flask-SQLAlchemy, Flask-Login, Flask-WTF и т. д., с помощью которых в дальнейшем было реализовано приложение. Кроме этого, была проведена работа по визуальному оформлению приложения при помощи CSS-фреймворка Bootstrap.

Таким образом, все поставленные задачи выполнены и цель работы достигнута. Результатом проделанной работы стало полноценное веб-приложение для изучения иностранной лексики, в котором реализованы все поставленные требования для двух видов пользователей — преподавателей и студентов. В дальнейшем планируется проведение более полного и тщательного тестирования приложения, а также его публикация и поддержка.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Flask web development, one drop at a time [Электронный ресурс]. — URL: <http://flask.pocoo.org/docs/1.0/> (Дата обращения 28.01.2019). Загл. с экр. Яз. англ.
- 2 Sqlite Documentation [Электронный ресурс]. — URL: <https://www.sqlite.org/draft/docs.html> (Дата обращения 14.02.2019). Загл. с экр. Яз. англ.
- 3 Flask-SQLAlchemy [Электронный ресурс]. — URL: <https://flask-sqlalchemy.palletsprojects.com/en/2.x/> (Дата обращения 14.02.2019). Загл. с экр. Яз. англ.
- 4 SQLALchemy ORM [Электронный ресурс]. — URL: <https://lectureswww.readthedocs.io/6.www.sync/2.coddig/9.databases/2.sqlalchemy/index.html> (Дата обращения 10.02.2019). Загл. с экр. Яз. англ.
- 5 Flask-Login [Электронный ресурс]. — URL: <https://flask-login.readthedocs.io/en/latest/> (Дата обращения 15.02.2019). Загл. с экр. Яз. англ.
- 6 Flask-Bcrypt [Электронный ресурс]. — URL: <https://flask-bcrypt.readthedocs.io/en/latest/> (Дата обращения 18.02.2019). Загл. с экр. Яз. англ.
- 7 Flask WTF [Электронный ресурс]. — URL: <https://flask-wtf.readthedocs.io/en/stable/> (Дата обращения 05.03.2019). Загл. с экр. Яз. англ.
- 8 Flask-Bootstrap [Электронный ресурс]. — URL: <https://pythonhosted.org/Flask-Bootstrap/> (Дата обращения 8.03.2019). Загл. с экр. Яз. англ.
- 9 WTForms Documentation [Электронный ресурс]. — URL: <https://wtforms.readthedocs.io/en/stable/> (Дата обращения 10.03.2019). Загл. с экр. Яз. англ.