

Рабочее название: SimpleLearn

Системы: web-сайт; android; IOS; windows-phone (расставлены в порядке приоритета)

Проектирование

Приложение разрабатывается в трех версиях: браузерная, мобильная (Android) и десктопная. Также для тестирования будет разрабатываться консольная версия приложения.

Описание серверных операций

Клиент – Сервер

1. reg.py
2. login.py
3. setconfig.py
4. getinfo.py
5. gettimers.py
6. settimer.py
7. deltimer.py
8. getlesson.py
9. gettest.py
10. gradetestresults.py
11. getstat.py
12. logout.py
13. unreg.py

1. reg.py

Сервер просит создать в БД таблицу Users с новым пользователем. В поля name, password, e-mail записываются введенные пользователем значения. ID пользователя генерируется автоматически, в поле xp записывается ноль, остальные поля остаются пустыми. Также создается таблица Timers с полями id (генерируется автоматически), user_id (записывается значение поля id из таблицы Users), остальные поля остаются пустыми. Сервер присылает ошибку, если не удалось создать таблицу (такой name уже существует) или записать значение в какое-либо поле. Сервер присылает «ОК», если это удалось.

Input:

Username	(string)
Password	(string)
E-mail	(string)

Output:

```
{
  "Status": ("Success" / "Failure")
}
```

2. login.py

Сервер просит найти в БД поле name таблицы Users, значение которого совпадает с введенным именем. Далее проверяется соответствие значения поля password и введенного пользователем пароля. Если они не совпадают, сервер присылает ошибку. Если совпадают Из найденной таблицы Users берется значение поля ID. Сервер просит создать в БД таблицу Sessions с новой сессией с полями session_id (генерируется автоматически), user_id (найденный id из таблицы пользователя), started (время начала сессии), expires (если пользователь нажал remember_me, то сюда записывается максимальное значение. Если нет, записывается фиксированный период времени окончания сессии). Сервер выдает ошибку, если не существует таблицы Users с данным name или она существует, но не совпадает пароль, или если не удастся записать значение в какое-либо поле.

Input:

Username (string)
Password (string)
RememberMe (boolean)

Output:

```
{  
  "Status":            ("Success" / "Failure")  
  "SessionId":        (string)  
  "StartTime":        (string)  
  "ExpiryPeriod":    (string)  
}
```

3. setconfig.py

Сервер просит найти в БД поле session_id таблицы Sessions, значение которого совпадает с переданным id сессии. По значению поля user_id в БД находится таблица Users, в которой это значение совпадает со значением поля id. В поля Nlang и Flang этой таблицы записываются выбранные пользователем из списка (в строки списка записываются значения из полей lang_name таблицы Languages) родной и изучаемый языки. Сервер просит создать в БД таблицу User_Level.

Вызывается gettest.

Сервер может передать в БД запрос на добавление нового языка. В таком случае БД передает серверу список языков (в строки списка записываются значения из полей lang_name таблицы Languages), сервер передает пару Nlang, Flang. Если значение поля Nlang таблицы Users совпадает с переданным значение Nlang, и значение Flang не соответствует значению поля lang_id ни одной из таблиц User_Level, значит добавляется Flang: сервер просит создать в БД таблицы User_Level и Progress. Значение поля Flang в таблице Users меняется на переданный иностранный язык. Вызывается gettest.

Если значение поля Nlang таблицы Users совпадает с переданным значение Nlang, и значение Flang соответствует значению поля lang_id какой-либо из таблиц User_Level, значит меняется Flang: в поле Flang таблицы Users записывается переданное значение иностранного языка.

Если значение поля Nlang таблицы Users не совпадает с переданным значением Nlang, значит изменяется Nlang: сервер просит БД удалить все таблицы User_Level, Progress, W_To_Repeat, W_To_Learn, найденные по id пользователя; удалить значения полей Flang и xp из таблицы Users. Затем БД просит сервер передать значение Flang. Вызывается gettest.

Сервер выдает ошибку, если не существует таблицы Sessions с данным session_id; если какую-либо из таблиц не удалось создать или не удалось записать значение в поле. Также сервер выдает ошибку, если значения Nlang и Flang совпадают.

Input:

SessionId (string)

NativeLanguage (string)

ForeignLanguage (string)

Level (int)

Output:

```
{  
"Status": ("Success" / "Failure")  
}
```

4. getinfo.py

Сервер просит найти в БД поле session_id таблицы Sessions, значение которого совпадает с переданным id сессию. По значению поля user_id в БД находится таблица Users, в которой это значение совпадает со значением поля id таблицы Sessions. БД передает серверу значения полей name, Nlang, xp из таблицы Users и список значений полей lang_id и level_id из всех таблиц User_Level, значение user_id которых совпадает с id из таблицы Users.

Сервер выдает ошибку, если не существует таблицы Sessions с данным session_id; если какую-либо из таблиц не удалось создать или удалить; не удалось записать значение в поле. Также сервер выдает ошибку, если значения Nlang и Flang совпадают или если после выхода из сессии поле Flang остается пустым.

Input:

SessionId (string)

Output:

```
{  
"Status": ("Success" / "Failure")  
"Information":  
{  
"Username": (string)  
"NativeLanguage": (string)  
"XP": (int)  
"Languages":  
[  
{  
"Language": (string)  
"Level": (int)  
}  
]  
}  
}
```

5. gettimers.py

Сервер просит найти в БД поле session_id таблицы Sessions, значение которого совпадает с переданным id сессии. По значению поля user_id в БД находится поле user_id в таблице Timers. На сервер передается список таймеров со значениями полей id, count, time, days из таблицы Timers. Если значения этих полей пустые, то БД сообщает серверу, что у данного пользователя таймеров не установлено.

Сервер выдает ошибку, если не существует таблицы Sessions с данным session_id или не найдено поле user_id таблицы Timers по заданному user_id.

Input:

SessionId (string)

Output:

```
{
  "Status":      ("Success" / "Failure")
  "Timers":
  [
    {
      "Id":        (int)
      "Time":      (string)
      "Days":      (string)
      "Count":     (int)
    }
  ]
}
```

6. settimer.py

Сервер просит найти в БД поле session_id таблицы Sessions, значение которого совпадает с переданным id сессии. По значению поля user_id в БД находится поле user_id в таблице Timers. Сервер передает базе данных значения count, time и days, которые записываются в одноименные поля найденной таблицы Timers.

Сервер выдает ошибку, если не существует таблицы Sessions с данным session_id, не найдено поле user_id таблицы Timers по заданному user_id или не удалось записать значение в какое-либо поле.

Input:

SessionId (string)

Time (string)

Days (string)

Count (int)

Output:

```
{  
"Status":      ("Success" / "Failure")  
}
```

7. deltimer.py

Сервер просит найти в БД поле session_id таблицы Sessions, значение которого совпадает с переданным id сессии. По значению поля user_id в БД находится поле user_id в таблице Timers. Если значение поля id таблицы Timers совпадает с переданным сервером, то удаляем таблицу с данным id.

Сервер выдает ошибку, если не существует таблицы Sessions с данным session_id или не найдено поле user_id таблицы Timers.

Input:

SessionId (string)

TimerId (int)

Output:

```
{  
"Status":      ("Success" / "Failure")  
}
```

8. getlesson.py

Производятся те же действия, что и в пункте 9 по созданию таблицы Test. По значению поля id из таблицы Test находится поле test_id из таблицы Question. По значению поля word_translation находим поля id таблиц Dictionary_Entries. На сервер передаются значения полей word1_id и word2_id из этой таблицы.

Сервер выдает ошибку, если не существует таблицы Sessions с данным session_id; если какую-либо из таблиц не удалось создать, найти или не удалось записать значение в поле.

Input:

SessionId (string)

TestId (int)

Type (string)

Output:

```
{  
"Status":      ("Success" / "Failure")  
"Lesson"  
[  
{
```

```
"NativeWord":    (string)
"ForeignWord":   (string)
}
]
}
```

9. gettest.py

Сервер просит найти в БД поле session_id таблицы Sessions, значение которого совпадает с переданным id сессии. Сервер просит БД создать таблицу Test с полями _id (генерируется автоматически), type (передается сервером), question. При создании поля question создаются таблица Question с полями:

id:

Генерируется автоматически;

type:

Присваивается значение поля type_quest из таблицы Type;

count:

Если в таблице User_Level поле level_id не пустое, то в поле count записываем фиксированное количество слов для обычного теста,

Если в таблице User_Level поле level_id пустое, то в поле count записываем фиксированное количество слов для начального теста;

word_translation:

По значению полей Flang и id таблицы Users находится таблица User_Level.

Если поле level_id этой таблицы не пустое и значение поля type таблицы Test соответствует обычному тесту, то по значению поля level_id таблицы User_Level находится поле level_id таблицы Dictionary. Затем находится определенное количество (равное значению поля count из таблицы Question) случайных таблиц Dictionary_entries со значением поля dict_id, совпадающим со значением поля id в таблице Dictionary. Значения полей id из этих таблиц записываются в поле word_translation,

Если поле level_id этой таблицы не пустое и значение поля type таблицы Test соответствует тесту на повторение, таблицы W_To_Repeat сортируются по возрастанию значения поля time_per этой таблицы. В поле word_translation таблицы Question записываются значения полей dict_entry_id из первых нескольких (число, равное значению поля count из таблицы Question) таких таблиц,

Если поле level_id этой таблицы не пустое и значение поля type таблицы Test соответствует тесту по определенной теме, то по значению поля level_id таблицы User_Level находится поле level_id таблицы Dictionary, значение поля theme которой равно переданной сервером теме. Затем находится определенное количество (равное значению поля count из таблицы Question) случайных таблиц Dictionary_entries со значением поля dict_id, совпадающим со значением поля id в таблице Dictionary. Значения полей id из этих таблиц записываются в поле word_translation,

Если поле level_id этой таблицы пустое, то для каждого из уровней в таблице Level сервер просит найти в БД таблицу поле level_id таблицы Dictionary со значением, совпадающим со значением поля id в таблице Level. Находим несколько (число, равное значению поля count из таблицы Question, деленное на количество уровней) случайных таблиц Dictionary_entries со значением поля dict_id, совпадающим со значением поля id в таблице Dictionary. В поле word_translation таблицы Questions записываем список значений полей id из найденных таблиц Dictionary_entries.

База данных передает таблицу Test на сервер.

Сервер выдает ошибку, если не существует таблицы Sessions с данным session_id; если какую-либо из таблиц не удалось создать или не удалось записать значение в поле.

Input:

```

SessionId      (string)

Type           (string)

Output:

{
  "Status":      ("Success" / "Failure")
  "TestId":      (int)
  "Questions":
  [
    {
      "Question": (string)
      "Answers":
      [
        (string)
      ]
    }
  ]
}

```

10. gradetestresults.py

Сервер просит найти в БД поле session_id таблицы Sessions, значение которого совпадает с переданным id сессии и поле id таблицы Test, значение которого совпадает с переданным id теста. По значению поля id из таблицы Test находится поле test_id из таблицы Question. По значению поля word_translation находим поля id таблиц Dictionary_Entries. БД сравнивает полученные с сервера ответы с полями word1_id и word2_id из этой таблицы. Ответы, совпадающие со значениями этих полей, проверяются на соответствие со словами из списка в поле dict_entry_id таблицы Progress. Те слова, которым не найдено соответствие, добавляются в этот список и записываются в поле dict_entry_id таблицы W_To_Repeat (полю num_rep присваивается значение 1, поле time_rep = (время в данный момент) + (значение поля time таблицы Time_Repeat, значение поля number которой совпадает со значением поля num_rep таблицы W_To_Repeat)). Для тех слов, которым найдено соответствие, в таблице W_To_Repeat значение поля num_rep увеличивается на 1, поле time_rep = (время в данный момент) + (значение поля time таблицы Time_Repeat, значение поля number которой совпадает со значением поля num_rep таблицы W_To_Repeat). Значение поля xp таблицы Users увеличивается на величину, равную количеству только что записанных в поле dict_entry_id таблицы Progress, умноженному на вес одного слова. Ответы, не совпадающие со значениями полей word1_id и word2_id таблицы Dictionary_Entries, проверяются на соответствие со словами из списка в поле dict_entry_id таблицы Progress. Те слова, которым найдено соответствие, удаляются из этого списка и также удаляются таблицы W_To_Repeat, значение поля dict_entry_id которых совпадает с id этих пар слов. Все ответы, не совпадающие со значениями полей word1_id и word2_id таблицы Dictionary_Entries, добавляются в список слов в поле dict_entry_id таблицы W_To_Learn.

Значение поля xp таблицы Users изменяется на величину, равную:

$(\text{количество слов, добавленных в список поля dict_entry_id таблицы Progress}) - (\text{количество слов, удаленных из списка поля dict_entry_id таблицы Progress}) * (\text{вес одного слова})$.

В БД находится таблица поле id таблицы Level со значением на 1 большим, чем значение поля lang_id таблицы User_Level. Если количество слов в списке поля dict_entry_id таблицы Progress больше, чем значение поля needed_words найденной таблицы Level, то полю level_id таблицы User_Level присваивается значение поля id из найденной таблицы.

БД возвращает серверу значения полей word1_id и word2_id из таблицы Dictionary_Entries.

Сервер выдает ошибку, если не существует таблицы Sessions с данным session_id; если какую-либо из таблиц не удалось

найти или не удалось записать значение в поле.

Input:

SessionId (string)

TestId (string)

Answer-1 (string)

Answer-2 (string)

...

Answer-N (string)

Output:

{

“Status”: (“Success” / “Failure”)

“RightAnswers”:

[

(string)

]

}

11. getstat.py

Сервер просит найти в БД поле session_id таблицы Sessions, значение которого совпадает с переданным id сессии. По значению поля user_id в БД находится таблица Users, в которой это значение совпадает со значением поля id таблицы Sessions. По значению поля Flang из этой таблицы, находятся все таблицы User_Level, значение полей lang_id которых совпадает со значением поля Flang из таблицы Users.

БД передает серверу значения полей name, xp из всех таблиц Users, значение поля id которых совпадает с значениями полей user_id найденных таблиц User_Level. Также передается значение поля level_name из таблицы Level, значение поля id которой совпадает с значением поля level_id найденных таблиц User_Level.

Сервер выдает ошибку, если не существует таблицы Sessions с данным session_id; если какую-либо из таблиц не удалось найти.

Input:

SessionId (string)

Output:

{

“Status”: (“Success” / “Failure”)

“Statistics”:

[

{

“Username”: (string)

“XP”: (int)

“Level”: (int)


```
}  
]  
}
```

12. logout.py

Сервер просит найти в БД поле session_id таблицы Sessions, значение которого совпадает с переданным id сессии. БД удаляет найденную таблицу.

Сервер выдает ошибку, если не существует таблицы Sessions с данным session_id; если ее не удалось удалить.

Input:

SessionId (string)

Output:

```
{  
"Status":      ("Success" / "Failure")  
}
```

13. unreg.py

Сервер просит найти в БД поле session_id таблицы Sessions, значение которого совпадает с переданным id сессии. По значению поля user_id в БД находится таблица Users, в которой это значение совпадает со значением поля id таблицы Sessions. По значению поля id таблицы Users находятся поля user_id таблиц Timers, User_Level, W_To_Learn, W_To_Repeat, Progress, Session (значения полей id которых не совпадают с переданным значением id сессии). Эти таблицы удаляются. Затем удаляется найденная таблица Users и таблица Session (значение поля id которой совпадает с переданным значением id сессии).

Сервер выдает ошибку, если не удалось найти или удалить какие-либо таблицы.

Input:

SessionId (string)

Output:

```
{  
"status":      ("Success" / "Failure")  
}
```

Описание сценария использования менеджером

Backup and restore

цель: сделать резервную копию системы;

актеры: менеджер, система;

предварительные условия: менеджер зарегистрирован, как администратор и менеджер вошел в систему;

активатор: нажатие на кнопку "Резервная копия";

последовательность событий:

1. система открывает новое окно с выбором пути для сохранения резервной копии;
2. менеджер указывает путь и нажимает "ok";
3. система сохраняет резервную копию.

цель: восстановить систему;

актеры: менеджер, система;

предварительные условия: менеджер зарегистрирован, как администратор и менеджер вошел в систему;

активатор: нажатие на кнопку "Восстановление системы";

последовательность событий:

1. система открывает новое окно с выбором пути для загрузки резервной копии;
2. менеджер указывает путь и нажимает "ok";
3. система восстанавливает систему по выбранной резервной копии.

Анализ требований

Основные определения:

Вид словаря:

Язык1 Язык2

Уровень_на_языке1 уровень_на_языке_2

Тема_на_языке1 тема_на_языке2

Слово_на_языке1 транскрипция Слово_на_языке2 транскрипция

Аккаунт - имя пользователя (строка), пароль (строка), список выученных им слов (список указателей на слова из таблицы), его уровень (число: Beginner - 1, ... Advanced - 5), хр (число), родной язык (строка), изучаемый язык (строка), настройки таймера (время (время), день (дата), количество повторяемых слов (число))

В **базе данных** хранятся аккаунты всех пользователей, аккаунт менеджера, словарь.

Урок - Показ карточек, содержащих слова на английском - перевод на русский - возможность прослушать слово, возможность сказать, что ты уже знаешь это слово. После просмотра карточек, пользователю предлагается закрепить знания, пройдя тест.

Тест - 1) первичный - тест на перевод на 20 слов из словарей разных уровней.

2) для закрепления - для нескольких слов используются все модули теста.

3) для повторения - для нескольких слов используются только один из модулей теста (выбор случайный).

Статистика - кол-во выученных слов и кол-во слов, находящихся на изучении у данного пользователя.

Модули теста:

1. Проверка перевода:

Для английского слова выводится 4 слова на русском, надо выбрать правильный перевод.

2. Проверка написания:

Дается слово на русском, набор букв, из которых состоит слово на английском (например, для слова кошка t, a, c). Пользователь выбирает буквы в правильном порядке (cat).

3. Проверка восприятие на слух:

Пользователь прослушивает слово на английском и пишет перевод на русский.

Возможности:

1. тест на проверку уровня английского,
2. словарные карточки - это собственно само обучение (русский-английский, английский-русский),
3. аккаунт, к которому привязываются твои достижения,
4. может спрашивать слова по таймеру или по желанию пользователя,
5. можно следить за своим прогрессом (сколько слов ты выучил, сколько на изучении),
6. приложение работает онлайн, но есть возможность загрузки слов для офлайн обучения,
7. система поощрения и рейтинг (хр рейтинг)
8. добавление такой тренировки: берется 5-6 слов, сначала знакомишься с этими словами (слово, перевод, транскрипция (по желанию), аудио), далее учишься переводить слова на русский, писать их на английском, а затем по произношению написать слово снова на английском,
9. повтор уже изученных слов, через какое-то время (недавно изученные слова – через сутки, далее если правильно отвечаешь, то через 2-3 дня, если неправильно, то через сутки),
10. слова распределены по сложности и по темам, возможность составить свой список слов для изучения из словаря
11. опечатки рассматриваться не будут. опечатка=ошибка.

Описание сценария использования пользователем

Войти и выйти из системы

Вход в аккаунт:

- Пользователь выбирает пункт "Вход".

Обычный вход:

- Пользователь вводит никнейм;
- Пользователь вводит пароль;
- Пользователь производит вход.

Вход в систему осуществляется с помощью аккаунта других социальных сетей (vk, facebook, одноклассники)

Вход через сторонний аккаунт:

- Пользователь выбирает сервис, через который он зарегистрирован в системе;
- Пользователь завершает вход.

Выход из аккаунта:

- Пользователь выбирает пункт "Выход";
- Пользователь завершает Выход.

Описание сценария использования менеджером

Добавить словарь

цель: добавить в программу словарь;

актеры: менеджер, система;

предварительные условия: менеджер зарегистрирован, как администратор и менеджер вошел в систему;

активатор: нажатие на кнопку "добавить словарь";

последовательность событий:

1. система открывает новое окно с кнопками "Добавить новую тему", "Добавить слова в тему";

2.1.1 Если менеджер нажимает "Добавить новую тему", то система открывает новое окно с поиском темы;

2.1.2 Менеджер пишет тему в строку поиска;

2.1.3 Система возвращает менеджеру окно с таблицей из 4-х колонок;

2.1.4 менеджер пишет словарь в виде:

Слово_на_языке1 транскрипция Слово_на_языке2 транскрипция, нажимает "ok";

2.2.1 Если менеджер нажимает "Добавить слова в тему", то система открывает окно со строками названия темы на языке_1 и на языке_2, уровнем на языке_1 и на языке_2 и таблицей из 4-х колонок;

2.2.2 менеджер пишет название и словарь в виде: Тема_на_языке1 тема_на_языке2

Слово_на_языке1 транскрипция Слово_на_языке2 транскрипция, нажимает "ok";

3.2.3 система вносит словарь в базу.

Описание сценария использования пользователем

Зарегистрировать и удалить пользователя

Регистрация:

- Пользователь выбирает пункт "Регистрация".

Обычная регистрация:

- Пользователь вводит свой никнейм;

- Пользователь вводит свою почту;

- Пользователь вводит свой пароль;

- Пользователь вводит капчу;

- На почту пользователя приходит запрос подтверждения;

- Пользователь подтверждает почту;

- Пользователь завершает регистрацию.

Регистрация через сторонний аккаунт:

- Пользователь выбирает один из сервисов, в котором у него есть аккаунт;

- Пользователь вводит капчу;

- Пользователь завершает регистрацию.

Удаление аккаунта:

- Пользователь выбирает пункт "Удалить аккаунт";

- Пользователь подтверждает удаление;

- Пользователь завершает удаление.

Восстановление пароля

цель: восстановить пароль пользователя;

актеры: пользователь, система;

предварительные условия: пользователь зашел в приложение;

активатор: нажатие на кнопку "Забыл пароль";

последовательность событий:

1. система высылает письмо со ссылкой на смену пароля на почту пользователя;
2. пользователь переходит по ссылке, вводит новый пароль;
3. система высылает на почту пользователя письмо с подтверждением смены пароля;

Описание сценария использования пользователем

Настроить аккаунт (выбор языков, выбор уровня, таймер)

Настроить аккаунт при регистрации (выбор языков, выбор уровня)

цель: настроить обучение под конкретного пользователя;

актеры: пользователь, система;

предварительные условия: пользователь зарегистрирован и пользователь вошел в систему;

активатор: вход в систему первый раз;

последовательность событий:

1. система спрашивает у пользователя его родной язык и язык, который он хочет изучать;
2. пользователь выбирает языки, нажимает "ok";
3. система спрашивает у пользователя хочет ли он сам указать свой уровень владения языком или же пройти входное тестирование;
- 4.1. если пользователь выбирает первый вариант, то система выдает ему список уровней, пользователь выбирает, нажимает "ok";
- 4.2. если пользователь выбирает второй вариант, то система отправляет его проходить тест, пользователь проходит тест, система возвращает ему результат, пользователь нажимает "ok";

альтернативные пути: нет;

результатирующее условие после выполнения: нажатие на кнопку "ok" (окно закрывается, пользователь попадает на начальную страницу).

Настроить аккаунт (выбор языков, таймер)

цель: изменить процесс обучения;

актеры: пользователь, система;

предварительные условия: пользователь зарегистрирован и пользователь вошел в систему;

активатор: нажатие на кнопку "настройки";

последовательность событий:

1. система открывает пользователю новое окно с разделами "изменить языки", "установить таймер";

2.1. пользователь выбирает "изменить родной язык", изменяет, нажимает "ok";

2.2. пользователь выбирает "изменить изучаемый язык", изменяет, нажимает "ok";

2.3. пользователь выбирает "включить" (таймер), система пишет: «Таймер дает вам возможность повторять слова в заданное вами время, вне зависимости от нахождения в приложении. Если в это время вы окажитесь заняты, вы сможете отложить его.», пользователь выбирает кол-во повторяемых слов, тему (опционально), время и дни недели, нажимает "ok";

альтернативные пути: нет;

результатирующее условие после выполнения: нажатие на кнопку "ok" (окно закрывается, пользователь попадает на начальную страницу).

Описание сценария использования пользователем

Пройти сессию обучения (урок, тест)

цель: обучение;

актеры: пользователь, система;

предварительные условия: пользователь зарегистрирован и пользователь вошел в систему;

активатор: нажатие на кнопку "выучить новые слова";

последовательность событий:

1. система открывает пользователю новое окно с уроком;

2. пользователь знакомится со словами, если он нажимает «уже знаю», то система заменяет слово на новое, нажимает "ok";

3. система отправляет пользователя пройти тест;

4. пользователь проходит тест, нажимает "ok".

5. Система выдает пользователю результат (кол-во правильных слов – число, кол-во неправильных слов- число, правильные ответы). Выученные слова отправляются на повторение, остальные на повторное изучение;

альтернативные пути: нет;

результатирующее условие после выполнения: нажатие на кнопку "ok" (окно закрывается, пользователь попадает на начальную страницу).

Описание сценария использования менеджером

Удалить из словаря

цель: удалить из словаря некоторое количество слов;

актеры: менеджер, система;

предварительные условия: менеджер зарегистрирован, как администратор и менеджер вошел в систему;

активатор: нажатие на кнопку "удалить из словаря";

последовательность событий:

1. система открывает новое окно с поиском;

2. менеджер выбирает язык слова из списка, пишет в строке поиска слово, которое хочет удалить, нажимает "ok";

3. система возвращает ему список вида "слово-перевод";

4. менеджер выделяет, какие из переводов этого слова нужно удалить, либо нажимает на кнопку "выделить все", нажимает на кнопку "удалить";

5. система удаляет выбранные переводы.

