**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа бакалавриата «Программная инженерия»

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Научный руководитель,  Преподаватель департамента  программной инженерии    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н. Степанов  «21» апреля 2022 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  профессор департамента  программной инженерии,  канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Серверная часть.**  **Программа и методика испытаний**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.06.02-01 51 01-1-ЛУ** | | |
|  |  | |
| Исполнитель:  студент группы БПИ191  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Царёв К.Д. /  «21» апреля 2022 г. | |
|  | | |
|  | |  |

**Москва 2022**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.04.01-01 51 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Программа и методика испытаний**  **RU.17701729.06.02-01 51 01-1**  **Листов 31** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2022**

**АННОТАЦИЯ**

Программа и методика испытаний – это документ, в котором содержится информация о программном продукте, а также полное описание приемочных испытаний для данного программного продукта.

Настоящая программа и методика испытаний для серверной части приложения для обмена навыками и услугами для студентов НИУ ВШЭ содержит следующие разделы: «Объект испытаний», «Цель испытаний», «Требования к программе», «Требования к программным документам», «Средства и порядок испытаний», «Приложения».

В разделе «Объект испытаний» указано наименование и область применения серверной части приложения для обмена навыками и услугами для студентов НИУ ВШЭ.

В разделе «Цель испытаний» указана цель проведения испытаний.

Раздел «Требования к программе» содержит основные требования к программе, которые подлежат проверке во время испытаний.

Раздел «Требования к программным документам» содержит состав программной документации, которая представляется на испытания, а также специальные требования к ней.

Раздел «Средства и порядок испытаний» содержит информацию о технических и программных средствах, которые следует использовать во время испытаний, а также порядок этих испытаний.

Раздел «Методы испытаний» содержит информацию об используемых методах испытаний. Настоящий документ разработан в соответствии с требованиями:

1) ГОСТ 19.101-77 Виды программ и программных документов;

2) ГОСТ 19.102-77 Стадии разработки;

3) ГОСТ 19.103-77 Обозначения программ и программных документов;

4) ГОСТ 19.104-78 Основные надписи;

5) ГОСТ 19.105-78 Общие требования к программным документам;

6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом;

7) ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению.

Изменения к данному документу оформляются согласно ГОСТ 19.603-7, ГОСТ 19.604- 78.

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 настоящей программы и методики испытаний.

ОГЛАВЛЕНИЕ

[1. ОБЪЕКТ ИСПЫТАНИЙ 5](#_Toc100014048)

[1.1. Наименование 5](#_Toc100014049)

[1.2. Область применения 5](#_Toc100014050)

[1.3. Обозначение испытуемой программы 5](#_Toc100014051)

[2. ЦЕЛЬ ИСПЫТАНИЙ 6](#_Toc100014052)

[3. ТРЕБОВАНИЯ К ПРОГРАММЕ 7](#_Toc100014053)

[3.1. Требования к функциональным характеристикам 7](#_Toc100014054)

[3.1.1. Требования к составу выполняемых функций 7](#_Toc100014055)

[3.1.2. Требования к организации входных данных 7](#_Toc100014056)

[3.1.3. Требования к организации выходных данных 7](#_Toc100014057)

[3.2. Требования к надежности 8](#_Toc100014058)

[3.3. Требования к интерфейсу 8](#_Toc100014059)

[4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 9](#_Toc100014060)

[4.1. Состав программной документации 9](#_Toc100014061)

[4.2. Специальные требования к программной документации 9](#_Toc100014062)

[5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ 10](#_Toc100014063)

[5.1. Технические средства, используемые во время испытаний 10](#_Toc100014064)

[5.2. Программные средства, используемые во время испытаний 10](#_Toc100014065)

[5.3. Порядок проведения испытаний 10](#_Toc100014066)

[5.4. Условия проведения испытаний 10](#_Toc100014067)

[5.4.1. Климатические условия 10](#_Toc100014068)

[5.4.2. Требования к численности и квалификации персонала 10](#_Toc100014069)

[6. МЕТОДЫ ИСПЫТАНИЙ 11](#_Toc100014070)

[6.1. Испытание выполнения требований к программной документации 11](#_Toc100014071)

[6.2. Испытание выполнения требований к интерфейсу 11](#_Toc100014072)

[6.3. Испытание выполнения требований к функциональным характеристикам 11](#_Toc100014073)

[6.3.1. Регистрация нового пользователя с помощью корпоративной почты НИУ ВШЭ 11](#_Toc100014074)

[6.3.2. Авторизация пользователя 12](#_Toc100014075)

[6.3.3. Добавление нового навыка 12](#_Toc100014076)

[6.3.4. Поиск навыков 12](#_Toc100014077)

[6.3.5. Создание обмена 12](#_Toc100014078)

[6.3.6. Завершение обмена 12](#_Toc100014079)

[6.3.7. Напоминание забытого пароля 12](#_Toc100014080)

[6.3.8. Оценка пользователей 12](#_Toc100014081)

[6.3.9. Присваивание пользователю статуса модератора 13](#_Toc100014082)

[6.3.10. Цензура текста 13](#_Toc100014083)

[6.3.11. Удаление отзывов 13](#_Toc100014084)

[6.3.12. Получение переписок пользователя 14](#_Toc100014085)

[6.3.13. Реализация внутреннего чата 14](#_Toc100014086)

[6.3.14. Получение частичной истории сообщений в чате 14](#_Toc100014087)

[6.3.15. Обмен сообщениями в реальном времени 15](#_Toc100014088)

[6.3.16. Отправка уведомлений 15](#_Toc100014089)

[ПРИЛОЖЕНИЕ 1 16](#_Toc100014090)

[ТЕРМИНОЛОГИЯ 16](#_Toc100014091)

[ПРИЛОЖЕНИЕ 2 17](#_Toc100014092)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 17](#_Toc100014093)

[ПРИЛОЖЕНИЕ 3 18](#_Toc100014094)

[ИДЕНТИФИКАТОРЫ ЛИЧНЫХ ДАННЫХ И КАТЕГОРИЙ НАВЫКОВ 18](#_Toc100014095)

[ПРИЛОЖЕНИЕ 4 21](#_Toc100014096)

[ОГРАНИЧЕНИЯ, НАКЛАДЫВАЕМЫЕ НА ДАННЫЕ, 21](#_Toc100014097)

[ХРАНИМЫЕ В БАЗЕ ДАННЫХ 21](#_Toc100014098)

[ПРИЛОЖЕНИЕ 5 25](#_Toc100014099)

[ОПИСАНИЕ И НАЗНАЧЕНИЕ МЕТОДОВ В СООТВЕТСТВИИ С REST API 25](#_Toc100014100)

[ПРИЛОЖЕНИЕ 6 27](#_Toc100014101)

[ОПИСАНИЕ СЕРВЕРНЫХ ОШИБОК, КОТОРЫЕ ВОЗВРАЩАЮТ HTTP МЕТОДЫ 27](#_Toc100014102)

[ПРИЛОЖЕНИЕ 7 29](#_Toc100014103)

[АВТОМАТИЧЕСКИ ГЕНЕРИРУЕМЫЕ ПИСЬМА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ 29](#_Toc100014104)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 31](#_Toc100014105)

# ОБЪЕКТ ИСПЫТАНИЙ

## Наименование

Наименование программы – Серверная часть приложения для обмена навыками и услугами для студентов НИУ ВШЭ.

## Область применения

Приложение “Sharing Skills HSE” может быть использовано студентами НИУ ВШЭ, желающими совершить обмен навыками/услугами с человеком из сообщества НИУ ВШЭ.

Серверная часть приложения предназначена для:

* Реализации внутренней логики приложения
* Обеспечения взаимодействия с базой данных
* Предоставления своего API для использования клиентской частью приложения

Область применения программы – сфера знакомств и услуг.

## Обозначение испытуемой программы

Краткое наименование программы – «Sharing Skills HSE».

# ЦЕЛЬ ИСПЫТАНИЙ

Цель проведения испытаний - проверка соответствия характеристик разработанной программы функциональным требованиям и отдельным требованиям к надежности, изложенным в Техническом задании к данной программе.

# ТРЕБОВАНИЯ К ПРОГРАММЕ

## Требования к функциональным характеристикам

### Требования к составу выполняемых функций

Программа должна обеспечивать возможность выполнения следующих функций:

1. Регистрация пользователя в сервисе с использованием корпоративной почты НИУ ВШЭ в домене edu.hse.ru
2. Поддержание актуальности личных данных в профилях пользователей
3. Поддержание актуальности списков обменов пользователей

В частности:

* Текущих
* Завершённых

1. Поддержание актуальности списков предложенных обменов пользователей

В частности:

* Входящих
* Исходящих

1. Предоставление клиенту результатов поиска по указанным навыкам

И опционально по следующим фильтрам:

* Корпус
* Факультет
* Курс
* Общежитие
* Пол
* “Может” / “Хочет”
* Категория
* Подкатегория

1. Отправка уведомлений пользователям

В частности:

* Уведомления о предложении обмена
* Уведомления об одобрении обмена
* Уведомления об отказе в обмене
* Уведомления о завершении обмена
* Уведомления о новом сообщении в чате
* Уведомления о новом отзыве
* Уведомления о присваивании статуса модератора

1. Поддержание актуальности информации в чате
2. Поддержание актуальности оценок и комментариев пользователей
3. Автоматическая цензура комментариев на предмет наличия нецензурной лексики
4. Возможность удаления неприемлемых комментариев при наличии у пользователя статуса модератора

### Требования к организации входных данных

Программа должна обрабатывать HTTP-запросы от клиентской части приложения.

### Требования к организации выходных данных

В качестве выходных данных программа должна отправлять HTTP-ответы на запросы клиентской части приложения, а также автоматически генерируемые электронные письма для пользователей.

## Требования к надежности

Система должна корректно обрабатывать неверные запросы любого вида и выдавать информативные сообщения об ошибках.

## Требования к интерфейсу

Клиент и сервер должны взаимодействовать между собой по REST Api.

# ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

## Состав программной документации

На испытания должна быть представлена документация к программе в следующем составе:

1. Техническое задание «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ.» (ГОСТ 19.201-78)
2. Программа и методика испытаний «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ.» (ГОСТ 19.301-79)
3. Техническое задание «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Серверная часть.» (ГОСТ 19.201-78)
4. Программа и методика испытаний «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Серверная часть.» (ГОСТ 19.301-79)
5. Пояснительная записка «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Серверная часть.» (ГОСТ 19.404-79)
6. Текст программы «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Серверная часть.» (ГОСТ 19.401-78)
7. Руководство программиста «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Серверная часть.» (ГОСТ 19.504-79)

## Специальные требования к программной документации

1. Все документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТ к этому виду документа (см. п. 5.1.).
2. Пояснительная записка должна быть загружена в систему Антиплагиат через Smart LMS (Learning management system) НИУ ВШЭ.
3. Вся документация и программа сдается в электронном виде в формате .pdf или .docx. в архиве формата .rar или .zip.
4. Все документы перед защитой курсовой работы должны быть загружены в информационно-образовательную среду НИУ ВШЭ Smart LMS (Learning management system) в личном кабинете, дисциплина - «Курсовой проект, 3 курс».

# СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

## Технические средства, используемые во время испытаний

Для надёжной и бесперебойной работы программы требуется следующий состав технических средств:

* 1. Минимальный размер хранилища базы данных 32 гб
  2. Минимальный размер оперативной памяти 1 гб
  3. Двухъядерный процессор 2 ггц

## Программные средства, используемые во время испытаний

Для надёжной и бесперебойной работы программы требуется следующий состав программных средств:

1. 32-битная операционная система Windows

## Порядок проведения испытаний

1) Проверка требований к программной документации;

2) Проверка требований к интерфейсу;

3) Проверка требований к надежности;

4) Проверка требований к функциональным характеристикам.

## Условия проведения испытаний

### Климатические условия

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к персональным компьютерам и компакт-дискам (CD) в части условий их эксплуатации.

Персональный компьютер предназначен для работы в закрытом отапливаемом помещении со стабильными климатическими условиями категории 4.1 согласно ГОСТ 15150-69.

Для предотвращения повреждений персонального компьютера и сохранения на нем записанной информации необходимо поддерживать следующие климатические условия:

1. влажность от 20% до 70%;
2. температура от 5°C до 30°C;
3. атмосферное давление — от 84 до 106,7 кПа (от 630 до 800 мм рт. ст).

### Требования к численности и квалификации персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 1 человека – системного администратора, знающего язык программирования C# и имеющего навыки работы с ASP.Net.

# МЕТОДЫ ИСПЫТАНИЙ

Испытания представляют собой процесс установления соответствия программы и программной документации заданным требованиям.

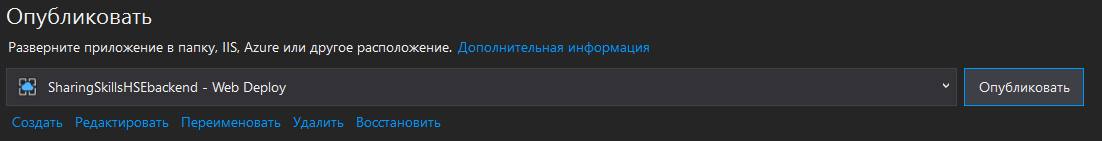
Перед проведением испытаний необходимо опубликовать приложение на портале Microsoft Azure. Для использования серверной части клиентами необходимо создать следующие ресурсы (рис. 1).

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 1*

После создания необходимых ресурсов производится непосредственно публикация через Visual Studio (рис. 2).



*Рисунок 2*

Теперь можно узнать URL-адрес службы приложений и приступить к тестированию серверной части (<https://sharingskillsapp.azurewebsites.net>).

Все методы контроллеров, возможные ошибки, ограничения на данные, хранимые в базе данных можно посмотреть в соответствующих приложениях данной Программы и Методики Испытаний.

## Испытание выполнения требований к программной документации

Состав программной документации проверяется визуально, проверяется наличие всех подписей и наличие программной документации в системе smart LMS. Также визуально проверяется соответствие документации требованиям ГОСТ. Все документы удовлетворяют представленным требованиям.

## Испытание выполнения требований к интерфейсу

Взаимодействие клиентов и сервера осуществляется по REST Api, так как используются только HTTP-методы GET, PUT, POST, DELETE, присутствует четкое разграничение между клиентом и сервером, а также сервер отправляет статические представления ресурсов в формате JSON.

## Испытание выполнения требований к функциональным характеристикам

### Регистрация нового пользователя с помощью корпоративной почты НИУ ВШЭ

Для регистрации нового пользователя с клиента отправляется POST-запрос, содержащий адрес корпоративной почты в домене @edu.hse.ru, который проверяет наличие такого пользователя в базе данных, и если такого пользователя нет, то добавляет его, а также отправляет автоматически сгенерированное письмо с кодом подтверждения на указанную почту. Затем код подтверждения и личные данные пользователя отправляются на сервер с помощью PUT-запросов.

Для более простого взаимодействия сервера и клиентов (чтобы сравнивать целые числа, а не строки), личные данные, одинаковые для всех пользователей (пол, курс, факультет и тд), имеют собственные идентификаторы, с которыми можно ознакомиться в приложении 3.

### Авторизация пользователя

GET-запросом клиент получает пароль, хранящийся на сервере, и сравнивает его с паролем, введённым пользователем в приложении. Такой примитивный процесс авторизации был реализован в связи с тем, что в приложении не используются никакие строго конфиденциальные данные.

### Добавление нового навыка

Новый навык добавляется одним POST-запросом, в котором вводятся сразу все данные (название, описание, категория, подкатегория). В дальнейшем можно изменить навык с помощью PUT-запроса, или удалить его с помощью DELETE-запроса.

### Поиск навыков

Поиск навыков осуществляется с помощью одного GET-запроса, который последовательно применяет к вообще всем навыкам отправленные фильтры, а также удаляет из результата навыки пользователя, который осуществляет поиск. В результате на клиент приходит список навыков, сформированный в соответствии с введенными фильтрами.

### Создание обмена

Когда один пользователь (отправитель) предлагает другому пользователю (получатель) совершить обмен навыками, на сервер посылается один POST-запрос с информацией об обмене (навыки, описание и тд), и получателю отправляется уведомление о новом предложенном обмене. Пока получатель не принял обмен, отправитель может изменять его данные с помощью PUT-запросов. Если получатель отказывается от обмена, отправителю приходит соответствующее уведомление и обмен удаляется из базы данных, а если соглашается – то отправителю также приходит уведомление.

### Завершение обмена

Для завершения обмена со стороны одного из пользователей, участвующих в нем, посылается PUT-запрос на сервер, в котором изменяется статус обмена на “завершенный”, а второму пользователю отправляется уведомление о завершении обмена.

### Напоминание забытого пароля

Для напоминания забытого пароля с клиента отправляется GET-запрос, в котором на указанную почту отправляется автоматически сгенерированное письмо, в котором указывается забытый пароль.

### Оценка пользователей

В случае, когда один пользователь хочет оценить другого, происходят следующие действия. Сначала пользователь на клиенте заполняет данные в полях ввода (оценку и комментарий). Не будем вдаваться в подробности, как это происходит, потому что это не имеет отношения к серверной части. Затем, опять же на клиенте, из всех данных формируется экземпляр сущности отзыва (Feedback), и выполняется POST-запрос (PostFeedback).

В этом методе происходит следующая последовательность действий. Сначала по автоматически сгенерированному ID отзыва проверяется, нет ли уже такого отзыва в базе данных. Затем по переданным почтам находятся отправитель и получатель, и проверяется, что они существуют в базе данных, и что это разные экземпляры сущности пользователя. Затем происходит автоматическая цензура комментария, вычисление и обновление новой средней оценки получателя. Отзыв добавляется в базу данных. В случае удовлетворения некоторых условий, пользователю присваивается статус модератора. Происходит уведомление получателя.

### Присваивание пользователю статуса модератора

В данном приложении присутствует система отзывов и оценок пользователей, которая включает в себя написание текстовых комментариев. Для борьбы с неприемлемыми комментариями был реализован класс для автоматической цензуры текста. Но так как любую машину можно обойти, было принято решение, в помощь авто цензуре добавить возможность ручной цензуры самими пользователями. Для этого им необходимо получить статус модератора.

Для того, чтобы получить статус модератора, пользователю не нужно ничего делать. Достаточно просто пользоваться приложением и получать отзывы от других пользователей. В POST-запросе (PostFeedback) каждый раз при добавлении нового отзыва для получателя отзыва проверяются условия (количество оценок больше 5 и средняя оценка больше 3.5). При удовлетворении условий, у пользователя меняется флаг (IsModer) на true, и ему становятся доступны функции модератора.

### Цензура текста

Для реализации алгоритма цензуры текста был создан соответствующий программный класс (Censorship). Хочется обратить внимание, что цензура работает только на русский язык.

Для начала рассмотрим внутренние поля этого класса. В нем присутствует список запрещенных нецензурных слов, которые будут изменяться на звездочки. Но так как человек существо хитрое, он может изменять некоторые буквы на другие символы, чтобы цензура не срабатывала (например, а = @). Такой вариант тоже предусмотрен нашим алгоритмом, и для этого в классе присутствует словарь с вариантами написания одних и тех же букв.

Теперь перейдем к методу, выполняющему непосредственную цензуру текста (DoCensorship). В качестве входного параметра ему передается строка для цензуры. Сначала производятся подготовительные действия (текст переводится в нижний регистр, и все буквы, имеющие похожее написание, меняются на буквы из русского алфавита). Затем для каждого нецензурного слова из списка проходимся по отрезкам текста, равным длине этого слова. Если отрезок оказался запрещенным словом, то заменяем его на звездочки (\*\*\*).

Но так как люди могут слегка изменять сами слова (например, банан = бонан), необходимо как-то обрабатывать и такие ситуации. Для этого пришлось находить расстояние Левенштейна, которое показывает степень похожести слов. Таким образом, мы считаем, что слово запрещено, если расстояние Левенштейна меньше 25% длины слова.

Затем все измененные символы (за исключением звездочек) возвращаются в первоначальный вариант (заглавные буквы, специальные символы), и зацензуренный текст возвращается.

Данная автоматическая цензура используется в комментариях отзывов и в графе “О себе” личных данных пользователя.

### Удаление отзывов

Процесс удаления отзыва реализован в двух видах:

* Удаление отзыва пользователем, оставившим этот отзыв. Данный вид удаления реализован с помощью DELETE-запроса (DeleteFeedbackUser). В этом методе происходит следующая последовательность действий. Сначала по переданному ID в базе данных находится отзыв. Затем происходит проверка на соответствие переданной в запрос почты и хранящейся в отзыве почты отправителя (SenderMail). Это сделано для того, чтобы пользователи не могли удалять чужие отзывы. Затем мы находим получателя отзыва, удаляем сам отзыв, изменяем значение средней оценки и сохраняем изменения в базе данных.
* Удаление отзыва модератором. Данный вид удаления реализован с помощью DELETE-запроса (DeleteFeedbackModer). В этом методе происходит следующая последовательность действий. Сначала по переданному ID в базе данных находится отзыв. Затем происходит проверка на соответствие переданной в запрос почты и хранящейся в отзыве почты получателя (ReceiverMail). Это сделано для того, чтобы модераторы не могли удалять отзывы, оставленные им. Затем мы находим получателя отзыва, удаляем сам отзыв, изменяем значение средней оценки и сохраняем изменения в базе данных.

### Получение переписок пользователя

Для того, чтобы со стороны клиентов на экране чатов отображались все переписки конкретного пользователя, в серверную часть была добавлена сущность переписки (Chat). При первоначальном создании экземпляра переписки выполняется POST-запрос (PostChat), в котором передаются данные о пользователях, необходимые для отображения на экране чатов.

При отправке сообщения в сущности чата автоматически обновляются поля последнего сообщения (LastMessage) и времени отправки (SendTime), без дополнительных PUT-запросов.

Для получения всех чатов конкретного пользователя выполняется GET-запрос (GetChats) с указанием почты пользователя.

### Реализация внутреннего чата

Глобально, внутренний чат приложения будет организован следующей последовательностью действий:

* Пользователь заходит в конкретную переписку.
* На клиенте вызывается GET-запрос (GetMessages), который подгружает историю сообщений в данном конкретном чате.
* Клиент подключает пользователя к хабу чатов для непосредственного обмена сообщениями в режиме реального времени.
* Если второй пользователь из чата тоже в данный момент подключен к хабу, то происходит обмен сообщениями, и сообщения добавляются в базу данных POST-запросом (PostMessage).
* Если второй пользователь не подключен, то при отправке сообщений первым, второму будут приходить уведомления о новых сообщениях. А когда второй зайдет в чат, ему подгрузится история сообщений из базы данных.

При отправке любых сообщений они добавляются в базу данных с флагом прочитанности (IsRead) равным false. А затем, когда получатель прочитает их, с его клиента PUT-запросами (PutMessage) флаги прочитанности будут изменяться на true.

### Получение частичной истории сообщений в чате

Для ускорения получения истории сообщений в чате было принято решение подгружать только 100 последних сообщений, чего должно быть более чем достаточно. А при необходимости подгружать следующие 100 и так далее.

Получение частичной истории сообщений в чате реализовано в GET-запросе (GetMessages). В этом методе в качестве входных параметров передаются почты двух участников чата, и число, означающее количество сотен сообщений, которые необходимо пропустить. Сначала по переданным почтам в базе данных находятся пользователи. (Они должны существовать, и должны быть разными.) Затем среди всех сообщений находятся сообщения только между этими двумя. Потом они сортируются по времени отправки, пропускаются n\*100 сообщений, где n – входной параметр, и возвращаются 100 сообщений.

### Обмен сообщениями в реальном времени

Обмен сообщениями в чатах в реальном времени реализован с использованием библиотеки SignalR. Для этого был написан программный класс хаба чатов (ChatHub).

Сначала клиент должен подключиться к хабу с использованием данного URL адреса (<https://sharingskillsapp.azurewebsites.net/chat>). При подключении в список подключенных клиентов добавляется новая сущность подключенного пользователя (UserChatInfo) с заполненным ID подключения, но не заполненным значением почты. Для сопоставления ID подключения и почты пользователя, на клиенте вызывается соответствующий метод хаба (SetMail). Теперь при обмене сообщениями, пользователи ищутся в списке подключенных, и при наличии получателя в этом списке, ему отправляется сообщение. При выходе из чата (при отключении от хаба) пользователь удаляется из списка.

### Отправка уведомлений

Изначально планировалось осуществлять отправку уведомлений в виде PUSH-уведомлений с использованием библиотеки PushSharp. Но из-за того, что клиент реализован только под iOS платформу, пришлось отказаться от такой идеи, потому что Apple разрешает отправлять PUSH-уведомления только через свои сервисы. А для этого необходимо завести Apple Developer аккаунт, подписка на который стоит 99 долларов в год, что является очень крупной суммой для учебного проекта.

Таким образом, отправка всех уведомлений происходит с помощью автоматически генерируемых писем, отправляемых с почты ([sharingskills@sharingskills.xyz](mailto:sharingskills@sharingskills.xyz)) через SMTP протокол с использованием библиотеки MailKit.

Уведомления в приложении отправляются в следующих случаях:

* Уведомления о предложении обмена
* Уведомления об одобрении обмена
* Уведомления об отказе в обмене
* Уведомления о завершении обмена
* Уведомления о новом сообщении в чате
* Уведомления о новом отзыве
* Уведомления о присваивании статуса модератора

Примеры автоматически генерируемых писем для отправки на почту можно посмотреть в приложении 7.

# ПРИЛОЖЕНИЕ 1

### ТЕРМИНОЛОГИЯ

Ниже приведен список необходимых терминов для ознакомления.

**Клиент** – аппаратный или программный компонент вычислительной системы, посылающий запросы серверу.

**Сервер** – программный компонент вычислительной системы, выполняющий сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определённым ресурсам или услугам.

**Навык** – что-то, что умеет пользователь.

**Обмен** – условная договоренность между двумя пользователями о том, что они готовы поделиться друг с другом своими навыками.

**Корпоративная почта** – это электронная почта на домене вашей компании.

**API** (Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

**HTTP** (HyperText Transfer Protocol) – протокол прикладного уровня передачи произвольных данных.

**SMTP** (Simple Mail Transfer Protocol) – это широко используемый [сетевой протокол](https://ru.wikipedia.org/wiki/IP), предназначенный для передачи [электронной почты](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%87%D1%82%D0%B0).

# ПРИЛОЖЕНИЕ 2

### СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способ
10. Клиент-сервер [Электронный ресурс] / Википедия. Режим доступа: <https://ru.wikipedia.org/wiki/Клиент_—_сервер>, свободный. (дата обращения: 20.11.2021).
11. Введение в REST API — RESTful веб-сервисы [Электронный ресурс] / Хабр. Режим доступа: <https://habr.com/ru/post/483202/>, свободный. (дата обращения: 09.10.2021).
12. Deploy an ASP.NET Core App with EF Core and SQL Server to Azure [Электронный̆ ресурс] / medium. Режим доступа: <https://medium.com/net-core/deploy-an-asp-net-core-app-with-ef-core-and-sql-server-to-azure-e11df41a4804>, свободный. (дата обращения: 15.12.2021).
13. Руководство по ASP.NET Core 5 [Электронный ресурс] / metanit. Режим доступа: <https://metanit.com/sharp/aspnet5>, свободный. (дата обращения: 29.12.2021).
14. PushSharp — простая работа с Push Notifications [Электронный ресурс] / Хабр. Режим доступа: <https://habr.com/ru/post/178775/>, свободный. (дата обращения: 09.01.2022).
15. iOS and Android Push Notifications with PushSharp [Электронный ресурс] / Matt Burke. Режим доступа: <https://www.mattburkedev.com/ios-and-android-push-notifications-with-pushsharp/>, свободный. (дата обращения: 12.01.2022).
16. Introduction to SignalR [Электронный ресурс] / Microsoft Docs. Режим доступа: <https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr>, свободный. (дата обращения: 01.02.2022).

# ПРИЛОЖЕНИЕ 3

## ИДЕНТИФИКАТОРЫ ЛИЧНЫХ ДАННЫХ И КАТЕГОРИЙ НАВЫКОВ

Курс:

1 – бакалавриат 1 курс

2 – бакалавриат 2 курс

3 – бакалавриат 3 курс

4 – бакалавриат 4 курс

5 – бакалавриат 5 курс

6 – магистратура 1 курс

7 – магистратура 2 курс

Образовательная программа:

1 – Городское планирование

2 - [Клеточная и молекулярная биотехнология](https://www.hse.ru/ba/cmb/)

3 - [Арабистика: язык, словесность, культура](https://www.hse.ru/ba/arabic/)

4 - [Востоковедение](https://www.hse.ru/ba/oriental/)

5 - [Христианский Восток](https://www.hse.ru/ba/christian/)

6 - [Язык, словесность и культура Китая](https://www.hse.ru/ba/chinese/)

7 - [Дизайн](https://design.hse.ru/ba/program/design)

8 - [Мода](https://design.hse.ru/ba/program/fashion)

9 - [Информатика и вычислительная техника](https://www.hse.ru/ba/isct/)

10 - [Программная инженерия](https://www.hse.ru/ba/se/)

11 - [Информационная безопасность](https://www.hse.ru/ba/is/)

12 - [Компьютерная безопасность](https://www.hse.ru/ba/cosec/)

13 - [История искусств](https://www.hse.ru/ba/histart/)

14 - [Современное искусство](https://art.hse.ru/ba)

15 - [Античность](https://www.hse.ru/ba/antiq/)

16 - [История](https://www.hse.ru/ba/hist/)

17 - [Культурология](https://www.hse.ru/ba/cultural/)

18 - [Компьютерные науки и анализ данных](https://www.hse.ru/ba/compds/)

19 - [Математика](https://www.hse.ru/ba/math/)

20 - [Прикладная математика](https://www.hse.ru/ba/am/)

21 - [Прикладная математика и информатика](https://www.hse.ru/ba/ami/)

22 - [Программа двух дипломов НИУ ВШЭ и Лондонского университета «Прикладной анализ данных»](https://www.hse.ru/ba/data/)

23 - [Совместный бакалавриат НИУ ВШЭ и ЦПМ](https://www.hse.ru/ba/cpm/)

24 - [География глобальных изменений и геоинформационные технологии](https://www.hse.ru/ba/geography/)

25 - [Международные отношения](https://www.hse.ru/ba/ia/)

26 - [Политология](https://www.hse.ru/ba/political/)

27 - [Программа двух дипломов НИУ ВШЭ и Лондонского университета по международным отношениям](https://www.hse.ru/ba/ir/)

28 - [Программа двух дипломов НИУ ВШЭ и Университета Кёнхи «Экономика и политика в Азии»](https://www.hse.ru/ba/epa/)

29 - [Психология](https://www.hse.ru/ba/psy/)

30 - [Социология](https://www.hse.ru/ba/soc/)

31 – [Журналистика](https://www.hse.ru/ba/journ/)

32 – [Медиакоммуникации](https://www.hse.ru/ba/media/)

33 – [Реклама и связи с общественностью](https://www.hse.ru/ba/ad/)

34 – [Физика](https://www.hse.ru/ba/physics/)

35 – [Философия](https://www.hse.ru/ba/phil/)

36 – [Химия](https://www.hse.ru/ba/chem/)

37 – [Бизнес-информатика](https://www.hse.ru/ba/bi/)

38 – [Государственное и муниципальное управление](https://www.hse.ru/ba/gmu/)

39 – [Логистика и управление цепями поставок](https://www.hse.ru/ba/logistics/)

40 – [Маркетинг и рыночная аналитика](https://www.hse.ru/ba/marketing/)

41 – [Мировая экономика](https://www.hse.ru/ba/we/)

42 – [Программа двух дипломов по экономике НИУ ВШЭ и Лондонского университета](https://www.hse.ru/ba/icef/)

43 – [Совместная программа по экономике НИУ ВШЭ и РЭШ](https://www.hse.ru/ba/nes/)

44 – [Управление бизнесом](https://www.hse.ru/ba/bba/)

45 – [Цифровые инновации в управлении предприятием (программа двух дипломов НИУ ВШЭ и Лондонского университета)](https://www.hse.ru/ba/digital/)

46 – [Экономика](https://www.hse.ru/ba/economics/)

47 – [Экономика и статистика](https://www.hse.ru/ba/stat/)

48 – [Инфокоммуникационные технологии и системы связи](https://www.hse.ru/ba/it/)

49 – [Юриспруденция](https://www.hse.ru/ba/law/)

50 – [Юриспруденция: частное право](https://www.hse.ru/ba/privlaw/)

51 – [Иностранные языки и межкультурная коммуникация](https://www.hse.ru/ba/lang/)

52 – [Филология](https://www.hse.ru/ba/philology/)

53 – [Фундаментальная и компьютерная лингвистика](https://www.hse.ru/ba/ling/)

Расположение корпуса:

1 – Комплекс зданий «Мясницкая»

2 – Комплекс зданий «Покровка»

3 – Комплекс зданий «Шаболовка»

4 – Комплекс зданий «Басманная»

5 – Остальные здания ВШЭ

Общежитие:

1 – Общежитие №1

2 – Общежитие №2

3 – Общежитие №3

4 – Общежитие №4

5 – Общежитие №5

6 – Общежитие №6

7 – Общежитие №7

8 – Общежитие №8 “Трилистник”

9 – Общежитие №9

10 – Общежитие №10

11 – Общежитие №11

12 – Студенческий городок Дубки

Пол:

1 – мужской

2 – женский

Категории:

0 – Учеба

1 – Не учеба

Подкатегории:

0 – Другое

1 – Транспорт

2 – Развлечения

3 – Съемка/монтаж

4 – Кулинария

5 – Компьютеры

6 – Красота

7 – Товары

8 – Быт

9 – Другое

10 – Языки

11 – Программирование

12 – Экономика

13 – Математика

14 – Работа с текстом

15 – Литература

16 – Обществознание

17 – Искусство

18 – Психология

19 – История

20 – Естествознание

21 – Презентации

# ПРИЛОЖЕНИЕ 4

### ОГРАНИЧЕНИЯ, НАКЛАДЫВАЕМЫЕ НА ДАННЫЕ,

### ХРАНИМЫЕ В БАЗЕ ДАННЫХ

Таблица 4.1

Ограничения полей класса Skill.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** |  |  |  |
| Имя | Тип | Ограничение | Сообщение при нарушении |
| Id | long | - | - |
| Status | int | Принимает значения от 0 до 2 | Некорректный статус навыка |
| Name | string | Длина не больше 50 символов | Слишком много символов |
| Description | string | Длина не больше 300 символов | Слишком много символов |
| Category | int | Принимает значения от 0 до 1 | Некорректная категория |
| Subcategory | int | Принимает значения от 0 до 21 | Некорректная подкатегория |
| UserMail | string | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес |
| UserName | string | Длина от 2 до 40 символов | Имя должно быть от 2 до 40 символов |
| UserName | string | Состоит из латинских и кириллических заглавных и строчных букв | В имени могут присутствовать только буквы |
| UserSurname | string | Длина от 2 до 40 символов | Фамилия должна быть от 2 до 40 символов |
| UserSurname | string | Состоит из латинских и кириллических заглавных и строчных букв | В фамилии могут присутствовать только буквы |
| UserPhoto | byte[] | - | - |

Таблица 4.2

Ограничения полей класса Transaction.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | |  | | |
| Имя | Тип | | Ограничение | Сообщение при нарушении |
| Id | long | | - | - |
| Skill1 | string | | Длина не больше 50 символов | Слишком много символов |
| Skill2 | string | | Длина не больше 50 символов | Слишком много символов |
| Description | string | | Длина не больше 300 символов | Слишком много символов |
| SenderMail | string | | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес отправителя |
| ReceiverMail | string | | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес получателя |
| WhoWantMail | string | | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес |
| Status | int | | Принимает значения от 0 до 2 | Некорректный статус обмена |
| Users | List<User> | | - | - |

Таблица 4.3

Ограничения полей класса User.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| Имя | Тип | Ограничение | Сообщение при нарушении |
| Mail | string | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес |
| Confirmation  CodeServer | int | - | - |
| Confirmation  CodeUser | int | Равенство с ConfirmationCodeServer | Неверный код подтверждения |
| Password | string | Длина от 6 до 40 символов | Пароль должен быть от 6 до 40 символов |
| Name | string | Длина от 2 до 40 символов | Имя должно быть от 2 до 40 символов |
| Name | string | Состоит из латинских и кириллических заглавных и строчных букв | В имени могут присутствовать только буквы |
| Surname | string | Длина от 2 до 40 символов | Фамилия должна быть от 2 до 40 символов |
| Surname | string | Состоит из латинских и кириллических заглавных и строчных букв | В фамилии могут присутствовать только буквы |
| BirthDate | Date  Time | Возраст должен быть от 13 до 100 лет | Некорректная дата рождения |
| Gender | int | Принимает значения от 0 до 2 | Некорректный пол |
| StudyingYearId | int | Принимает значения от 0 до 7 | Некорректный ID курса |
| MajorId | int | Принимает значения от 0 до 53 | Некорректный ID образовательной программы |
| Campus  LocationId | int | Принимает значения от 0 до 5 | Некорректный ID расположения корпуса |
| DormitoryId | int | Принимает значения от 0 до 12 | Некорректный ID общежития |
| About | string | Длина не больше 200 символов | Слишком много символов |
| Contact | string | Состоит из комбинации "vk.com/" или "t.me/" в начале, а также латинских заглавных и строчных букв, цифр, подчеркиваний, тире и точек | Некорректный формат ссылки |
| Photo | byte[] | - | - |
| Transactions | List  <Transaction> | - | - |
| Skills | List  <Skill> | - | - |
| Feedbacks | List<Feedback> | - | - |
| Chats | List  <Chat> | - | - |
| GradesCount | int | - | - |
| GradesSum | int | - | - |
| AverageGrade | double | - | - |
| IsModer | bool | - | - |

Таблица 4.4

Ограничения полей класса Message.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** |  |  |  |
| Имя | Тип | Ограничение | Сообщение при нарушении |
| Id | long | - | - |
| SendTime | DateTime | - | - |
| Text | string | Длина не больше 300 символов | Слишком много символов |
| SenderMail | string | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес отправителя |
| ReceiverMail | string | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес получателя |

Таблица 4.5

Ограничения полей класса Feedback.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | |  | | |
| Имя | Тип | | Ограничение | Сообщение при нарушении |
| Id | long | | - | - |
| Grade | int | | Принимает значения от 0 до 5 | Некорректная оценка |
| Comment | string | | Длина не больше 300 символов | Слишком много символов |
| SenderMail | string | | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес отправителя |
| ReceiverMail | string | | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес получателя |

Таблица 4.6

Ограничения полей класса Chat.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| Имя | Тип | Ограничение | Сообщение при нарушении |
| Id | long | - | - |
| LastMessage | string | Длина не больше 300 символов | Слишком много символов |
| SendTime | DateTime | - | - |
| Mail1 | string | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес собеседника |
| Name1 | string | Длина от 2 до 40 символов | Имя должно быть от 2 до 40 символов |
| Name1 | string | Состоит из латинских и кириллических заглавных и строчных букв | В имени могут присутствовать только буквы |
| Surname1 | string | Длина от 2 до 40 символов | Фамилия должна быть от 2 до 40 символов |
| Surname1 | string | Состоит из латинских и кириллических заглавных и строчных букв | В фамилии могут присутствовать только буквы |
| Photo1 | byte[] | - | - |
| Mail2 | string | Состоит из латинских заглавных и строчных букв, цифр, подчеркиваний, и комбинации "@edu.hse.ru" на конце | Некорректный почтовый адрес собеседника |
| Name2 | string | Длина от 2 до 40 символов | Имя должно быть от 2 до 40 символов |
| Name2 | string | Состоит из латинских и кириллических заглавных и строчных букв | В имени могут присутствовать только буквы |
| Surname2 | string | Длина от 2 до 40 символов | Фамилия должна быть от 2 до 40 символов |
| Surname2 | string | Состоит из латинских и кириллических заглавных и строчных букв | В фамилии могут присутствовать только буквы |
| Photo2 | byte[] | - | - |
| Users | List<User> | - | - |

# ПРИЛОЖЕНИЕ 5

## ОПИСАНИЕ И НАЗНАЧЕНИЕ МЕТОДОВ В СООТВЕТСТВИИ С REST API

Таблица 5.1

Описание методов класса MessagesController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Пример URL-адреса | | Назначение | | |
| GetMessage | GET | | api/Messages/1 | | Возвращает конкретное сообщение. | | |
| GetMessages | GET | | api/Messages | | Возвращает все сообщения. | | |
| GetMessages | GET | | api/Messages/kdtsaryov@edu.hse.ru/  eoshtanko@edu.hse.ru/0 | | Возвращает 100 последних сообщений между двумя конкретными пользователями. | | |
| PutMessage | PUT | | api/Messages/1 | | Изменение конкретного сообщения. | | |
| PostMessage | POST | | api/Messages | | Добавление нового сообщения. | | |
| DeleteMessage | DELETE | | api/Messages/5 | | Удаление сообщения. | | |

Таблица 5.2

Описание методов класса FeedbacksController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Пример URL-адреса | | Назначение | | |
| GetFeedbacks | GET | | api/Feedbacks | | Возвращает все отзывы. | | |
| GetFeedback | GET | | api/Feedbacks/1 | | Возвращает конкретный отзыв. | | |
| PostFeedback | POST | | api/Feedbacks | | Добавление нового отзыва. | | |
| PutFeedback | PUT | | api/Feedbacks/1 | | Изменение отзыва. | | |
| DeleteFeedbackUser | DELETE | | api/Feedbacks/user/  eoshtanko@edu.hse.ru/1 | | Удаление отзыва пользователем. | | |
| DeleteFeedbackModer | DELETE | | api/Feedbacks/moder/  eoshtanko@edu.hse.ru/1 | | Удаление отзыва модератором. | | |

Таблица 5.3

Описание новых методов класса UsersController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Пример URL-адреса | | Назначение | | |
| GetUserFeedbacks | GET | | api/Users/eoshtanko@edu.hse.ru  /feedbacks | | Возвращает отзывы конкретного пользователя. | | |

Таблица 5.4

Описание методов класса ChatsController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Пример URL-адреса | | Назначение | | |
| GetChat | GET | | api/Chats/1 | | Возвращает конкретную переписку. | | |
| GetChats | GET | | api/Chats | | Возвращает все переписки. | | |
| GetChats | GET | | api/Chats/kdtsaryov@edu.hse.ru/  user | | Возвращает все переписки конкретного пользователя. | | |
| PutChat | PUT | | api/Chats/1 | | Изменение конкретной переписки. | | |
| PostChat | POST | | api/Chats | | Добавление новой переписки. | | |
| DeleteChat | DELETE | | api/Chats/5 | | Удаление переписки. | | |

# ПРИЛОЖЕНИЕ 6

## ОПИСАНИЕ СЕРВЕРНЫХ ОШИБОК, КОТОРЫЕ ВОЗВРАЩАЮТ HTTP МЕТОДЫ

Таблица 6.1

Описание ошибок в методах класса MessagesController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Ошибки | | Описание | | |
| GetMessage | GET | | 404 NOT FOUND | | Сообщения с таким ID нет. | | |
| GetMessages | GET | | - | | - | | |
| GetMessages | GET | | 400 BAD REQUEST | | Нет такого первого пользователя или второго пользователя, или первый равен второму. | | |
| PutMessage | PUT | | 400 BAD REQUEST | | Переданный ID не равен ID сообщения. | | |
| PutMessage | PUT | | 404 NOT FOUND | | Сообщения с таким ID нет. | | |
| PostMessage | POST | | 400 BAD REQUEST | | Такое сообщение уже есть. | | |
| PostMessage | POST | | 400 BAD REQUEST | | Нет такого отправителя или получателя, или отправитель равен получателю. | | |
| PostMessage | POST | | 400 BAD REQUEST | | Нет такой переписки. | | |
| DeleteMessage | DELETE | | 404 NOT FOUND | | Сообщения с таким ID нет. | | |

Таблица 6.2

Описание ошибок в методах класса FeedbacksController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Ошибки | | Описание | | |
| GetFeedbacks | GET | | - | | - | | |
| GetFeedback | GET | | 404 NOT FOUND | | Отзыва с таким ID нет. | | |
| PostFeedback | POST | | 400 BAD REQUEST | | Такой отзыв уже есть. | | |
| PostFeedback | POST | | 400 BAD REQUEST | | Нет такого отправителя или получателя, или отправитель равен получателю. | | |
| PutFeedback | PUT | | 400 BAD REQUEST | | Переданный ID не равен ID отзыва. | | |
| PutFeedback | PUT | | 404 NOT FOUND | | Отзыва с таким ID нет. | | |
| DeleteFeedbackUser | DELETE | | 404 NOT FOUND | | Отзыва с таким ID нет. | | |
| DeleteFeedbackUser | DELETE | | 400 BAD REQUEST | | Попытка удаления чужого отзыва. | | |
| DeleteFeedbackModer | DELETE | | 404 NOT FOUND | | Отзыва с таким ID нет. | | |
| DeleteFeedbackModer | DELETE | | 400 BAD REQUEST | | Попытка удаления отзыва о себе. | | |

Таблица 6.3

Описание ошибок в новых методах класса UsersController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Ошибки | | Описание | | |
| GetUserFeedbacks | GET | | 404 NOT FOUND | | Пользователя с такой почтой нет. | | |
| GetUserFeedbacks | GET | | 404 NOT FOUND | | У пользователя с такой почтой нет отзывов. | | |

Таблица 6.4

Описание ошибок в методах класса ChatsController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Ошибки | | Описание | | |
| GetChat | GET | | 404 NOT FOUND | | Переписки с таким ID нет. | | |
| GetChats | GET | | - | | - | | |
| GetChats | GET | | 400 BAD REQUEST | | Нет такого пользователя. | | |
| PutChat | PUT | | 400 BAD REQUEST | | Переданный ID не равен ID переписки. | | |
| PutChat | PUT | | 404 NOT FOUND | | Переписки с таким ID нет. | | |
| PostChat | POST | | 400 BAD REQUEST | | Такая переписка уже есть. | | |
| PostChat | POST | | 400 BAD REQUEST | | Нет такого первого собеседника или второго собеседника, или собеседники равны. | | |
| DeleteChat | DELETE | | 404 NOT FOUND | | Переписки с таким ID нет. | | |

# ПРИЛОЖЕНИЕ 7

## АВТОМАТИЧЕСКИ ГЕНЕРИРУЕМЫЕ ПИСЬМА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ

1. Письмо при регистрации нового пользователя:

Тема –

Код подтверждения регистрации

Само письмо –

Здравствуйте!

Спасибо за регистрацию в Обмене Навыками

Ваш код подтверждения - 5678

1. Письмо для напоминания забытого пароля

Тема –

Напоминание пароля

Само письмо –

Здравствуйте!

В Обмене Навыками была нажата кнопка "Забыли пароль?"

Ваш пароль – simSIM2237

1. Оповещение о новом обмене

Тема –

Новый обмен

Само письмо –

Екатерина Штанько предложил(а) Вам новый обмен.

Зайдите в приложение "Обмен навыками", чтобы узнать детали.

1. Оповещение об отказе в обмене

Тема –

Отказ в обмене

Само письмо –

Екатерина Штанько отказал(а) Вам в обмене.

Зайдите в приложение "Обмен навыками", чтобы узнать детали.

1. Оповещение о принятии обмена

Тема –

Принятый обмен

Само письмо –

Екатерина Штанько принял(а) Ваш обмен.

Зайдите в приложение "Обмен навыками", чтобы узнать детали.

1. Оповещение о завершении обмена

Тема –

Обмен завершен

Само Письмо –

Екатерина Штанько завершил(а) обмен.

Зайдите в приложение "Обмен навыками", чтобы узнать детали.

1. Оповещение о новом сообщении в чате

Тема –

Новое сообщение

Само Письмо –

Вам отправили новое сообщение.

Зайдите в приложение "Обмен навыками", чтобы узнать детали.

1. Оповещение о присвоении статуса модератора

Тема –

Статус модератора

Само Письмо –

Поздравляем!

Вам стали доступны функции модератора в приложении "Обмен навыками".

Теперь Вы можете удалять неприличные отзывы пользователей приложения.

1. Оповещение о новом отзыве

Тема –

Новый отзыв

Само письмо –

Екатерина Штанько оставил(а) Вам новый отзыв.

Зайдите в приложение "Обмен навыками", чтобы узнать детали.

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированх |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |