**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  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  профессор департамента  программной инженерии,  канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 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.**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.06.02-01 81 01-1-ЛУ** | | |
|  |  | |
| Исполнитель:  студент группы БПИ193  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Царёв К.Д. /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. | |
|  | | |
|  | |  |

**Москва 2021**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.06.02-01 81 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Пояснительная записка**  **RU.17701729.06.02-01 81 01-1**  **Листов 28** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2021**

**АННОТАЦИЯ**

В данном программном документе приведена пояснительная записка к серверной части приложения для обмена навыками и услугами для студентов НИУ ВШЭ.

В разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

− постановка задачи на разработку программы;

− описание алгоритма и функционирования программы;

− описание метода организации входных и выходных данных;

− описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

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 19.404-79 Пояснительная записка. Требования к содержанию и оформлению.

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78, ГОСТ 19.604-78.

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 настоящей пояснительной записки.

ОГЛАВЛЕНИЕ

[1. ВВЕДЕНИЕ 5](#_Toc71974179)

[1.1. Наименование программы 5](#_Toc71974180)

[1.2. Документы, на основании которых ведется разработка 5](#_Toc71974181)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 6](#_Toc71974182)

[2.1. Назначение программы 6](#_Toc71974183)

[2.1.1. Функциональное назначение 6](#_Toc71974184)

[2.1.2. Эксплуатационное назначение 6](#_Toc71974185)

[2.2. Краткая характеристика области применения 6](#_Toc71974186)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 7](#_Toc71974187)

[3.1. Постановка задачи на разработку программы 7](#_Toc71974188)

[3.2. Описание алгоритма и функционирования программы 7](#_Toc71974189)

[3.2.1. Описание доступных пользователю функций программы и их связь между собой 7](#_Toc71974190)

[3.2.2. Описание алгоритма регистрации нового пользователя 8](#_Toc71974191)

[3.2.3. Описание процесса авторизации пользователя 8](#_Toc71974192)

[3.2.4. Описание процесса добавления нового навыка 8](#_Toc71974193)

[3.2.5. Описание алгоритма поиска навыков 8](#_Toc71974194)

[3.2.6. Описание алгоритма создания обмена 8](#_Toc71974195)

[3.2.7. Описание процесса завершения обмена 8](#_Toc71974196)

[3.2.8. Описание процесса напоминания забытого пароля 8](#_Toc71974197)

[3.3. Описание и обоснование выбора метода организации входных и выходных данных 9](#_Toc71974198)

[3.3.1. Описание метода организации входных и выходных данных 9](#_Toc71974199)

[3.3.2. Обоснования выбора метода организации входных и выходных данных 9](#_Toc71974200)

[3.4. Описание и обоснование выбора состава технических и программных средств 9](#_Toc71974201)

[3.4.1. Состав технических и программных средств 9](#_Toc71974202)

[3.4.2. Обоснование выбора технических и программных средств 9](#_Toc71974203)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 10](#_Toc71974204)

[4.1. Предполагаемая потребность 10](#_Toc71974205)

[4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 10](#_Toc71974206)

[ПРИЛОЖЕНИЕ 1 11](#_Toc71974207)

[ТЕРМИНОЛОГИЯ 11](#_Toc71974208)

[ПРИЛОЖЕНИЕ 2 12](#_Toc71974209)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 12](#_Toc71974210)

[ПРИЛОЖЕНИЕ 3 13](#_Toc71974211)

[ИДЕНТИФИКАТОРЫ ЛИЧНЫХ ДАННЫХ И КАТЕГОРИЙ НАВЫКОВ 13](#_Toc71974212)

[ПРИЛОЖЕНИЕ 4 16](#_Toc71974213)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ 16](#_Toc71974214)

[ПРИЛОЖЕНИЕ 5 17](#_Toc71974215)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ 17](#_Toc71974216)

[ПРИЛОЖЕНИЕ 6 23](#_Toc71974217)

[ОПИСАНИЕ И НАЗНАЧЕНИЕ МЕТОДОВ В СООТВЕТСТВИИ С REST API 23](#_Toc71974218)

[ПРИЛОЖЕНИЕ 7 25](#_Toc71974219)

[ОПИСАНИЕ СЕРВЕРНЫХ ОШИБОК, КОТОРЫЕ ВОЗВРАЩАЮТ HTTP МЕТОДЫ 25](#_Toc71974220)

[ПРИЛОЖЕНИЕ 8 27](#_Toc71974221)

[АВТОМАТИЧЕСКИ ГЕНЕРИРУЕМЫЕ ПИСЬМА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ 27](#_Toc71974222)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 28](#_Toc71974223)

# ВВЕДЕНИЕ

## Наименование программы

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

Наименование программы на английском языке – Server part of application for sharing skills and services for HSE students.

Краткое наименование программы - «Sharing Skills HSE».

## Документы, на основании которых ведется разработка

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 "Программная инженерия" и утвержденная академическим руководителем тема курсового проекта.

# НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

## Назначение программы

### Функциональное назначение

Программный продукт представляет собой приложение, в котором пользователь может: заполнять свой профиль, искать других пользователей, предлагать обмены, соглашаться или отказываться от обменов, завершать обмены, смотреть свои истории обменов.

Серверная часть приложения предназначена для:

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

### Эксплуатационное назначение

Данное приложение может быть использовано студентами НИУ ВШЭ, желающими совершить обмен навыками/услугами с человеком из сообщества НИУ ВШЭ.

## Краткая характеристика области применения

«Sharing Skills HSE» – программа, позволяющая находить людей, готовых делиться своими знаниями и умениями друг с другом. Область применения программы – сфера знакомств и услуг.

# ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

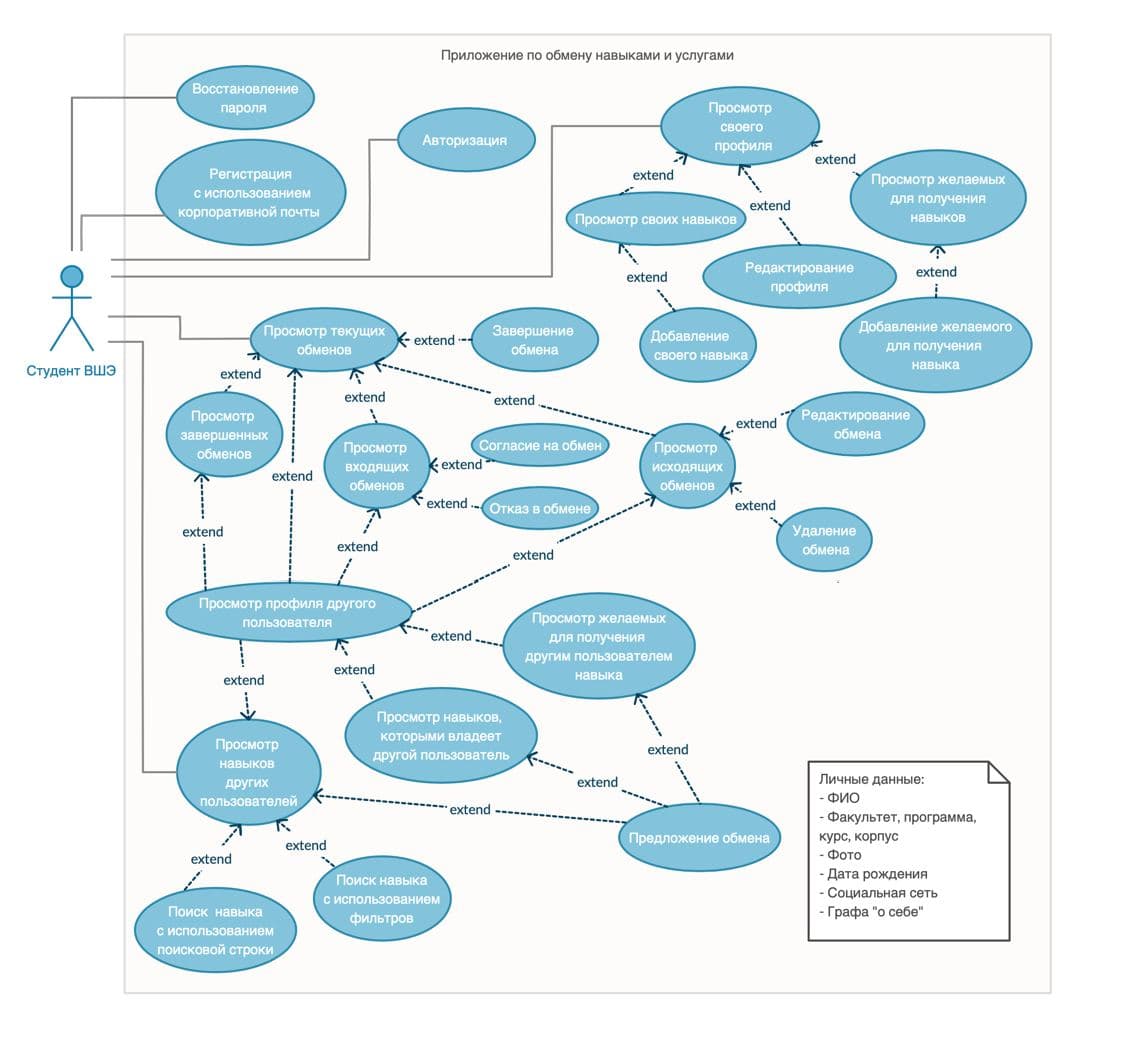
## Постановка задачи на разработку программы

Разрабатываемая программа должна соответствовать функциональным характеристикам, требованиям к интерфейсу и надежности, описанным в техническом задании.

## Описание алгоритма и функционирования программы

### Описание доступных пользователю функций программы и их связь между собой

Функции, доступные конечному пользователю программы, и их связь между собой приведены на рис. 1:



*Рисунок 1*

### Описание алгоритма регистрации нового пользователя

Пользователь вводит адрес своей корпоративной почты в домене @edu.hse.ru, с клиента отправляется POST-запрос, 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 пользователь постепенно вводит код подтверждения, который сравнивается со сгенерированным, и свои личные данные, которые 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.

### Описание процесса авторизации пользователя

В связи с тем, что в приложении не используются никакие строго конфиденциальные данные, процесс авторизации был реализован достаточно примитивно. GET-запросом клиент получает пароль, хранящийся на сервере, и сравнивает его с введенным в приложении.

### Описание процесса добавления 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.

### Описание алгоритма поиска 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, а уже непосредственно на клиенте осуществляется поиск по названиям и описаниям в этом списке.

### Описание алгоритма создания 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.

### Описание процесса завершения обмена

При завершении 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.

### Описание процесса напоминания забытого пароля

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

## Описание и обоснование выбора метода организации входных и выходных данных

### Описание метода организации входных и выходных данных

Входными данными для серверной части приложения являются HTTP-запросы от клиентской части приложения.

Выходными данными для приложения являются HTTP-ответы на запросы клиентской части приложения, а также автоматически генерируемые электронные письма, отправляющиеся пользователям.

### Обоснования выбора метода организации входных и выходных данных

Организация входных и выходных данных в виде HTTP запросов и ответов является наиболее удобным способом для взаимодействия клиентов и сервера, а отправка автоматически генерируемых писем на электронную почту является общепринятым средством взаимодействия сервера с конечным пользователем.

## Описание и обоснование выбора состава технических и программных средств

### Состав технических и программных средств

Для работы серверной части приложения необходим следующий состав технических средств:

* 1. 32-битная операционная система Windows
  2. Минимальный размер хранилища базы данных 32 гб
  3. Минимальный размер оперативной памяти 1 гб
  4. Двухъядерный процессор 2 ггц

### Обоснование выбора технических и программных средств

Данный состав технических и программных средств позволит программе осуществлять быстрое взаимодействие клиентов с сервером, быстро посылать автоматически генерируемые письма пользователям, и работать без перебоев. Столь незначительный размер хранилища базы данных обусловлен тем, что приложение будет доступно только для студентов НИУ ВШЭ.

# ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## Предполагаемая потребность

Предполагаемая потребность обуславливается тем фактом, что на данный момент не существует инструмента, позволяющего облегчить студентам НИУ ВШЭ процесс поиска людей для обмена навыками и услугами.

## Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

На момент написания данного документа не существует прямых аналогов разрабатываемого инструмента.

Есть сайты позволяющие обмениваться навыками и услугами:

• Авито

• Profi.ru

• Бартер.рф

Но в них отсутствует главная особенность и преимущество разрабатываемого приложения, а именно, безопасность, обеспечивающаяся за счет ограничения аудитории пользователей сообществом НИУ ВШЭ (регистрация исключительно через корпоративную почту). Данная разработка - уникальное приложение, не только позволяющее людям обмениваться услугами, но и являющаяся связующим звеном для студентов НИУ ВШЭ.

# 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.

**HTTP** – протокол прикладного уровня передачи произвольных данных.

**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.

# 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. [2020—2021]. Дата обновления: 21.10.2017. URL: https://ru.wikipedia.org/wiki/Клиент\_—\_сервер (дата обращения: 20.11.2020).

11. Deploy an ASP.NET Core App with EF Core and SQL Server to Azure [Электронный ресурс] // medium. [2020-2021]. Дата обновления: 01.01.2020. URL: https://medium.com/net-core/deploy-an-asp-net-core-app-with-ef-core-and-sql-server-to-azure-e11df41a4804 (дата обращения: 15.02.2021).

12. Руководство по ASP.NET Core 5 [Электронный ресурс] // metanit. [2020—2021]. Дата обновления: 15.12.2020. URL: https://metanit.com/sharp/aspnet5 (дата обращения: 29.12.2020).

# 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

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Таблица 4.1

Описание и функциональное назначение классов

|  |  |
| --- | --- |
| Класс | Назначение |
| SkillsController | Осуществляет взаимодействие клиента с сервером в отношении сущности навыков. |
| TransactionsController | Осуществляет взаимодействие клиента с сервером в отношении сущности обменов. |
| UsersController | Осуществляет взаимодействие клиента с сервером в отношении сущности пользователей. |
| SharingSkillsContext | Описывает контекст базы данных. |
| Skill | Описывает сущность навыка. |
| Transaction | Описывает сущность обмена. |
| User | Описывает сущность пользователя. |
| DateRangeAttribute | Атрибут валидации даты рождения. |
| Mail | Отправляет автоматически генерируемые электронные письма пользователям. |
| Startup | Производит первоначальную конфигурацию сервера перед публикацией. |

# ПРИЛОЖЕНИЕ 5

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ

Таблица 5.1

Описание полей, методов и свойств класса SkillsController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| \_context | | private | | SharingSkillsContext | | Контекст базы данных. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| SkillsController | public | | void | | SharingSkillsContext context | | Конструктор. |
| GetSkills | public | | Task<ActionResult  <IEnumerable<Skill>>> | | - | | Возвращает все навыки. |
| GetSkill | public | | Task<ActionResult  <Skill>> | | long id | | Возвращает конкретный навык. |
| GetUserSkills | public | | Task<ActionResult  <IEnumerable<Skill>>> | | string mail | | Возвращает все навыки конкретного пользователя. |
| PutSkill | public | | Task<IActionResult> | | long id, Skill skill | | Изменение навыка. |
| PostSkill | public | | Task<ActionResult  <Skill>> | | Skill skill | | Добавление навыка. |
| DeleteSkill | public | | Task<IActionResult> | | long id | | Удаление навыка. |
| SkillExists | private | | bool | | long id | | Проверка наличия навыка. |

Таблица 5.2

Описание полей, методов и свойств класса TransactionsController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| \_context | | private | | SharingSkillsContext | | Контекст базы данных. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| TransactionsController | public | | void | | SharingSkillsContext context | | Конструктор. |
| GetTransaction | public | | Task<ActionResult  <Transaction>> | | int id | | Возвращает конкретный обмен. |
| GetTransactions | public | | Task<ActionResult  <IEnumerable  <Transaction>>> | | - | | Возвращает все обмены. |
| PutTransaction | public | | Task  <IActionResult> | | int id,  Transaction transaction,  string mail = "" | | Изменение обмена. |
| PostTransaction | public | | Task<ActionResult  <Transaction>> | | Transaction transaction | | Добавление нового обмена. |
| DeleteTransaction | public | | Task  <IActionResult> | | long id,  bool sendNotification = false | | Удаление обмена. |
| TransactionExists | private | | bool | | long id | | Проверка наличия обмена. |

Таблица 5.3

Описание полей, методов и свойств класса UsersController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| \_context | | private | | SharingSkillsContext | | Контекст базы данных. | |
| rnd | | private | | Random | | Генератор случайных чисел. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| UsersController | public | | void | | SharingSkillsContext context | | Конструктор. |
| ForgetPassword | public | | Task<ActionResult  <User>> | | string mail | | Напоминание забытого пароля по почте. |
| GetUsers | public | | Task<ActionResult  <IEnumerable  <User>>> | | - | | Возвращает всех пользователей. |
| GetUser | public | | Task<ActionResult  <User>> | | string mail | | Возвращает пользователя с конкретной почтой. |
| GetUsersSkills | public | | Task<ActionResult  <IEnumerable  <Skill>>> | | string mail,  int studyingYearID  = -1,  int majorID = -1,  int campusLocationID  = -1,  int dormitoryID  = -1,  int gender = -1,  int skillstatus = -1, int category = -1,  int subcategory = -1 | | Возвращает навыки, удовлетворяющие каким-то критериям, без навыков конкретного пользователя |
| GetUserActive  Transactions | public | | Task<ActionResult  <Transaction>> | | string mail | | Возвращает текущие обмены конкретного пользователя. |
| GetUser  Completed  Transactions | public | | Task<ActionResult  <Transaction>> | | string mail | | Возвращает завершенные обмены конкретного пользователя. |
| GetUserIn  Transactions | public | | Task<ActionResult  <Transaction>> | | string mail | | Возвращает входящие обмены конкретного пользователя. |
| GetUserOut  Transactions | public | | Task<ActionResult  <Transaction>> | | string mail | | Возвращает исходящие обмены конкретного пользователя. |
| PutUser | public | | Task<IActionResult> | | string mail,  User user | | Обновление данных конкретного пользователя. |
| PostUser | public | | Task<ActionResult  <User>> | | User user | | Добавление нового пользователя. |
| DeleteUser | public | | Task<IActionResult> | | string mail | | Удаление пользователя. |
| UserExists | private | | bool | | string mail | | Проверка наличия пользователя. |

Таблица 5.4

Описание полей, методов и свойств класса SharingSkillsContext.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| SharingSkillsContext | public | | void | | DbContextOptions  <SharingSkillsContext> options | | Конструктор. |
| OnModelCreating | protected | | void | | ModelBuilder builder | | Первоначальное создание моделей |
| **Свойства** |  | |  | |  | |  |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение |
| Skills | public | | DbSet<Skill> | | get, set | | Свойство доступа к навыкам. |
| Transactions | public | | DbSet  <Transaction> | | get, set | | Свойство доступа к обменам. |
| Users | public | | DbSet<User> | | get, set | | Свойство доступа к пользователям. |

Таблица 5.5

Описание полей, методов и свойств класса Skill.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Свойства** |  |  |  |  |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Id | public | long | get, set | ID навыка. |
| Status | public | int | get, set | Статус (1-могу, 2-хочу). |
| Name | public | string | get, set | Название. |
| Description | public | string | get, set | Описание. |
| Category | public | int | get, set | Категория |
| Subcategory | public | int | get, set | Подкатегория. |
| UserMail | public | string | get, set | Почта пользователя. |
| User | public | User | get, set | Пользователь. |

Таблица 5.6

Описание полей, методов и свойств класса Transaction.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Свойства** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение |
| Id | public | | long | | get, set | | ID обмена. |
| Skill1 | public | | string | | get, set | | Первый навык. |
| Skill2 | public | | string | | get, set | | Второй навык. |
| Description | public | | string | | get, set | | Описание. |
| SenderMail | public | | string | | get, set | | Почта отправителя. |
| ReceiverMail | public | | string | | get, set | | Почта получателя. |
| WhoWantMail | public | | string | | get, set | | Почта того, кто "хочет". |
| Status | public | | int | | get, set | | Статус  (0-отправлен,  1-подтвержден (активен),  2-завершен). |
| Users | public | | List<User> | | get, set | | Отправитель и получатель. |

Таблица 5.7

Описание полей, методов и свойств класса User.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Свойства** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение |
| Mail | public | | string | | get, set | | Почта. |
| ConfirmationCodeServer | public | | int | | get, set | | Код подтверждения почты, хранящийся на сервере. |
| ConfirmationCodeUser | public | | int | | get, set | | Код подтверждения почты, запрашиваемый у пользователя. |
| Password | public | | string | | get, set | | Пароль. |
| Name | public | | string | | get, set | | Имя. |
| Surname | public | | string | | get, set | | Фамилия. |
| BirthDate | public | | DateTime | | get, set | | Дата рождения. |
| Gender | public | | int | | get, set | | Пол. |
| StudyingYearId | public | | int | | get, set | | Курс. |
| MajorId | public | | int | | get, set | | Образовательная программа. |
| CampusLocationId | public | | int | | get, set | | Расположение корпуса. |
| DormitoryId | public | | int | | get, set | | Общежитие. |
| About | public | | string | | get, set | | О себе. |
| Contact | public | | string | | get, set | | Ссылка на мессенджер (telegram или vk). |
| Photo | public | | byte[] | | get, set | | Фотография. |
| Transactions | public | | List<Transaction> | | get, set | | Обмены. |
| Skills | public | | List<Skill> | | get, set | | Навыки. |

Таблица 5.8

Описание полей, методов и свойств класса DateRangeAttribute.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| maxYear | | private | | int | | Максимальный год рождения. | |
| minYear | | private | | int | | Минимальный год рождения. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| DateRangeAttribute | public | | void | | int min, int max | | Конструктор. |
| IsValid | public | | bool | | object value | | Валидация даты рождения. |

Таблица 5.9

Описание полей, методов и свойств класса Mail.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| SendEmailAsync | public | | Task | | string email, string subject, string message | | Отправка письма на почту. |

Таблица 5.10

Описание полей, методов и свойств класса Startup.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Startup | public | | void | | IConfiguration configuration | | Конструктор. |
| ConfigureServices | public | | void | | IServiceCollection services | | Подключение сервисов. |
| Configure | public | | void | | IApplicationBuilder app, IWebHostEnvironment env | | Первоначальная настройка конфигурации перед публикацией. |
| **Свойства** |  | |  | |  | |  |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение |
| Configuration | public | | IConfiguration | | get | | Свойства доступа к конфигурации. |

# ПРИЛОЖЕНИЕ 6

## ОПИСАНИЕ И НАЗНАЧЕНИЕ МЕТОДОВ В СООТВЕТСТВИИ С REST API

Таблица 6.1

Описание методов класса SkillsController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Пример URL-адреса | | Назначение | | |
| GetSkills | GET | | api/Skills | | Возвращает все навыки. | | |
| GetSkill | GET | | api/Skills/1 | | Возвращает конкретный навык. | | |
| GetUserSkills | GET | | api/Skills/  kdtsaryov@edu.hse.ru/skills | | Возвращает все навыки конкретного пользователя. | | |
| PutSkill | PUT | | api/Skills/1 | | Изменение навыка. | | |
| PostSkill | POST | | api/Skills | | Добавление навыка. | | |
| DeleteSkill | DELETE | | api/Skills/5 | | Удаление навыка. | | |

Таблица 6.2

Описание методов класса TransactionsController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Пример URL-адреса | | Назначение | | |
| GetTransaction | GET | | api/Transactions/1 | | Возвращает конкретный обмен. | | |
| GetTransactions | GET | | api/Transactions | | Возвращает все обмены. | | |
| PutTransaction | PUT | | api/Transactions/1?  mail=kdtsaryov@edu.hse.ru | | Изменение обмена. | | |
| PostTransaction | POST | | api/Transactions | | Добавление нового обмена. | | |
| DeleteTransaction | DELETE | | api/Transactions/1?  sendNotification=true | | Удаление обмена. | | |

Таблица 6.3

Описание методов класса UsersController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Пример URL-адреса | | Назначение | | |
| ForgetPassword | GET | | api/Users/  kdtsaryov@edu.hse.ru/password | | Напоминание забытого пароля по почте. | | |
| GetUsers | GET | | api/Users | | Возвращает всех пользователей. | | |
| GetUser | GET | | api/Users/kdtsaryov@edu.hse.ru | | Возвращает пользователя с конкретной почтой. | | |
| GetUsersSkills | GET | | api/Users/kdtsaryov@edu.hse.ru/skills?  studyingYearID=1&majorID=1&  campusLocationID=1&dormitoryID=1&  gender=1&skillstatus=1&  category=1&subcategory=1 | | Возвращает навыки, удовлетворяющие каким-то критериям, без навыков конкретного пользователя | | |
| GetUserActive  Transactions | GET | | api/Users/kdtsaryov@edu.hse.ru/  transactions/active | | Возвращает текущие обмены конкретного пользователя. | | |
| GetUser  Completed  Transactions | GET | | api/Users/kdtsaryov@edu.hse.ru/  transactions/completed | | Возвращает завершенные обмены конкретного пользователя. | | |
| GetUserIn  Transactions | GET | | api/Users/kdtsaryov@edu.hse.ru/  transactions/in | | Возвращает входящие обмены конкретного пользователя. | | |
| GetUserOut  Transactions | GET | | api/Users/kdtsaryov@edu.hse.ru/  transactions/out | | Возвращает исходящие обмены конкретного пользователя. | | |
| PutUser | PUT | | api/Users/kdtsaryov@edu.hse.ru | | Обновление данных конкретного пользователя. | | |
| PostUser | POST | | api/Users | | Добавление нового пользователя. | | |
| DeleteUser | DELETE | | api/Users/kdtsaryov@edu.hse.ru | | Удаление пользователя. | | |

# ПРИЛОЖЕНИЕ 7

## ОПИСАНИЕ СЕРВЕРНЫХ ОШИБОК, КОТОРЫЕ ВОЗВРАЩАЮТ HTTP МЕТОДЫ

Таблица 7.1

Описание ошибок в методах класса SkillsController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Ошибки | | Описание | | |
| GetSkills | GET | | - | | - | | |
| GetSkill | GET | | 404 NOT FOUND | | Навыка с таким ID нет. | | |
| GetUserSkills | GET | | - | | - | | |
| PutSkill | PUT | | 400 BAD REQUEST | | Переданный ID не равен ID навыка. | | |
| PutSkill | PUT | | 404 NOT FOUND | | Навыка с таким ID нет. | | |
| PostSkill | POST | | 400 BAD REQUEST | | Такой навык уже есть. | | |
| PostSkill | POST | | 400 BAD REQUEST | | Нет такого пользователя. | | |
| DeleteSkill | DELETE | | 404 NOT FOUND | | Навыка с таким ID нет. | | |

Таблица 7.2

Описание ошибок в методах класса TransactionsController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Ошибки | | Описание | | |
| GetTransaction | GET | | 404 NOT FOUND | | Обмена с таким ID нет. | | |
| GetTransactions | GET | | - | | - | | |
| PutTransaction | PUT | | 400 BAD REQUEST | | Переданный ID не равен ID обмена. | | |
| PutTransaction | PUT | | 404 NOT FOUND | | Обмена с таким ID нет. | | |
| PostTransaction | POST | | 400 BAD REQUEST | | Такой обмен уже есть. | | |
| PostTransaction | POST | | 400 BAD REQUEST | | Нет такого отправителя или получателя, или отправитель равен получателю. | | |
| DeleteTransaction | DELETE | | 404 NOT FOUND | | Обмена с таким ID нет. | | |

Таблица 7.3

Описание ошибок в методах класса UsersController.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | |  | |  | |  |
| Имя | HTTP тип | | Ошибки | | Описание | | |
| ForgetPassword | GET | | 400 BAD REQUEST | | Пользователя с такой почтой нет. | | |
| GetUsers | GET | | - | | - | | |
| GetUser | GET | | 404 NOT FOUND | | Пользователя с такой почтой нет. | | |
| GetUsersSkills | GET | | 404 NOT FOUND | | Пользователя с такой почтой нет. | | |
| GetUserActive  Transactions | GET | | 404 NOT FOUND | | Пользователя с такой почтой нет. | | |
| GetUserActive  Transactions | GET | | 404 NOT FOUND | | Нет обменов у пользователя. | | |
| GetUser  Completed  Transactions | GET | | 404 NOT FOUND | | Пользователя с такой почтой нет. | | |
| GetUser  Completed  Transactions | GET | | 404 NOT FOUND | | Нет обменов у пользователя. | | |
| GetUserIn  Transactions | GET | | 404 NOT FOUND | | Пользователя с такой почтой нет. | | |
| GetUserIn  Transactions | GET | | 404 NOT FOUND | | Нет обменов у пользователя. | | |
| GetUserOut  Transactions | GET | | 404 NOT FOUND | | Пользователя с такой почтой нет. | | |
| GetUserOut  Transactions | GET | | 404 NOT FOUND | | Нет обменов у пользователя. | | |
| PutUser | PUT | | 400 BAD REQUEST | | Переданная почта не равна почте пользователя. | | |
| PutUser | PUT | | 404 NOT FOUND | | Пользователя с такой почтой нет. | | |
| PostUser | POST | | 400 BAD REQUEST | | Пользователя с такой почтой нет. | | |
| DeleteUser | DELETE | | 404 NOT FOUND | | Пользователя с такой почтой нет. | | |

# ПРИЛОЖЕНИЕ 8

## АВТОМАТИЧЕСКИ ГЕНЕРИРУЕМЫЕ ПИСЬМА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ

Все письма отправляются с почты sharing-skills@sharing-skills.xyz (Обмен Навыками).

1. Письмо при регистрации нового пользователя:

Тема –

Код подтверждения регистрации

Само письмо –

Здравствуйте!

Спасибо за регистрацию в Обмене Навыками

Ваш код подтверждения - 1234

1. Письмо для напоминания забытого пароля

Тема –

Напоминание пароля

Само письмо –

Здравствуйте!

В Обмене Навыками была нажата кнопка "Забыли пароль?"

Ваш пароль – qwerty123

1. Оповещение о новом обмене

Тема –

Новый обмен

Само письмо –

Царёв Кирилл 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 –

Царёв Кирилл отказал(а) Вам в обмене.

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

1. Оповещение о принятии обмена

Тема –

Принятый обмен

Само письмо –

Царёв Кирилл принял(а) Ваш обмен.

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

1. Оповещение о завершении обмена

Тема –

Обмен завершен

Само Письмо –

Царёв Кирилл завершил(а) обмен.

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 | Аннулированных |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |