**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.**  **Программа и методика испытаний**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.06.02-01 51 01-1-ЛУ** | | |
|  |  | |
| Исполнители:  студент группы БПИ193  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Штанько Е.О. /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.  студент группы БПИ193  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Царёв К.Д. /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. | |
|  | | |
|  | |  |

**Москва 2021**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 46** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2021**

**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 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 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](#_Toc72267914)

[1.1. Наименование 5](#_Toc72267915)

[1.2. Область применения 5](#_Toc72267916)

[1.3. Обозначение испытуемой программы 5](#_Toc72267917)

[2. ЦЕЛЬ ИСПЫТАНИЙ 6](#_Toc72267918)

[3. ТРЕБОВАНИЯ К ПРОГРАММЕ 7](#_Toc72267919)

[3.1. Требования к функциональным характеристикам 7](#_Toc72267920)

[3.1.1. Требования к составу выполняемых функций 7](#_Toc72267921)

[3.1.2. Требования к организации входных данных 8](#_Toc72267922)

[3.1.3. Требования к организации выходных данных 8](#_Toc72267923)

[3.2. Требования к надежности 8](#_Toc72267924)

[3.3. Требования к интерфейсу 8](#_Toc72267925)

[4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 10](#_Toc72267926)

[4.1. Состав программной документации 10](#_Toc72267927)

[4.2. Специальные требования к программной документации 10](#_Toc72267928)

[5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ 11](#_Toc72267929)

[5.1. Технические средства, используемые во время испытаний 11](#_Toc72267930)

[5.2. Программные средства, используемые во время испытаний 11](#_Toc72267931)

[5.3. Порядок проведения испытаний 11](#_Toc72267932)

[5.4. Условия проведения испытаний 11](#_Toc72267933)

[5.4.1. Климатические условия 11](#_Toc72267934)

[5.4.2. Требования к численности и квалификации персонала 11](#_Toc72267935)

[6. МЕТОДЫ ИСПЫТАНИЙ 12](#_Toc72267936)

[6.1. Испытание выполнения требований к программной документации 12](#_Toc72267937)

[6.2. Испытание выполнения требований к интерфейсу взаимодействия клиентов и сервера 12](#_Toc72267938)

[6.3. Испытание выполнения требований к функциональным характеристикам, интерфейсу, организации входных и выходных данных, требований к надежности и к временным характеристикам 13](#_Toc72267939)

[6.3.1. Начальный экран 13](#_Toc72267940)

[6.3.2. Авторизация 13](#_Toc72267941)

[6.3.3. Напоминание забытого пароля по почте 14](#_Toc72267942)

[6.3.4. Регистрация 15](#_Toc72267943)

[6.3.5. Общая архитектура UI приложения 23](#_Toc72267944)

[6.3.6. Обзор страницы профиля 23](#_Toc72267945)

[6.3.7. Описание процесса редактирования профиля. 24](#_Toc72267946)

[6.3.8. Обзор страниц, содержащих навыки, которыми владеет пользователь "Могу", и которые он желает получить "Хочу" 26](#_Toc72267947)

[6.3.9. Описание процесса добавления навыка 27](#_Toc72267948)

[6.3.10. Описание процесса редактирования навыка 28](#_Toc72267949)

[6.3.11. Описание процесса удаления навыка 29](#_Toc72267950)

[6.3.12. Обзор страницы поиска навыков 29](#_Toc72267951)

[6.3.13. Описание процесса поиска навыков 30](#_Toc72267952)

[6.3.14. Обзор страницы другого пользователя 32](#_Toc72267953)

[6.3.15. Описание процесса предложения обмена 33](#_Toc72267954)

[6.3.16. Обзор страницы текущих обменов 34](#_Toc72267955)

[6.3.17. Описание процесса завершения текущего обмена 35](#_Toc72267956)

[6.3.18. Обзор страницы исходящих обменов 36](#_Toc72267957)

[6.3.19. Описание процесса удаления исходящего обмена 36](#_Toc72267958)

[6.3.20. Описание процесса редактирования исходящего обмена 37](#_Toc72267959)

[6.3.21. Обзор страницы входящих обменов 37](#_Toc72267960)

[6.3.22. Описание процесса согласия на обмен 38](#_Toc72267961)

[6.3.23. Описание процесса отказа в обмене 39](#_Toc72267962)

[6.3.24. Обзор страницы завершенных обменов 39](#_Toc72267963)

[6.3.25. Вывод о соответствии программы требованиям 39](#_Toc72267964)

[ПРИЛОЖЕНИЕ 1 40](#_Toc72267965)

[ПРИЛОЖЕНИЕ 2 41](#_Toc72267966)

[ПРИЛОЖЕНИЕ 3 43](#_Toc72267967)

[ИДЕНТИФИКАТОРЫ ЛИЧНЫХ ДАННЫХ И КАТЕГОРИЙ НАВЫКОВ 43](#_Toc72267968)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 46](#_Toc72267969)

# ОБЪЕКТ ИСПЫТАНИЙ

## Наименование

Наименование программы – Приложение 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.

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, изложенных 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 в приложении
2. Авторизация в приложении
3. Напоминание забытого пароля по почте
4. Возможность просматривать список всех навыков других пользователей
5. Возможность поиска навыка с использованием поисковой строки
6. Возможность поиска навыка с использованием фильтров:

* Категория
* Подкатегория (только при указании категории)
* Курс
* Пол
* Образовательная программа
* Расположение корпуса
* Общежитие
* "Может/Хочет"

1. Возможность просматривать профили других пользователей

В частности:

* личные данные
* список навыков, которыми владеет другой пользователь
* список навыков, которые желает получить другой пользователь

1. Возможность просматривать текущие обмены
2. Возможность просматривать завершенные обмены
3. Возможность просматривать входящие обмены
4. Возможность просматривать исходящие предложения
5. Возможность просматривать собственный профиль

В частности:

* личные данные
* список своих навыков
* список желаемых для получения навыков

1. Возможность устанавливать/редактировать информацию собственного профиля

В частности:

* Информацию о своих навыках
* Информацию о личных данных
* Информацию о желаемых для получения навыках

1. Возможность предложить обмен
2. Возможность удалить исходящий обмен
3. Возможность редактировать исходящий обмен
4. Возможность согласиться на входящий обмен
5. Возможность отказаться от входящего обмена
6. Возможность завершить текущий обмен

### Требования к организации входных данных

Программа должна обрабатывать строки, числа, файлы и прочие данные различных форматов, запрашиваемые от пользователя через графический интерфейс приложения.

### Требования к организации выходных данных

Выходными данными программы должны являться строки, числа, файлы и прочие данные различных форматов, получаемые с сервера и демонстрируемые пользователям через графический интерфейс приложения, а также автоматически генерируемые электронные письма, отправляемые по почте пользователям.

## Требования к надежности

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

## Требования к интерфейсу

* Приложение должно обладать понятным интерфейсом
* Интерфейс должен предоставляющий пользователю возможность выполнять следующий состав функций:

1. Авторизация
2. Регистрация с использованием корпоративной почты
3. Напоминание забытого пароля по почте
4. Просмотр своего профиля
5. Просмотр своих навыков
6. Добавление своего навыка
7. Редактирование своего навыка
8. Просмотр навыков, желаемых для получения
9. Добавление желаемого для получения навыка
10. Редактирования желаемого для получения навыка
11. Редактирование профиля
12. Просмотр текущих обменов
13. Завершение текущего обмена
14. Просмотр завершенных обменов
15. Просмотр входящих обменов
16. Согласие на обмен
17. Отказ в обмене
18. Просмотр исходящих обменов
19. Редактирование исходящего обмена
20. Удаление исходящего обмена
21. Предложение обмена
22. Просмотр профиля другого пользователя
23. Просмотр навыков, которыми владеет другой пользователь
24. Просмотр навыков, которые желает получить другой пользователь
25. Просмотр списка всех навыков других пользователей
26. Поиск навыка с использованием поисковой строки
27. Поиск навыка с использованием фильтров:

* Категория
* Подкатегория (только при указании категории)
* Курс
* Пол
* Образовательная программа
* Расположение корпуса
* Общежитие
* "Может/Хочет"
* Дизайн системы должен быть однородным, выполненным в спокойных тонах

Клиенты 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)
8. Техническое задание «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Клиентская часть.» (ГОСТ 19.201-78)
9. Программа и методика испытаний «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Клиентская часть.» (ГОСТ 19.301-79)
10. Пояснительная записка «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Клиентская часть.» (ГОСТ 19.404-79)
11. Текст программы «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Клиентская часть.» (ГОСТ 19.401-78)
12. Руководство оператора «Клиент-серверное приложение для обмена навыками и услугами для студентов НИУ ВШЭ. Клиентская часть.» (ГОСТ 19.505-79)

## 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 ЛМС 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 LMS (Learning management system) в личном кабинете, дисциплина - «Курсовой проект», одним архивом.

# СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

## Технические средства, используемые во время испытаний

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

1. Мобильный телефон со следующими минимальными требованиями:

* Операционная система Android версии 4.4.4 и выше или iOS версии 10.3.4 и выше
* 64-разрядный (x64) процессор
* 1 ГБ оперативной памяти (ОЗУ)
* 100 МБ свободного места на внутреннем накопителе

Для корректной работы серверной части приложения требуется следующий состав технических средств и характеристик:

* 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 сервера требуется следующий состав программных средств:

1. 32-битная операционная система Windows

Для использования приложения необходим следующий состав программных средств:

1. Операционная система Android версии 4.4.4 и выше
2. Операционная система iOS версии 10.3.4 и выше

## 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 компакт-диска (CD) 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 всей системы, должно составлять не менее 1 человека – системного администратора, знающего язык программирования C# и имеющего навыки работы с ASP.Net (для серверной части).

# МЕТОДЫ ИСПЫТАНИЙ

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

Перед проведением испытаний необходимо:

* для Android: скачать приложение из PlayMarket-a, которое можно найти по ссылке (https://play.google.com/store/apps/details?id=skillsHSE.courseWork) или, используя QR-код (рис.1) (Откройте встроенное приложение камеры на совместимом телефоне или планшете Android. Наведите камеру на QR-код. Нажмите на баннер, который появится на экране. Следуйте инструкциям).



Рисунок 1 - QR-код для скачивания приложения в PlayMarket

* для iOS: открыть папку с файлом "SharingSkillsHSE.zip" на флэш-накопителе и скопировать папку на жесткий диск. Для установки программы необходимо разархивировать "SharingSkillsHSE.zip" в желаемую папку. Затем, необходимо открыть проект с помощью среды разработки "IntelliJ IDEA", предварительно убедившись в том, что на устройстве установлено SDK Flutter [12] и среда разработки "Xcode" (для операционной системы Windows можно воспользоваться виртуальной системой macOS с помощью программы "VMware Workstation 15 Pro"). Подключить устройство под управлением iOS 10.3.3 или более поздней версии и собрать проект на устройство.

## Испытание выполнения требований к программной документации

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

## Испытание выполнения требований к интерфейсу взаимодействия клиентов и сервера

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, интерфейсу, организации входных и выходных данных, требований к надежности и к временным характеристикам

Начальный экран

Начальный экран (рис.2) содержит текстовые поля ввода почты и пароля для авторизации, кнопки "Вход" (подтверждение авторизации и переход на страницу профиля), "Регистрация" (переход на страницу начала регистрации), "Восстановление пароля" (переход на страницу восстановления пароля).

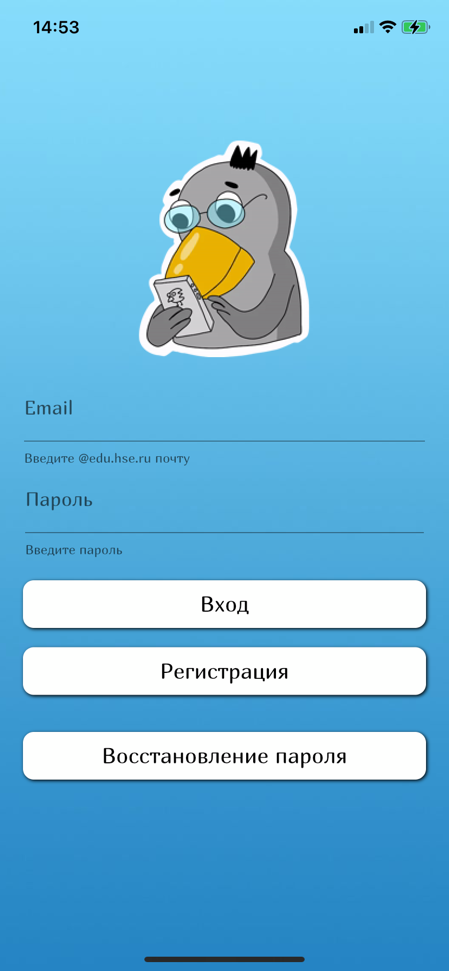
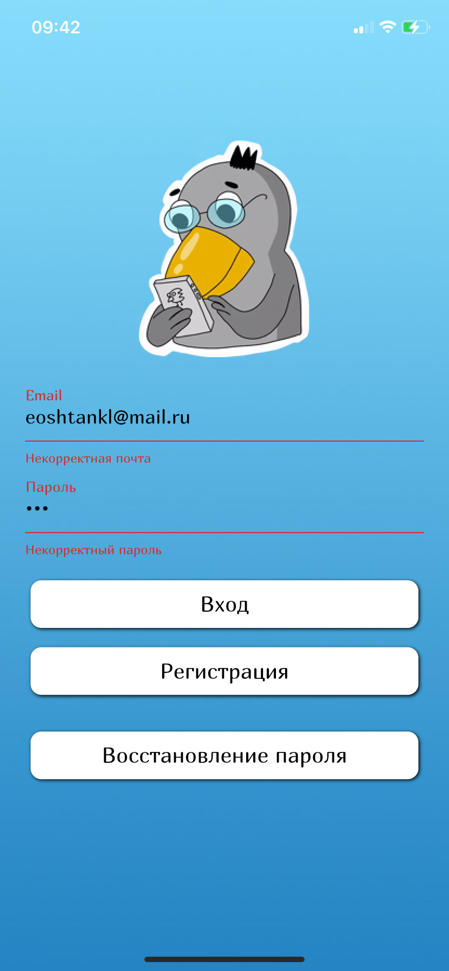


Рисунок 2 - Начальный экран

Рисунок 4 - Сообщение выводимое при попытке авторизоваться, введя неверный пароль или логин

Рисунок 3 - Введенные данные не прошли первичную проверку при попытке авторизации

Авторизация

После нажатия кнопки "Вход" осуществляется первичная проверка данных (оканчивается ли введенная почта на "@edu.hse.ru" и содержится ли в ней хотя бы 12 символов, содержит ли пароль допустимое количество символов (от 6 до 40)).

Если введенные данные не прошли проверку, выводится информирующее сообщение (рис.4).

Если введенные данные прошли проверку, осуществляется GET-запрос (api/Users/${email}, где email - введенный адрес почты) с целью получения пароля данного пользователя, хранящегося на сервере.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 404, означающий отсутствие в базе данных пользователя, привязанного к введенной почте, выводится соответствующее сообщение (рис.3).

Если сервер вернул ответ с кодом 200, производится сравнение пароля, хранящегося на сервере, и пароля, введенного пользователем. Если пароли не совпадают - выводится соответствующее сообщение (рис.4), иначе - авторизация завершается успешно и происходит переход на страницу профиля.

Авторизация реализована корректно.

Напоминание забытого пароля по почте

Страница (рис.5) содержит поле для ввода почты, а также кнопку "Дальше". После нажатия кнопки "Дальше" осуществляется первичная проверка введенных в текстовое поле данных (оканчивается ли введенная почта на "@edu.hse.ru" и содержится ли в ней хотя бы 12 символов).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется GET-запрос (api/Users/${email}/password, где email - введенный адрес почты).

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200 (операция прошла успешно, и на почту пользователю был выслан пароль), происходит переход к начальному экрану.

Если сервер вернул ответ с кодом 400 (пользователь с такой почтой не зарегистрирован), выводится соответствующее сообщение.

Интерфейс страницы восстановления пароля соответствует требованиям.

Восстановление реализовано корректно.

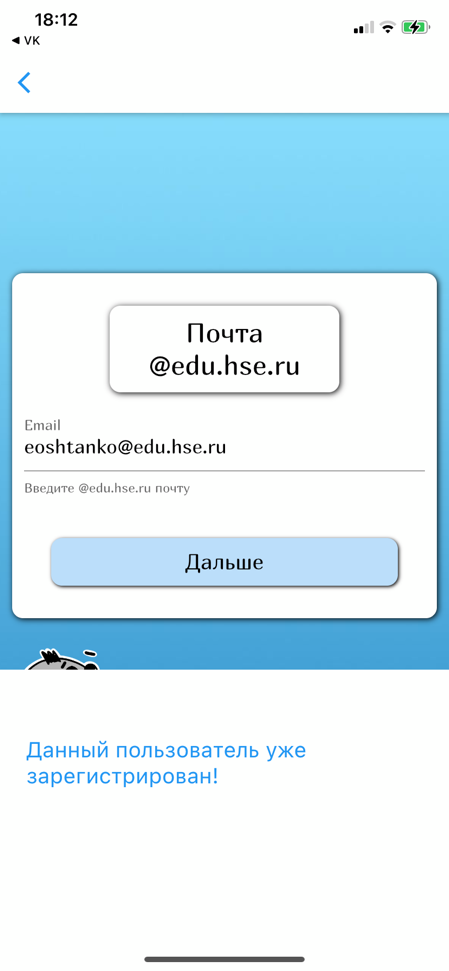


Рисунок 6 - Попытка регистрации уже существующего пользователя

Рисунок 6 - Попытка регистрации уже существующего пользователя

Рисунок 5 - Страница ввода почты

Регистрация

Регистрация состоит из 11 страниц.

1. Первая страница - ввод почты.

Первая регистрации страница (рис.5) содержит поле для ввода адреса почты и кнопку "Дальше". После нажатия кнопки "Дальше" осуществляется первичная проверка данных (оканчивается ли введенная почта на "@edu.hse.ru").

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется GET-запрос (api/Users/${email}, где email - введенный адрес почты) с целью получения информации о наличии данного пользователя в базе данных.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если пользователь в базе данных уже существует (ответ сервера содержит код 200), и у него заполнены все обязательные поля (имя, фамилия, пароль, дата рождения, пол, факультет, курс, общежитие, корпус), выводится соответствующее сообщение (рис.6), и регистрация прерывается.

Если такой пользователь в базе данных уже существует (ответ сервера содержит код 200), но у него не все обязательные поля заполнены (это свидетельствует о том, что пользователь в прошлом начинал регистрацию, но не завершил ее), осуществляется DELETE - запрос (api/Users/${email}, где email - введенный адрес почты) , удаляющий данного пользователя в базе данных.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

В последнем вышеописанном случае и в случае, если ответ сервера содержит код 404 (пользователь с такой почтой еще не зарегистрирован), осуществляется POST - запрос (api/Users/${email}, где email - введенный адрес почты), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу верификации почты.

Если сервер вернул ответ с иным нежели 200, выводится сообщение об ошибке.

1. Вторая страница - ввод кода подтверждения.

Вторая страница регистрации (рис.7) содержит поле для ввода кода, который отправляется сервером на указанную почту для верификации при POST - запросе, описанном выше, и кнопку "Дальше". После нажатия кнопки "Дальше" осуществляется первичная проверка данных (код состоит из 4 символов).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется PUT - запрос (api/Users/${email}, где email - введенный на первой странице регистрации адрес почты), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу установления пароля.

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

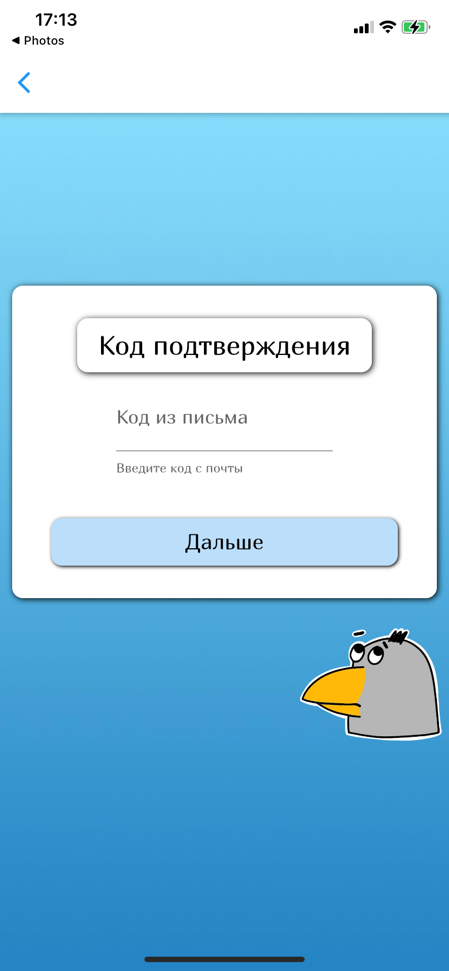


Рисунок 7 - Страница ввода кода подтверждения почты

1. Третья страница - ввод пароля.

Третья страница содержит поле для ввода пароля и кнопку "Дальше". После нажатия кнопки "Дальше" осуществляется первичная проверка данных (Пароль должен содержать не менее 6 символов и не более 40).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется переход к странице повторения пароля.

1. Четвертая страница - повтор пароля.

Четвертая страница содержит поле для повтора пароля и кнопку "Дальше". После нажатия кнопки "Дальше" осуществляется проверка на совпадение введенных паролей).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется PUT - запрос (api/Users/${email}, где email - введенный на первой странице регистрации адрес почты), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу ввода имени и фамилии.

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

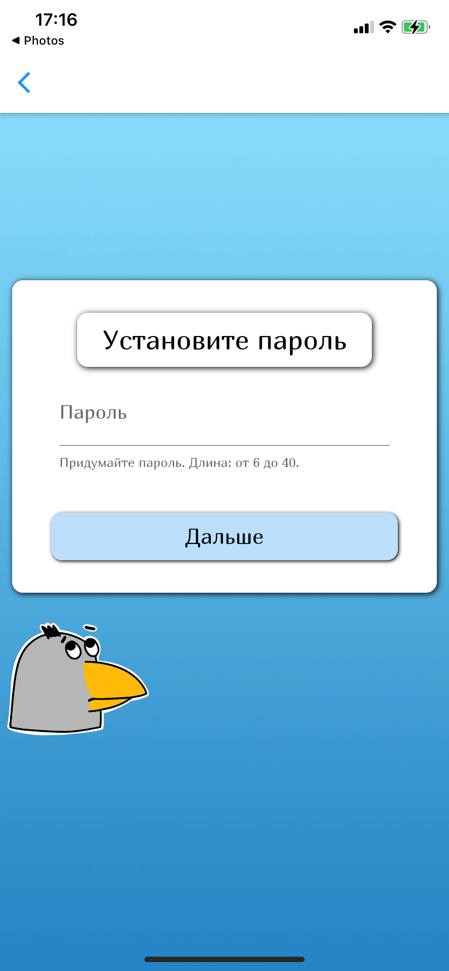
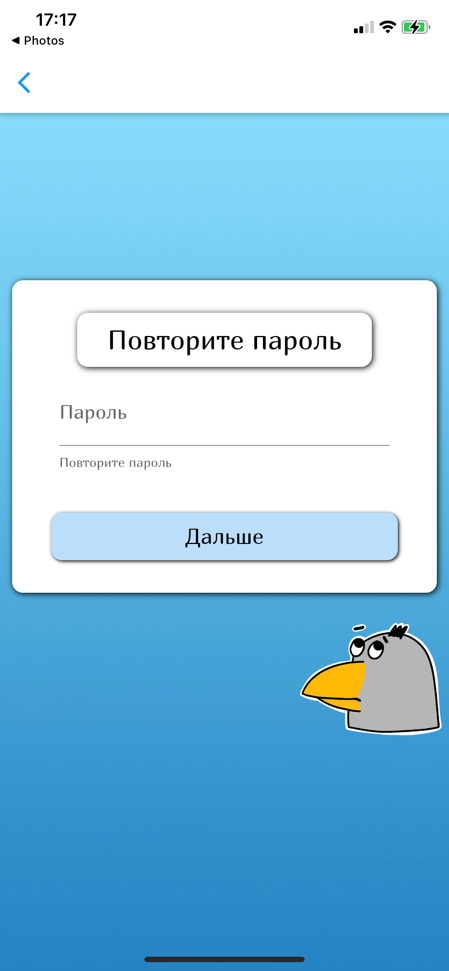


Рисунок 9 - Страница подтверждения установленного пароля

Рисунок 8 - Страница установления пароля

1. Пятая страница - ввод имени и фамилии.

Пятая страница (рис.10) содержит поля для ввода имени и фамилии, а также, кнопку "Дальше". После нажатия кнопки "Дальше" осуществляется первичная проверка данных (имя и фамилия должны содержать не менее двух символов).

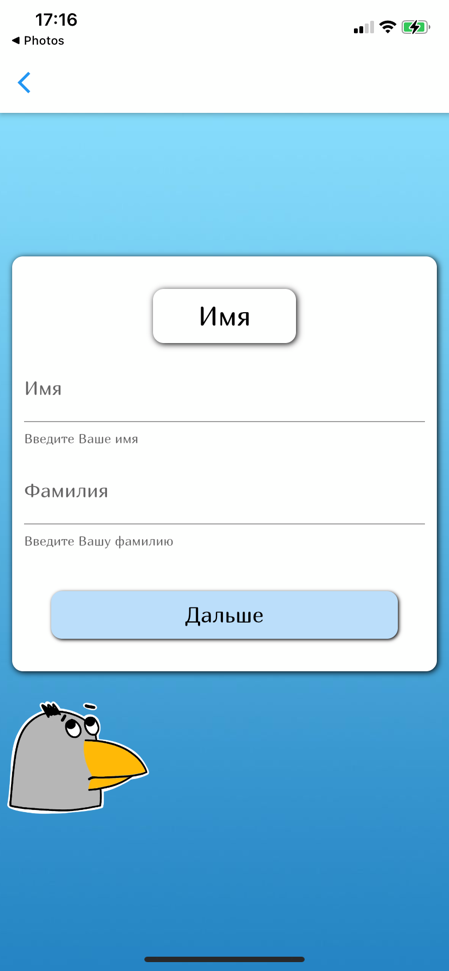


Рисунок 10 - Страница ввода имени и фамилии

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется PUT - запрос (api/Users/${email}, где email - введенный на первой странице регистрации адрес почты), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу ввода даты рождения.

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

1. Шестая страница - дата рождения.

Шестая страница содержит поле типа CupertinoDatePicker (см. приложение) для ввода даты рождения и кнопку "Дальше". После нажатия кнопки "Дальше" осуществляется первичная проверка данных (пользователь выбрал дату).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется PUT - запрос (api/Users/${email}, где email - введенный на первой странице регистрации адрес почты), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу ввода информации о поле.

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 - Страница ввода даты рождения. Иллюстрация CupertinoDatePicker- а

Рисунок 11 - Страница ввода даты рождения

1. Седьмая страница - выбор пола.

Седьмая страница (рис.13) содержит две кнопки "Мужской" и "Женский" для ввода информации о поле. После нажатия кнопки "Дальше" осуществляется PUT - запрос (api/Users/${email}, где email - введенный на первой странице регистрации адрес почты), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу ввода информации о курсе.

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

1. Восьмая страница - ввод информации о курсе.

Восьмая страница (рис.14) содержит поле типа DropdownButton (см. приложение) для ввода информации о курсе. После нажатия кнопки "Дальше" осуществляется первичная проверка данных (пользователь выбрал курс).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется PUT - запрос (api/Users/${email}, где email - введенный на первой странице регистрации адрес почты), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу ввода информации об образовательной программе.

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

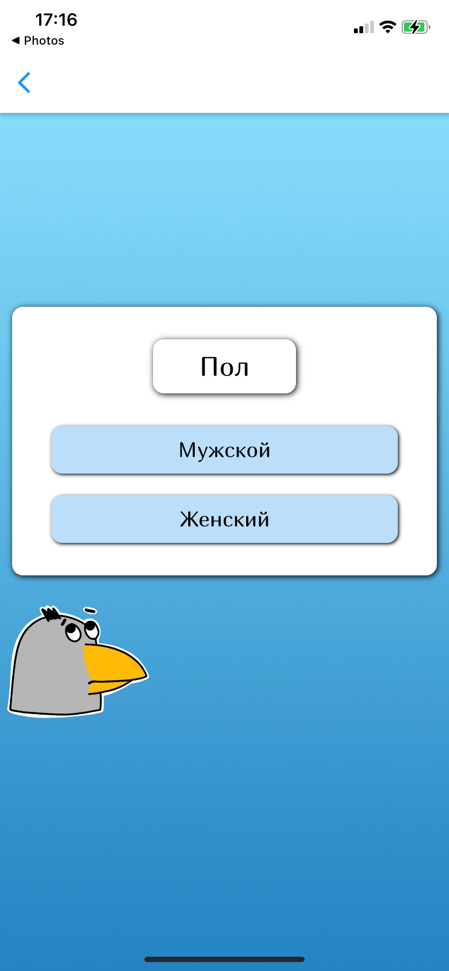


Рисунок 13 - Страница указания информации о поле

Рисунок 15 - Страница указания информации об образовательной программе

Рисунок 14 - Страница указания информации о курсе

1. Девятая страница - ввод информации об образовательной программе.

Девятая страница (рис.15) содержит поле типа DropdownButton для ввода информации об образовательной программе. После нажатия кнопки "Дальше" осуществляется первичная проверка данных (пользователь выбрал образовательную программу).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется PUT - запрос (api/Users/${email}, где email - введенный на первой странице регистрации адрес почты), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу ввода информации о расположении корпуса.

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

1. Десятая страница - ввод информации о корпусе.

Десятая страница (рис.16) содержит поле типа DropdownButton (рис. 17) для ввода информации о расположении корпуса. После нажатия кнопки "Дальше" осуществляется первичная проверка данных (пользователь выбрал расположение корпуса).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется PUT - запрос (api/Users/${email}, где email - введенный на первой странице регистрации адрес почты), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу ввода информации об общежитии.

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

Изображение выглядит как текст

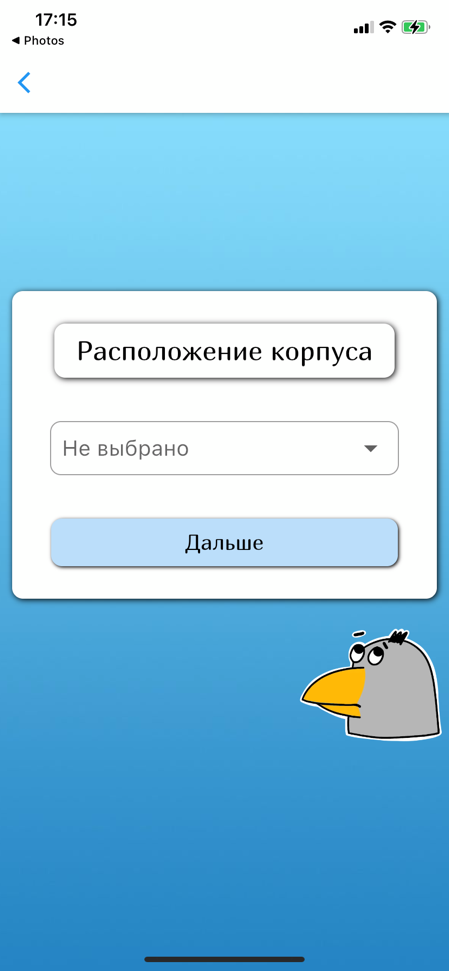
Автоматически созданное описание

Рисунок 18 - Страница указания информации об общежитии

Рисунок 17 - Страница указания информации о расположении корпуса. Иллюстрация DropdownButton

Рисунок 16 - Страница указания информации о расположении корпуса

Рисунок 16 - Страница указания информации о расположении корпуса

Рисунок 17 - Страница указания информации о расположении корпуса. Иллюстрация DropdownButton

Рисунок 18 - Страница указания информации об общежитии

1. Одиннадцатая страница - ввод информации о корпусе.

Одиннадцатая страница (рис.18) содержит поле типа DropdownButton для ввода информации об общежитии. После нажатия кнопки "Дальше" осуществляется первичная проверка данных (пользователь выбрал общежитие).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли проверку, осуществляется PUT - запрос (api/Users/${email}, где email - введенный на первой странице регистрации адрес почты), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу обзорной информации о приложении.

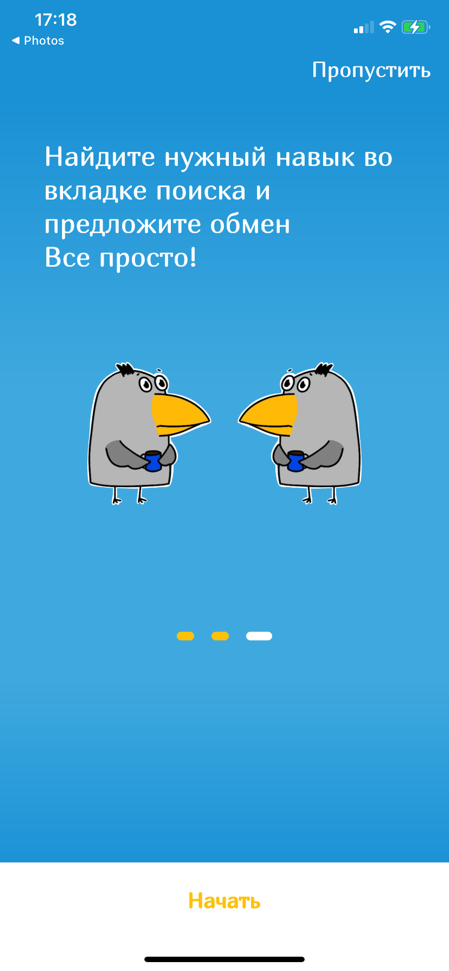
Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

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

Таким, образом интерфейс и функциональная составляющая регистрации корректны. Раздел регистрации полностью соответствует требованиям к входным и выходным данным, а также требованиям надежности.

Вводная информация.

Завершается регистрация предоставлением пользователю небольшой (3 страницы), обзорной информации о сути приложения (рис.19, рис.20, рис.21).

Изображение выглядит как текст, снимок экрана

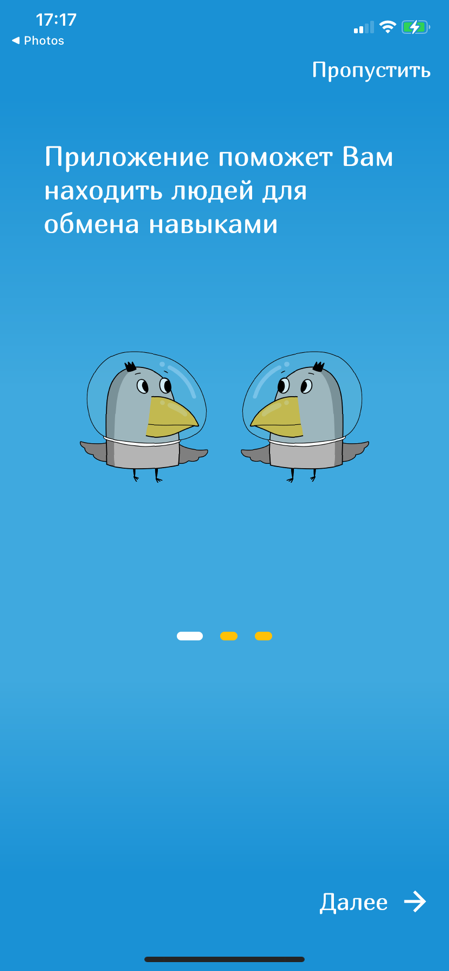
Автоматически созданное описание

Рисунок 21 - Страница обзорной информации №3

Рисунок 20 - Страница обзорной информации №2

Рисунок 19 - Страница обзорной информации №1

На всех трех страницах в правом верхнем углу находится кнопка "Пропустить", с ее помощью можно пропустить обзорную информацию и перейти к работе с приложением (к странице с профилем (3.4.7.)).

Страницы можно перелистывать с помощью технологии swipe (способ ввода текста без отрыва пальца/стилуса от «кнопок клавиатуры» на сенсорном экране) или используя кнопку "Далее", расположенную на первых двух страницах в правом нижнем углу.

На последней странице внизу располагается кнопка "Начать", с ее помощью можно перейти к работе с приложением (к странице с профилем).

Общая архитектура UI приложения



Рисунок 22 - Нижняя панель приложения

На нижней панели приложения (рис.22) находятся три кнопки:

* + 1. Поиск

При нажатии осуществляется открытие страницы, где располагаются все навыки других пользователь, а также средства для их поиска (3.4.13). Прежде, чем перейти на страницу, с сервера загружаются все навыки других пользователей через GET-запрос (api/Users/${email}/skills, где email - адрес почты текущего пользователя). Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке. Если сервер возвращает код, свидетельствующий об ошибке - выводится соответствующее сообщение.

* + 1. Обмены

При нажатии осуществляется открытие страницы, где организованы все обмены текущего пользователя (3.4.17). Прежде, чем перейти на страницу, с сервера загружаются все текущие обмены через GET-запрос (api/Users/${email}/transactions/active, где email - адрес почты текущего пользователя). Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке. Если сервер возвращает код, свидетельствующий об ошибке - выводится соответствующее сообщение.

* + 1. Профиль

При нажатии осуществляется открытие страницы профиля (3.4.7).

Обзор страницы профиля

Профиль (рис.23, рис.24, рис.25) содержит поле "о себе", информацию об имени, фамилии, почте, социальной сети, образовательной программе, общежитии, корпусе, курсе, поле, дате рождения, фотографию и кнопку для ее изменение, копки редактирование профиля и "Log out" (выход), кнопки для перехода к просмотру навыков, которыми пользователь владеет "Могу" и тех, которые пользователь желает получить "Хочу".

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Рисунок 24 - Страница профиля (2)

Рисунок 23 - Страница профиля (1)

Рисунок 25 - Страница профиля (3)

Описание процесса редактирования профиля.

Редактирование имени, фамилии, даты рождения, пароля, сведений о поле, курсе, образовательной программе, корпусе, общежитии, социальной сети, поля "о себе".

При нажатии на кнопку редактирования (рис.26) состояние профиля меняется (рис.27) и поля становятся доступными для изменения.



Рисунок 26 - Кнопка редактирования профиля

При нажатии на кнопку сохранить (рис. 28) происходит проверка, были ли сделаны какие-либо изменения, а также первичная проверка корректности введенных данных (если данные были введены некорректно, будет выведено соответствующее сообщение). Если изменений сделано не было, состояние профиля меняется и поля становятся недоступными для изменения. Если изменения были сделаны, осуществляется PUT - запрос (api/Users/${email}, где email - адрес почты текущего пользователя), в теле которого передается сериализованный в формат JSON экземпляр класса Profile.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, состояние профиля меняется и поля становятся недоступными для изменения

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

Изображение выглядит как текст

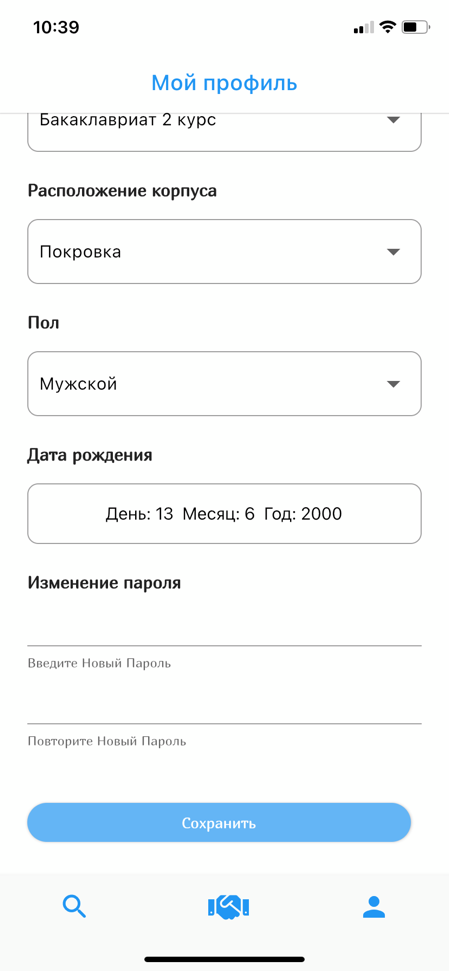
Автоматически созданное описание

Рисунок 28 - Профиль в состоянии редактирования (2)

Рисунок 27 - Профиль в состоянии редактирования (1)

Изменение фотографии.

Изображение выглядит как бильярдный шар

Автоматически созданное описание

Рисунок 29 - Кнопка загрузка фотографии

Получение изображения реализована посредством библиотеки "Image Picker".

При нажатии на кнопку изменения фотографии (рис.29), открывается окно выбора источника (рис.30).

Если источником выбирается галерея (ImageSource.gallery), то изображение будет выбрано из локального хранилища.

Если пользователь выбирает "Сделать фотографию" (ImageSource.camera), то при наличии аппаратной поддержки, открывается камера.

После выбора изображения из галереи/снимка открывается страница обработки фотографии (рис.31 и рис.32).

Обработка фотографии реализуется посредством библиотеки "Image Cropper". Доступный пользователю функционал: поворот фотографии, масштабирование, обрезка фотографии под определенное соотношение сторон.

Изображение выглядит как текст, экран, электроника, компьютер

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст, экран

Автоматически созданное описание

Рисунок 30 - Окно выбора источника фотографии

Рисунок 31 - Окно редактирования фотографии (1)

Рисунок 32 - Окно редактирования фотографии (2)

Обзор страниц, содержащих навыки, которыми владеет пользователь "Могу", и которые он желает получить "Хочу"

Страницы (рис.33 и рис.34) схожи по своему содержанию. Они содержат "карточки" навыков, в каждой из которых указаны наименование, описание, категория и подкатегория навыка, есть опции "удалить навык" и "редактировать навык". Также, в правом нижнем углу страницы присутствует кнопка добавления навыка.

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Рисунок 34 - Страница, содержащая желаемые для получения текущим пользователем навыков

Рисунок 33 - Страница, содержащая навыки, которыми владеет текущий пользователь

Описание процесса добавления навыка

При нажатии на кнопку добавления навыка, происходит переход на страницу добавления навыка.

Страница (рис.35, рис.36) содержит два текстовых поля: для ввода наименования навыка и для ввода описания навыка, два поля типа DropdownButton для выбора категории и подкатегории (причем второе поле появляется только после выбора элемента в первом).

При нажатии кнопки "Сохранить" осуществляется первичная проверка данных (пользователь выбрал категорию и подкатегорию, длина наименования навыка не менее 3 и не более 50 символов).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли первичную проверку, проверяется, не существует ли уже данного навыка у текущего пользователя.

Если такой навык уже существует, выводится соответствующее сообщение.

Если такого навыка еще не существует, осуществляется POST - запрос (api/Skills), в теле которого передается сериализованный в формат JSON экземпляр класса Skill.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 201, происходит возвращение на предыдущую страницу.

Если сервер вернул ответ с иным нежели 201 кодом, выводится сообщение об ошибке.

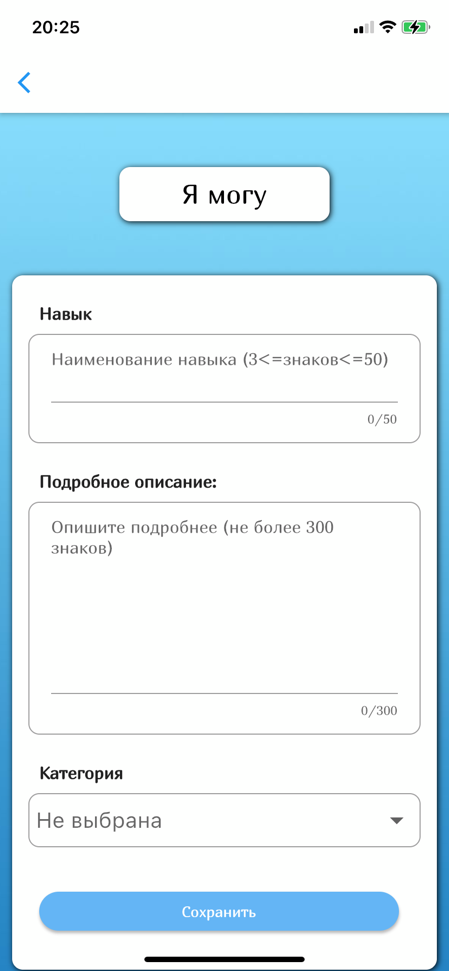
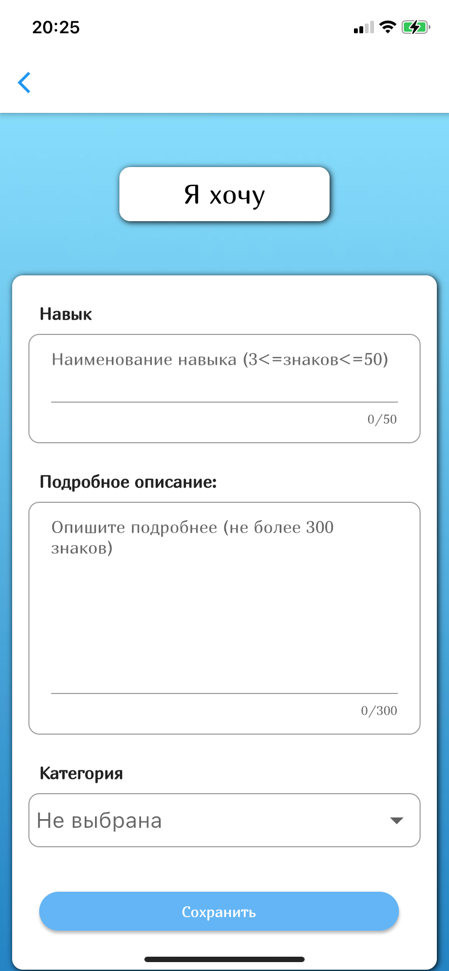


Рисунок 36 - Добавление нового желанного для получения навыка

Рисунок 35 - Добавление нового навыка, которым владеет пользователь

Описание процесса редактирования навыка

При нажатии на кнопку редактирования навыка, происходит переход на страницу редактирования (она выглядит идентично страницам добавления навыков).

Страница содержит два текстовых поля: для ввода наименования навыка и для ввода описания навыка, два поля типа DropdownButton для выбора категории и подкатегории.

При нажатии кнопки "Сохранить" осуществляется первичная проверка данных (длина наименования навыка не менее 3 и не более 50 символов).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли первичную проверку, проверяется, не существует ли уже данного навыка у текущего пользователя.

Если такой навык уже существует, выводится соответствующее сообщение.

Если такого навыка еще не существует, осуществляется PUT - запрос (api/Skills/ ${id}, где id - уникальный идентификатор навыка), в теле которого передается сериализованный в формат JSON экземпляр класса Skill.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 201, происходит возвращение на предыдущую страницу.

Если сервер вернул ответ с иным нежели 201 кодом, выводится сообщение об ошибке.

Описание процесса удаления навыка

При нажатии на кнопку удаления навыка происходит переход на страницу удаления навыка (рис.37), в которой от пользователя требуется либо подтвердить удаление, либо отказаться от удаления.

При отказе от удаления происходит возврат на предыдущую страницу.

При согласии на удаление осуществляется DELETE - запрос (api/Skills/$id, где id - уникальный идентификатор удаляемого навыка).

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 204, происходит возвращение на предыдущую страницу.

Если сервер вернул ответ с иным нежели 204 кодом, выводится сообщение об ошибке.

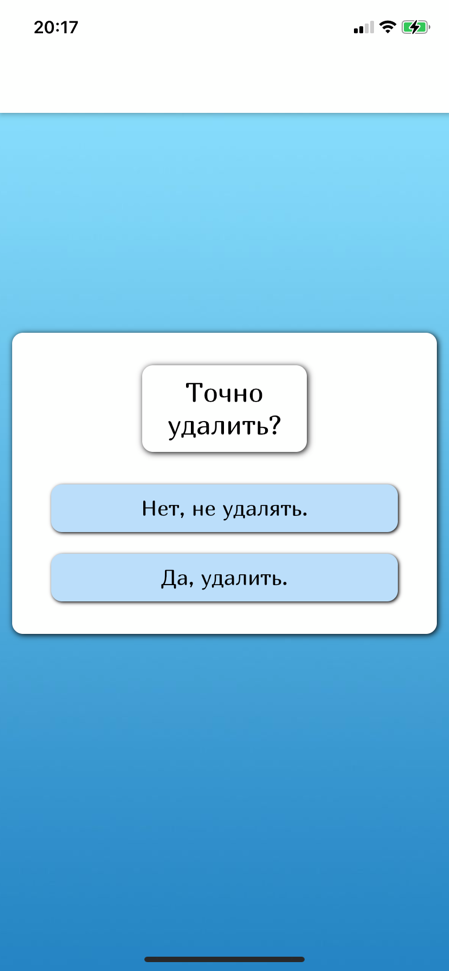


Рисунок 37 - Страница подтверждения удаления навыка

Обзор страницы поиска навыков

Страница поиска навыков (рис.38), содержит все "карточки" навыков других пользователей. На каждой "карточке" содержатся:

* фотография и имя владельца навыка (при нажатии на которые происходит переход в профиль соответствующего пользователя (3.4.15))
* название, описание, категория и подкатегория навыка
* кнопка "Обмен", при нажатии на которую происходит переход к странице создания обмена (3.4.16).

Также, на странице поиска содержатся кнопки:

* обновления (расположена в правом нижнем углу. Запрашивает и отображает все существующие навыки других пользователей. Нужна, главным образом, для того, чтобы сбрасывать результаты, выведенные при поиске с помощью параметров);
* поиска по параметрам (находится в левом верхнем углу. Осуществляет переход на страницу выбора параметров для поиска);
* перехода к поисковой строке (находится в правом верхнем углу. Осуществляет переход к странице с поисковой строкой);

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 38 - Страница поиска навыков

Описание процесса поиска навыков

Приложение предоставляет два способа поиска навыков: поиск с использование поисковой строки, поиск с использованием фильтров.

Поиск через поисковую строку

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Рисунок 40 - Поиск через поисковую строку. Пример фильтрации

Рисунок 39 - Поиск через поисковую строку

Поиск происходит по названиям навыков (рис.39, рис.40). При нажатии на элемент поиска происходит переход к странице назначения обмена (3.4.16).

Поиск по параметрам

Страница поиска по параметрам (рис.41, рис.42) предоставляет 7 полей типа DropdownButton для выбора параметров: "Категория", "Курс", "Пол", "Образовательная программа", "Расположение корпуса", "Общежитие", "Может/хочет".

После нажатия на кнопку "Поиск" осуществляется GET - запрос (api/Users/${email}/skills?studyingYearID=$studyingYearId&majorID=$majorId& 'campusLocationID=$campus&dormitoryID=$dorm&gender=$gender&skillstatus=$status& 'category=$category&subcategory=$subcategory', где email - адрес почты текущего пользователя, все прочие составляющие - коды выбранных параметров).

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит переход на страницу поиска навыков, где содержатся результаты фильтрации.

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

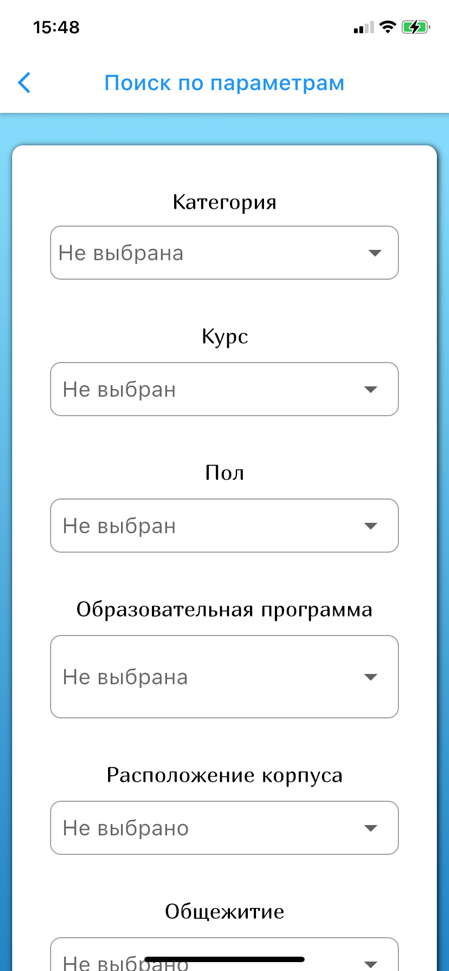
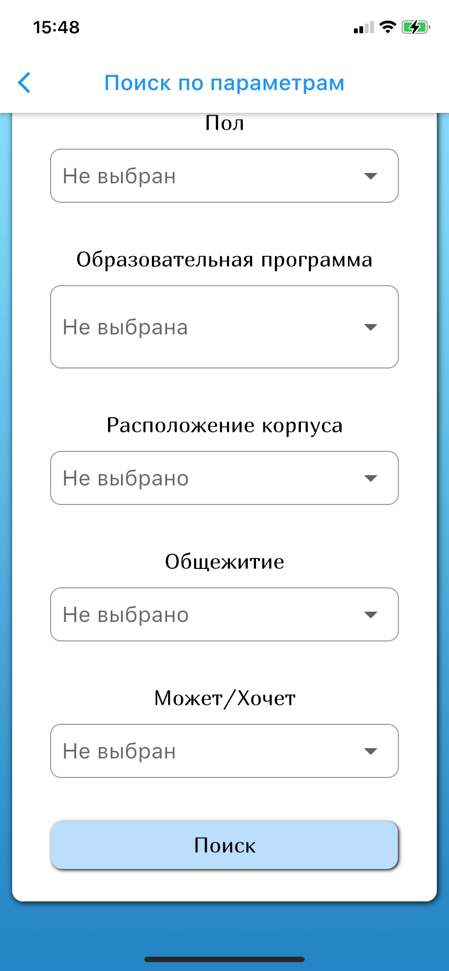


Рисунок 42 - Страница поиска по параметрам (2)

Рисунок 41 - Страница поиска по параметрам (1)

Обзор страницы другого пользователя

Страница другого пользователя (рис. 43, рис.44) содержит поле "о себе", информацию об имени, фамилии, почте, социальной сети, образовательной программе, общежитии, корпусе, курсе, поле, дате рождения, фотографию. Также, на странице присутствуют кнопки "Хочет" и "Может", для перехода к просмотру навыков, которыми пользователь владеет и тех, которые пользователь желает получить соответственно.

Изображение выглядит как текст

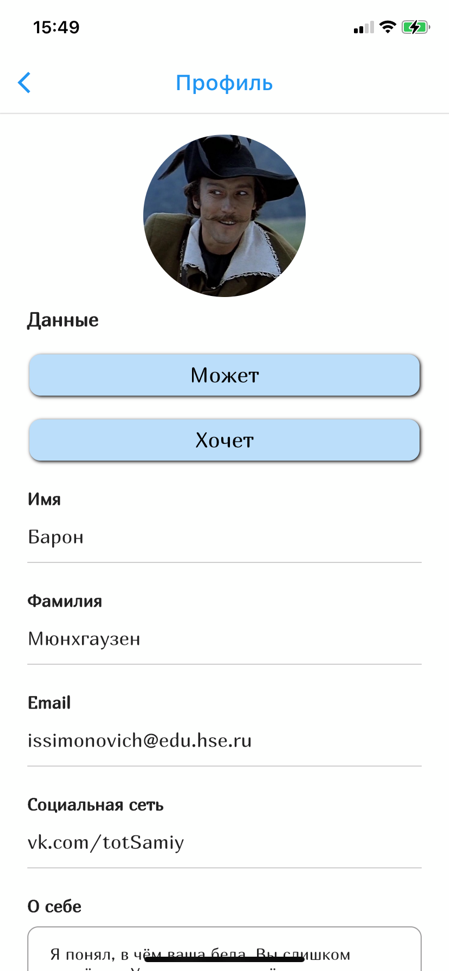
Автоматически созданное описание

Рисунок 44 - Страница другого пользователя (2)

Рисунок 43 - Страница другого пользователя (1)

Описание процесса предложения обмена

Переход к странице предложения обмена (рис.45, рис.46) возможен как из профиля другого пользователя (3.4.15), так и со страницы поиска навыков (3.4.13).

На странице отображены наименование и описание навыка, который был выбран для обмена, фотография, имя и фамилия его обладателя (если переход на страницу предложения обмена был осуществлен со страницы поиска навыков, то при нажатии на фотографию/имя/фамилию можно перейти в профиль соответствующего пользователя. Если же переход был осуществлен со страницы другого пользователя - данная функция отключена).

На странице присутствует поле типа DropdownButton для выбора навыка, который текущий пользователь предлагает взамен (Если текущий пользователь еще не добавил навыки в своем профиле, будет выведено соответствующее сообщение).

При предложении обмена есть возможность добавить сообщение, для этого на странице есть текстовое поле.

После нажатия кнопки "Предложить обмен" осуществляется первичная проверка данных (текущий пользователь выбрал навык, который он предлагает со свей стороны).

Если введенные данные не прошли проверку, выводится информирующее сообщение.

Если введенные данные прошли первичную проверку, проверяется, не существует ли уже данного обмена.

Если такой обмен уже существует, выводится соответствующее сообщение.

Если такого обмена еще не существует, осуществляется POST - запрос (api/Transactions), в теле которого передается сериализованный в формат JSON экземпляр класса Exchange.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 201, происходит возвращение на предыдущую страницу.

Если сервер вернул ответ с иным нежели 201 кодом, выводится сообщение об ошибке.

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

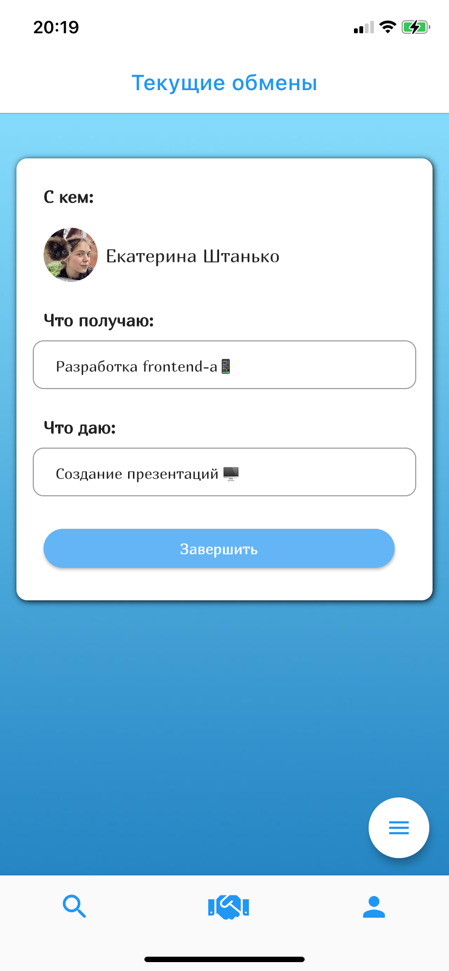
Рисунок 45 - Страница предложения обмена (1)

Рисунок 46 - Страница предложения обмена (2)

Обзор страницы текущих обменов

На странице (рис.47) располагаются "карточки" текущих обменов, в которых содержится информация о том, с кем производится обмен (при нажатии на фотографию/имя/фамилию происходит переход на страницу соответствующего пользователя), наименование получаемого в результате обмена навыка и наименование навыка, который "отдается" в обмен, кнопка завершения обмена.

Также, в правом нижнем углу есть кнопка, при нажатии на которую происходит раскрытие панели кнопок перехода к другим категориям обменов ("Входящие", "Исходящие", "Завершенные") (рис.48).



Изображение выглядит как текст

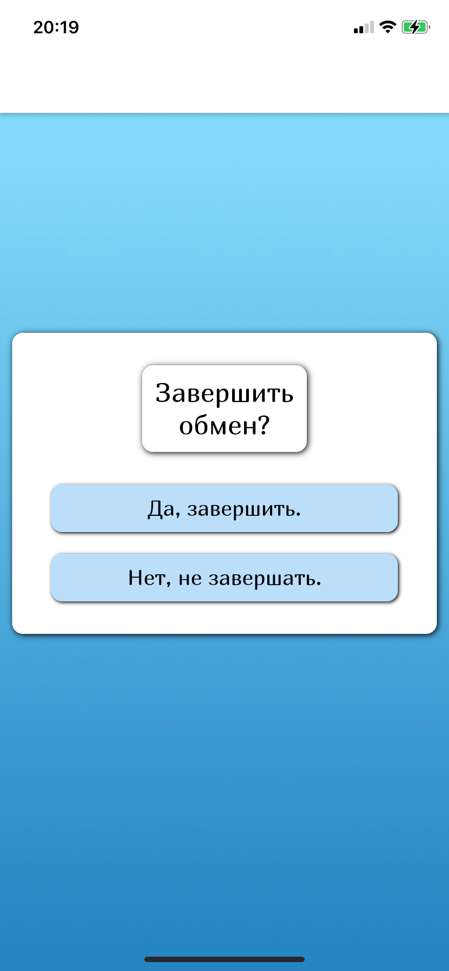
Автоматически созданное описание

Рисунок 49 - Страница завершения текущего обмена

Рисунок 48 - Страница текущих обменов. Демонстрация кнопок

Рисунок 47 - Страница текущих обменов

Описание процесса завершения текущего обмена

При нажатии на кнопку завершения обмена происходит переход на страницу завершения обмена (рис.49), в которой от пользователя требуется либо подтвердить завершение, либо отказаться от завершения.

При отказе от завершения происходит возврат на предыдущую страницу (страницу содержащую текущие обмены).

При согласии на завершение осуществляется PUT - запрос (api/Transactions/${id}?mail=${email}, где id - уникальный идентификатор завершаемого обмена, email - адрес почты текущего пользователя), в теле которого передается сериализованный в формат JSON экземпляр класса Exchange. С помощью данного запроса происходит изменение статуса обмена.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 204, происходит возвращение на страницу текущих обменов.

Если сервер вернул ответ с иным нежели 204 кодом, выводится сообщение об ошибке.

Обзор страницы исходящих обменов

На странице (рис.50) располагаются "карточки" исходящих обменов, в которых содержится информация о том, кому был предложен обмен (при нажатии на фотографию/имя/фамилию происходит переход на страницу соответствующего пользователя (3.4.15)), наименование получаемого в результате обмена навыка и наименование навыка, который "отдается" в обмен, кнопки удаления и редактирования обмена.

Описание процесса удаления исходящего обмена

При нажатии на кнопку удаления обмена происходит переход на страницу удаления обмена (рис.51), на которой от пользователя требуется либо подтвердить удаление, либо отказаться от удаления.

При отказе происходит возврат на предыдущую страницу.

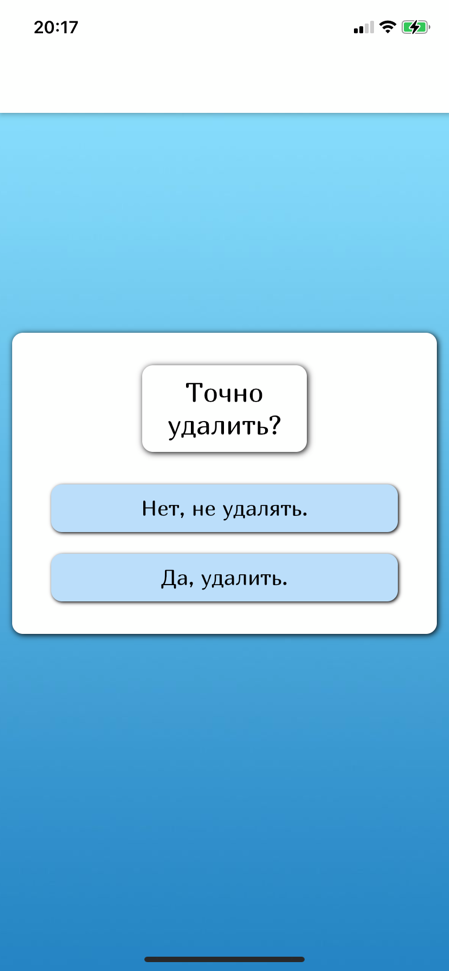
При согласии осуществляется DELETE - запрос (api/Transactions/${id}?sendNotification=$sendNotification, где id - уникальный идентификатор завершаемого обмена, email - адрес почты текущего пользователя, sendNotification = false).

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 204, операция прошла успешно, происходит возвращение на предыдущую страницу.

Если сервер вернул ответ с иным нежели 204 кодом, выводится сообщение об ошибке.

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Рисунок 50 - Страница исходящих обменов

Рисунок 52 - Страница редактирования обмена

Рисунок 51 - Страница удаления обмена

Описание процесса редактирования исходящего обмена

На странице (рис.52) отображены наименование и описание навыка, который был выбран для обмена, фотография, имя и фамилия его обладателя (при нажатии на фотографию/имя/фамилию можно перейти в профиль соответствующего пользователя (3.4.15)).

На странице присутствует поле типа DropdownButton для выбора навыка, который текущий пользователь предлагает взамен и текстовое поле для редактирования сообщения.

После нажатия кнопки "Сохранить изменения" проверяется не существует ли уже данного обмена.

Если такой обмен уже существует, выводится соответствующее сообщение.

Если такого обмена еще не существует, осуществляется PUT - запрос (api/Transactions/${id}, где id - уникальный идентификатор завершаемого обмена), в теле которого передается сериализованный в формат JSON экземпляр класса Exchange.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 200, происходит возвращение на предыдущую страницу.

Если сервер вернул ответ с иным нежели 200 кодом, выводится сообщение об ошибке.

Обзор страницы входящих обменов Изображение выглядит как текст

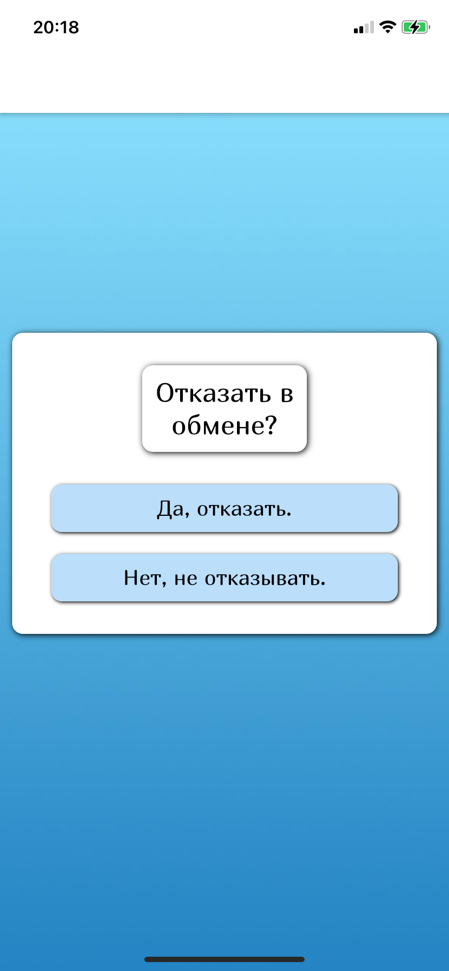
Автоматически созданное описание

Рисунок 53 - Страница входящих обменов

Рисунок 54 - Страница назначения обмена

Рисунок 55 - Страница отказа от обмена

На странице (рис.53) располагаются "карточки" входящих обменов, в которых содержится информация о том, кто предложил обмен (при нажатии на фотографию/имя/фамилию происходит переход на страницу соответствующего пользователя (3.4.15)), наименование получаемого в результате обмена навыка и наименование навыка, который предлагается "отдать" в обмен, кнопки согласия и отказа.

Описание процесса согласия на обмен

При нажатии на кнопку принятия обмена происходит переход на страницу назначения обмена (рис.54), на которой от пользователя требуется либо подтвердить назначение, либо отказаться от назначения.

При отказе происходит возврат на предыдущую страницу.

При согласии осуществляется PUT - запрос (api/Transactions/${email}, email - адрес почты текущего пользователя), в теле которого передается сериализованный в формат JSON экземпляр класса Exchange. С помощью данного запроса происходит изменение статуса обмена.

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 204, операция прошла успешно, происходит возвращение на предыдущую страницу.

Если сервер вернул ответ с иным нежели 204 кодом, выводится сообщение об ошибке.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 56 - Страница завершенных обменов

Описание процесса отказа в обмене

При нажатии на кнопку отказа от обмена происходит переход на страницу отказа от обмена (рис.55), на которой от пользователя требуется либо подтвердить отказ, либо отменить отказа.

При отмене происходит возврат на предыдущую страницу.

При согласии осуществляется DELETE - запрос (api/Transactions/${id}?sendNotification=$sendNotification, где id - уникальный идентификатор завершаемого обмена, sendNotification = true).

Если сервер не отвечает более чем 90 секунд - загрузка прерывается и выводится сообщение об ошибке.

Если сервер вернул ответ с кодом 204, операция прошла успешно, происходит возвращение на предыдущую страницу.

Если сервер вернул ответ с иным нежели 204 кодом, выводится сообщение об ошибке.

Обзор страницы завершенных обменов

На странице (рис. 56) располагаются "карточки" завершенных обменов, в которых содержится информация о том, с кем был произведен обмен (при нажатии на фотографию/имя/фамилию происходит переход на страницу соответствующего пользователя (3.4.15)), наименование полученного в результате обмена навыка и наименование навыка, который был "отдан" при обмен.

### Вывод о соответствии программы требованиям

На основе данных, приведенных выше (6.2.1 - 6.2.24), можно сделать следующие выводы:

* Программа полностью соответствует требованиям к составу выполняемых функций, а именно, корректно выполняет первичную проверку данных и взаимодействие с сервером, реализую весь заявленный в пункте 3.1.1 функционал.
* Требования к интерфейсу также соблюдены: реализован интерфейс для выполнения всех заявленных в пункте, приложение обладает понятным интерфейсом, дизайн системы однороден и выполнен в спокойных тонах
* Система корректно обрабатывает неверные запросы любого вида и выдает информативные сообщения. Требования к надёжности соблюдены.
* Требования к организации входных данных (входные данные приложения - различные действия пользователя и результаты HTTP-запросов) и требования об организации выходных данных (выходные данные приложения - HTTP-запросы и отображаемый на экране UI) соблюдены.
* Требования к временным характеристикам соблюдены (время ожидания ответа сервера не превышает 90 секунд)

# 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.

**Flutter –** SDK с открытым исходным кодом для создания мобильных приложений.

**JSON –** текстовый формат обмена данными.

**Widget(виджет) –** центральный элемент приложения на Flutter, визуальный компонент, из совокупности которых состоит графический интерфейс.

**State (состояние) –** это некоторая информация, которая может быть считана синхронно при создании виджета и которая в процессе жизненного цикла виджета может измениться.

**BuildContext –** это описание позиции виджета в дереве виджетов.

**Material Design –** стиль графического дизайна интерфейсов программного обеспечения и приложений, разработанный компанией Google.

**CupertinoDatePicker –** виджет для выбора даты в стиле iOS.

**DropdownButton –** кнопка material design для выбора из списка элементов.

**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 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
9. Устинов В. Надежность оптических дисков: как их правильно хранить и использовать. //Журнал «625» №7. М.: Издательство «625», 2005.
10. ГОСТ Р 7.02-2006 Консервация документов на компакт-дисках. Общие требования. – М.: ИПК Издательство стандартов, 2006.
11. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. 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).
13. 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).
14. 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).
15. Flutter. System requirements [Электронный ресурс]// URL https://flutter.dev/docs/get-started/install (Дата обращения: 20.02.2021, режим доступа: свободный).
16. Flutter. Material library [Электронный ресурс]// URL https://api.flutter.dev/flutter/material/material-library.html#classes (Дата обращения: 20.02.2021, режим доступа: свободный).
17. Flutter. Tutorials [Электронный ресурс]// URL https://flutter.dev/docs/reference/tutorials (Дата обращения: 20.02.2021, режим доступа: свободный).
18. Flutter. Tutorials [Электронный ресурс]// URL https://flutter.dev/docs/reference/tutorials (Дата обращения: 20.02.2021, режим доступа: свободный).
19. HTTP status codes [Электронный ресурс]// URL https://www.tutorialspoint.com/http/http\_sta-tus\_codes.htm (Дата обращения: 19.02.2021, режим доступа: свободный).
20. HTTP methods [Электронный ресурс]// URL https://www.tutorialspoint.com/http/http\_met-hods.htm (Дата обращения: 17.02.2021, режим доступа: свободный).
21. HTTP requests [Электронный ресурс]// URL https://www.tutorialspoint.com/http/http\_requests.htm (Дата обращения: 25.01.2021, режим доступа: свободный).
22. HTTP responses [Электронный ресурс]// URL https://www.tutorialspoint.com/http/http\_respo-nses.htm (Дата обращения: 20.02.2021, режим доступа: свободный).

# 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

# 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 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |