**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа бакалавриата «Программная инженерия»

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Научный руководитель,  приглашённый преподаватель ОП «Программная инженерия» департамента программной инженерии факультета компьютерных наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г.М. Сосновский  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. | УТВЕРЖДАЮ  Академический руководитель  ОП «Программная инженерия»  профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. Инв. № |  |
| Подп. и дата |  |
| Инв. № подл. | **RU.17701729.04.05-01 81 01-1-ЛУ** |

«**HSE APPLE**» **– ОБРАЗОВАТЕЛЬНАЯ СОЦИАЛЬНАЯ СЕТЬ**

**ДЛЯ НИСа ПО iOS РАЗРАБОТКЕ**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.05-01 81 01-1-ЛУ**

Исполнитель

**Изображение выглядит как текст

Автоматически созданное описание**студент группы БПИ204

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Туракулов И. У. /

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

УТВЕРЖДЕН

RU.17701729.06.08-01 81 01-1-ЛУ

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. Инв. № |  |
| Подп. и дата |  |
| Инв. № подл. | **RU.17701729.04.05-01 81 01-1-ЛУ** |

«**HSE APPLE**» **– ОБРАЗОВАТЕЛЬНАЯ СОЦИАЛЬНАЯ СЕТЬ**

**ДЛЯ НИСа ПО iOS РАЗРАБОТКЕ**

**Пояснительная записка**

**RU.17701729.04.05-01 81 01–1**

**Листов 54**

**АННОТАЦИЯ**

В данном программном документе приведена пояснительная записка к программе ««HSE Apple» – образовательная социальная сеть для НИСа по iOS разработке»

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:

1. постановка задачи на разработку программы;
2. описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;
3. описание и обоснование выбора метода организации входных и выходных данных;
4. 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 [1];
2. ГОСТ 19.102–77 Стадии разработки [2];
3. ГОСТ 19.103–77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104–78 Основные надписи [4];
5. ГОСТ 19.105–78 Общие требования к программным документам [5];
6. ГОСТ 19.106–78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.404–79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603–78 [8], ГОСТ 19.604–78 -[9].

**СОДЕРЖАНИЕ**

[**1. ВВЕДЕНИЕ** 5](#_Toc102769017)

[**2. НАЗНАЧЕНИЕ РАЗРАБОТКИ** 6](#_Toc102769018)

[**2.1. Функциональное назначение** 6](#_Toc102769019)

[**2.2. Эксплуатационное назначение** 7](#_Toc102769020)

[**3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ** 8](#_Toc102769021)

[**3.1. Постановка задачи на разработку программы** 8](#_Toc102769022)

[**3.2. Описание алгоритма и функционирования программы** 8](#_Toc102769023)

[**3.3. Описание и обоснование выбора состава технических и программных средств** 34](#_Toc102769024)

[**4. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ** 35](#_Toc102769025)

[**4.1. Ориентировочная экономическая эффективность** 35](#_Toc102769026)

[**4.2. Предполагаемая потребность** 35](#_Toc102769027)

[**4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами** 35](#_Toc102769028)

[**ПРИЛОЖЕНИЕ 2** 39](#_Toc102769029)

[**ПРИЛОЖЕНИЕ 3** 40](#_Toc102769030)

[**ПРИЛОЖЕНИЕ 4** 42](#_Toc102769031)

[**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ** 59](#_Toc102769032)

# **ВВЕДЕНИЕ**

* 1. **Наименование программы**

Наименование программы – ««HSE Apple» – образовательная социальная сеть для НИСа по iOS разработке».

Наименование программы на английском языке – ««HSE Apple» – educational social network for a research seminar on iOS development».

Наименование программы для пользователя – «HSE Apple».

* 1. **Краткая характеристика и область назначения**

««HSE Apple» образовательная социальная сеть для НИСа по iOS разработке» - мобильное приложение на Android, предназначающееся для курса НИСа среди студентов «НИУ ВШЭ».

# **НАЗНАЧЕНИЕ РАЗРАБОТКИ**

## **Функциональное назначение**

Функциональным назначением приложения «HSE Apple» предоставление возможности пользователям:

Обмен/Просмотр учебными материалами

Прохождение тестов

Просмотр событий и ведомости с оценками.

Выполнение заданий

В сервисе присутствует разделение на три роли пользователей:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип ресурса | Роль пользователя | | |
| Студент | Ассистент | Преподаватель |
| Чаты | чтение | создание  чтение  изменение  удаление | создание  чтение  изменение  удаление |
| Событие | чтение | чтение  изменение  удаление | создание  чтение  изменение  удаление |
| Ведомости успеваемости | чтение | чтение  изменение | создание  чтение  изменение  удаление |
| Права доступа заданиям | чтение | чтение  изменение | создание  чтение  изменение  удаление |
| Права доступа к системе приложения | - | чтение  изменение  удаление | создание  чтение  изменение  удаление |

## **Эксплуатационное назначение**

Данное приложение предназначено для студентов и преподавателей «НИУ ВШЭ», которые проходят тесты, выполняют лабораторные работы, общаются с пользователями, просматривают события и ведомость с оценками. Также предоставляются две локализованные версии: версия на Русском языке, версия на Английском языке для мобильных устройств на базе операционных систем Android.

# **ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**

## **Постановка задачи на разработку программы**

Основная задача разработать клиентское приложение для операционной системы Android.

* + 1. Задачи работы (Android клиент)

1. Реализовать авторизацию через ЕЛК;
2. Реализовать получение, хранение и обновление JWT (Javascript Web Tokens);
3. Разработать чат для пользователей курса;
4. Реализовать создание заданий для пользователей курса;
5. Реализовать создание новостей для участников с ролями “Преподаватель” и “Ассистент”
6. Реализовать взаимодействие с серверной частью через HTTP запросы;
7. Реализовать возможность ввода или изменения текстовых данных с валидацией;
8. Реализовать загрузку или изменение изображения через POST и PUT запросы;
9. Реализовать новостную ленту для пользователей курса;
10. Реализовать прохождение заданий для пользователей курса;
11. Реализовать выбор языковой локализации приложения;

## **Описание алгоритма и функционирования программы**

* + 1. Постановка задачи на разработку программы

#### **Структура мобильного приложения**

1. Загрузочная страница



Рисунок 1 Стартовая страница (Альбомная ориентация)



Рисунок 2 Стартовая страница (Портретная ориентация)

При запуске приложения, пользователю отображается загрузочная страница и в это время проверяется наличие refresh-токена, который был взят с локального хранилища устройства. Если токен присутствует пользователь автоматически переходит на страницу “Выбор курсов” иначе на страницу «Авторизация».

1. Авторизация

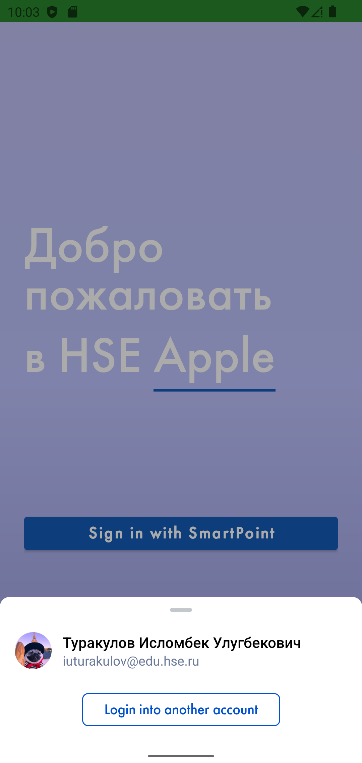


Рисунок 3 Авторизация

При переходе на страницу «Авторизация» Рис. 3 пользователю предлагается авторизация через ЕЛК (Smart Point). При нажатии на кнопку выводятся компоненты BottomSheet (<https://developer.android.com/reference/com/google/android/material/bottomsheet/BottomSheetBehavior>) (Рис. 3) панель для авторизации через аккаунт. Для авторизации используется уникальный client\_ID c redirect\_URI от HSE Digital и библиотеки для работы с ними: HSEAuth (<https://github.com/hseapp/HSEAuth-Android>) и HSECore (<https://github.com/hseapp/HSECore-Android>).

AuthHelper.login(activity, requestcode)

…..…

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
 super.onActivityResult(requestCode, resultCode, data)  
 if (requestCode == *REQUEST\_LOGIN*) {  
 if (resultCode != *RESULT\_OK* || data == null) {  
 return  
 }  
 *ACCESS\_TOKEN* = data.getStringExtra(AuthConstants.KEY\_ACCESS\_TOKEN)!!  
 *REFRESH\_TOKEN* = data.getStringExtra(AuthConstants.KEY\_REFRESH\_TOKEN)!!  
 Timber.d("Token got, success")  
 Toast.makeText(this, "Success", Toast.*LENGTH\_LONG*).show()  
 initializeToken()  
 }  
}

Листинг 1. Авторизация пользователя через ЕЛК.

При успешной авторизация приложение получает access и refresh токен, которые нужны для регистрации на сервер.

1. Выбор курса

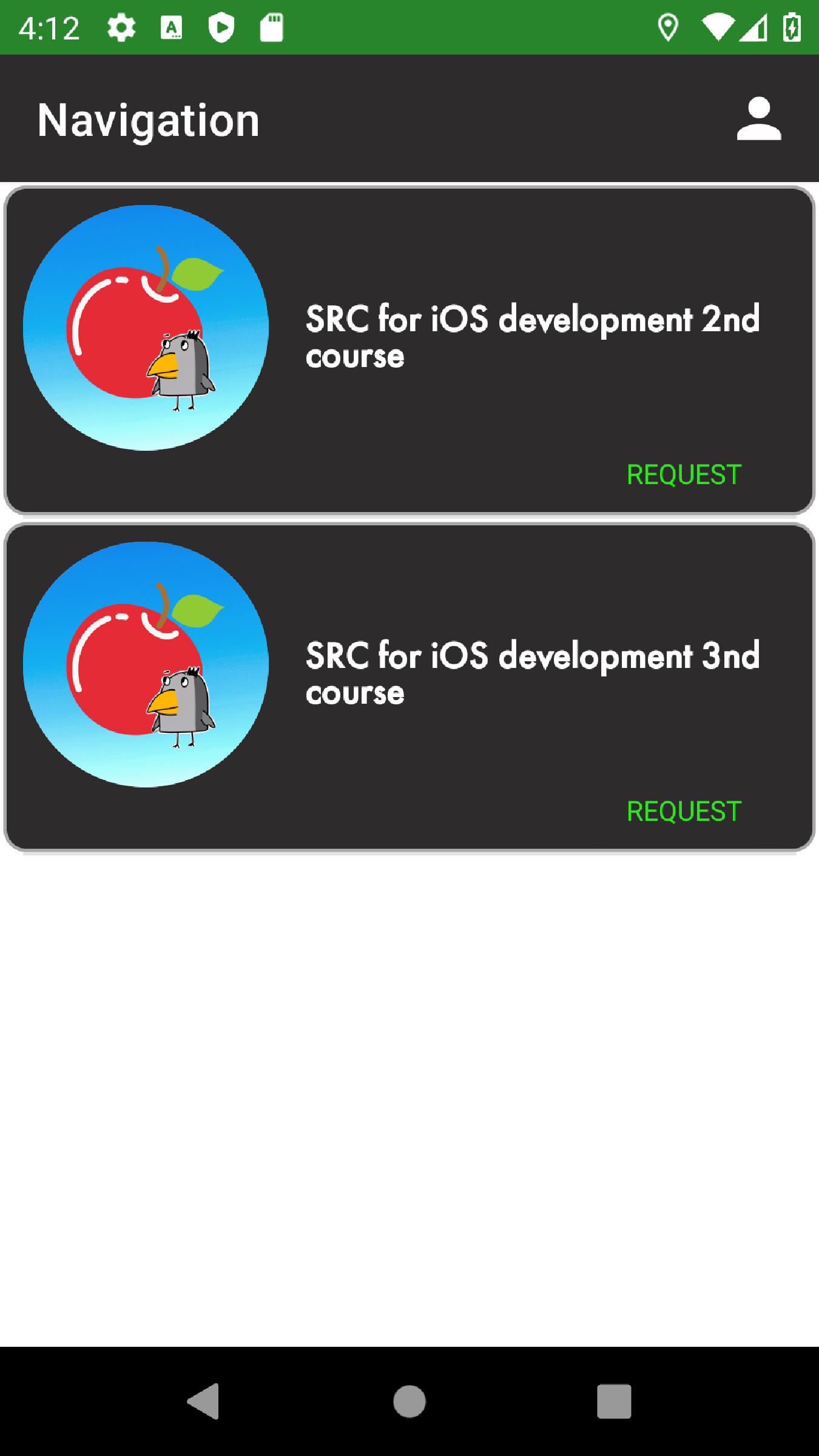


Рисунок 4 Выбор курса

При успешной авторизации пользователь переходит на страницу «Выбор курса» Рис. 4. При нажатии на кнопку “Request”, происходит подача заявки на курс или переход на страницу “Главная страница”. Пользователь ожидает подтверждения заявки от Преподавателя.

1. Главная страница

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 Чаты

В главной странице имеется меню выбора: Чаты, Тесты, Новости и Профиль. По-умолчанию выбирается страница «Чаты» Рис.5. В жизненном цикле фрагмента (<https://developer.android.com/guide/fragments/lifecycle>) была подключена библиотека CometChatUIKit (<https://github.com/cometchat-pro/android-kotlin-chat-ui-kit>) для корректной работы функционала.

Пользователь может ввести в поисковой панели название сообщения, группы или документа. Вводимые данные валидируются и показывается результат поиска.

При нажатии на кнопку, которая находится в правом верхнем углу, открывается новая страница просмотра всех доступных участников и групп.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 Выбор групп

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7 Просмотр участников

Страница содержит ViewPager (<https://developer.android.com/reference/androidx/viewpager/widget/ViewPager>) в котором находятся две страницы-фрагмента: «Просмотр участников» и «Выбор доступных групп». Загрузочной страницей по умолчанию является «Просмотр участников» Рис. 10. При вводе в поисковой панели текста, происходит запрос на получение похожего по значению слова.

val request =  
 UsersRequestBuilder().setSearchKeyword(s).setLimit(100).build()  
request.fetchNext(object : CallbackListener<List<User?>?>() {  
 override fun onSuccess(users: List<User?>?) = if (role.*isNotEmpty*()) {  
 val filteredItems = users?.*filter* **{ it**?.*role* != role **}** recyclerViewUserList!!.searchUserList(filteredItems)  
 } else {  
 recyclerViewUserList!!.searchUserList(users)  
 }  
  
 override fun onError(e: CometChatException) {  
 ErrorMsgUtils.showChatErrorMessage(*context*, e.*code*)  
 }  
})

Листинг 2. Авторизация пользователя через ЕЛК.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8 Групповой чат

Для отображения сообщений используется библиотека Shimmer Facebook RecyclerView (<https://github.com/facebook/shimmer-android>).

При долгом нажатии на кнопку, которая расположена в левом нижнем углу экрана появляется всплывающее меню со следующими возможностями:

1. Сфотографировать и отправить
2. Получить фото или видео с галереи
3. Отправить файл(-ы) или документ(-ы)
4. Поделиться геопозицией
5. Создать опросник

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 Отправка данных. BottomSheet (Альбомный режим)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 Отправка данных. BottomSheet (Портретный режим)

После нажатия на описанные кнопки Рис. 13-14 происходит действие между самой системой Android.

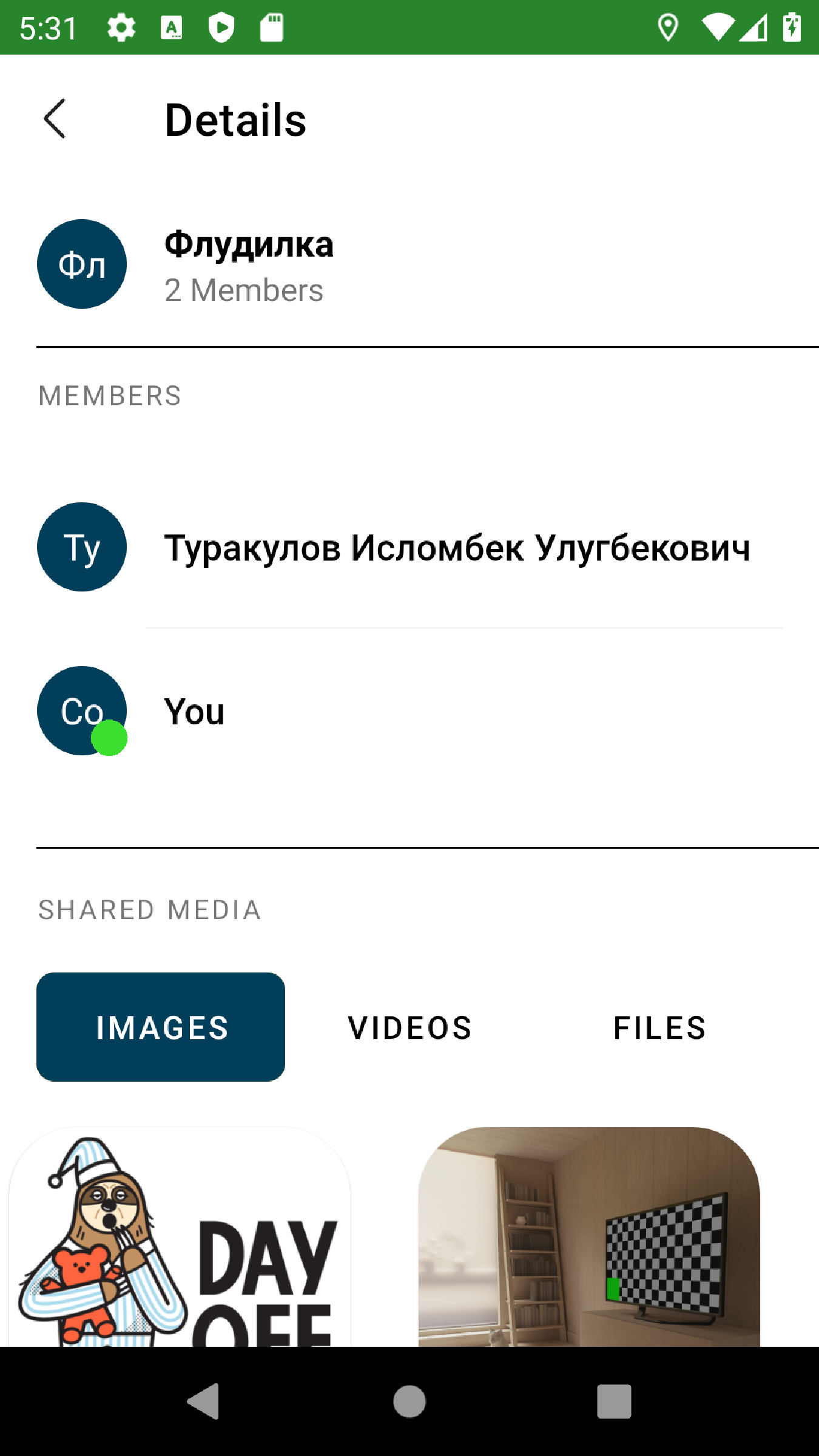


Рисунок 11 Информация о группе (вкладка изображение)

При нажатии на кнопку, которая расположена в правом верхнем углу (как информация) происходит переход на новую страницу с информацией о группе Рис. 11-12. В меню, которая расположена после информации об участников “Shared Media” меняется вкладка при нажатии на выбранную и соответственно при смене вкладок показываются разные данные.

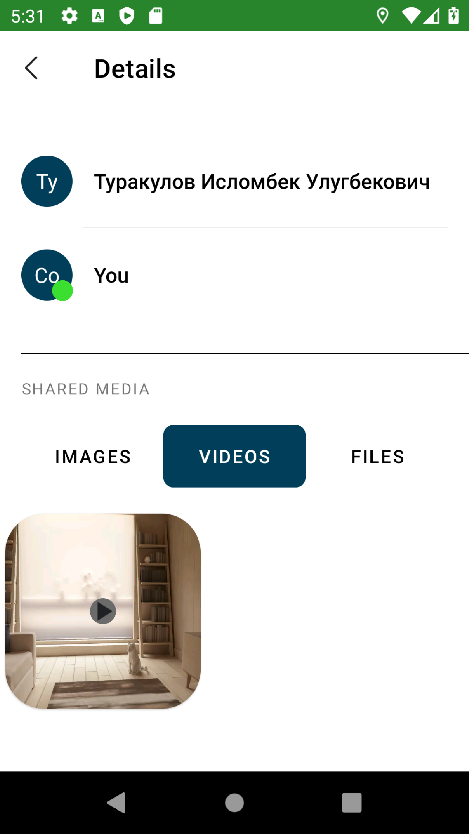


Рисунок 12 Информация о группе (вкладка видео)

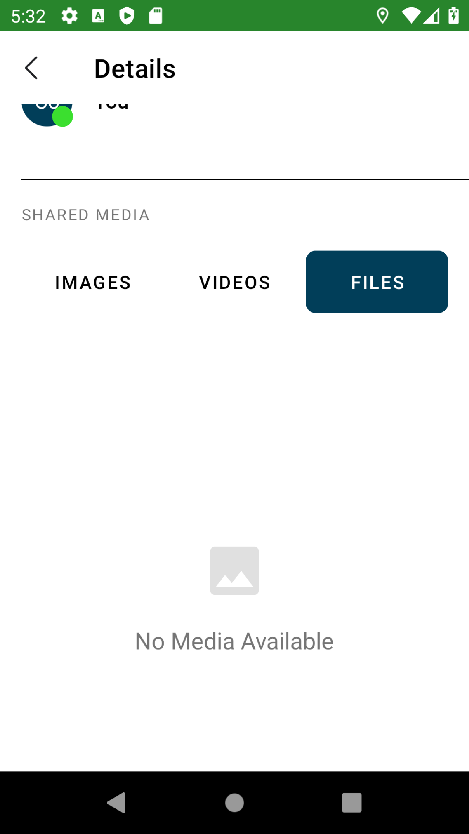


Рисунок 13 Информация о группе (вкладка файлы)

Кнопкой возвращения в чат, служит кнопка, которая расположена в левом верхнем углу.

Пользователь во время общения с другими участниками группы, может отправить эмодзи, гиф анимации и стикеры. Для этого используются официальная библиотека EmojiCompat от (<https://developer.android.com/jetpack/androidx/releases/emoji>).

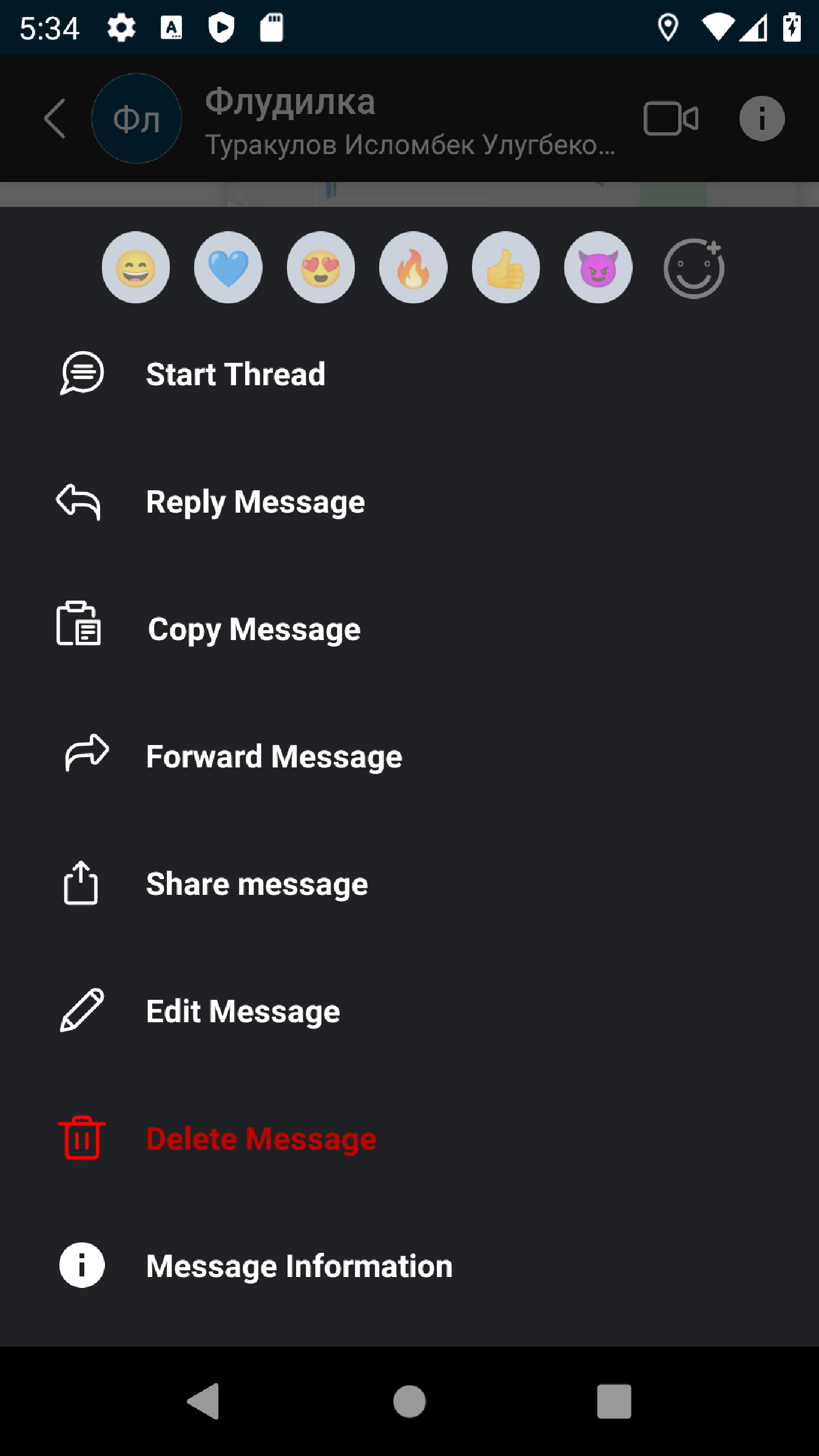


Рисунок 14 Операции над сообщением

При долгом нажатии на сообщение появляется всплывающее меню Рис. 16 выбора дополнительных действий:

1. Реакция на сообщение смайликами Рис 17.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 15 Реакция на сообщение

1. Начать дискуссию – при нажатии на данное действие, открывается страница для дополнительного ответа на сообщение, которое было нажато пользователем. Пользователь может поделиться сообщением, оставить реакцию на сообщение.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 16 Начать дискуссию

1. Ответить на сообщение – при нажатии на данное действие, пользователь может ответить на сообщение создавая цепочку ответов Рис. 17.

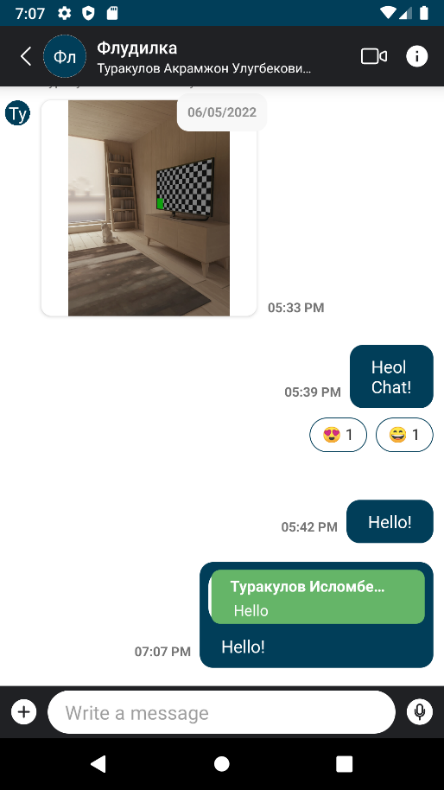


Рисунок 17 Ответить на сообщение

1. Скопировать сообщение – при нажатии на данное действие сообщение отправляется в буфер обмена.
2. Переслать сообщение – при нажатии на данное действие открывается страница выбора чатов Рис. 21

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 18 Переслать сообщение

1. Поделиться сообщением - при нажатии на данное действие открывается страница выбора отправки Рис. 19. При выборе приложения “HSE Apple”, показывается список чатов, описанное выше Рис. 18.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 19 Поделиться сообщением

1. Изменить сообщение - при нажатии на данное действие показывается маленькое окошка с подсказкой о том, что данный текст изменяется Рис. 18.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 20 Изменение текста

1. Удалить сообщение – при нажатии на данное действие сообщение удаляется из чата Рис. 19.

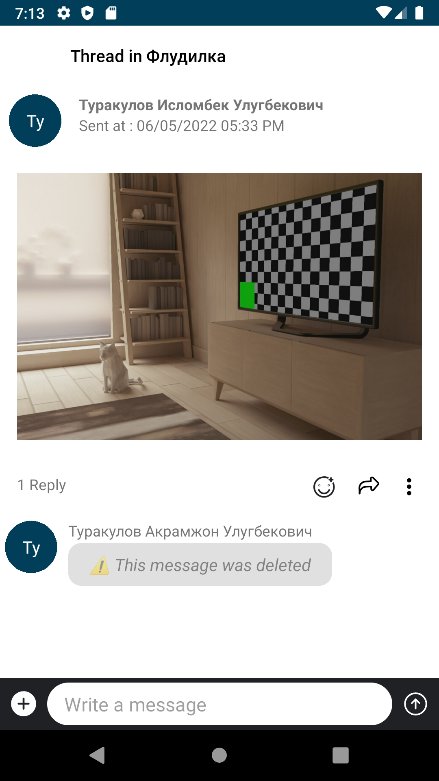


Рисунок 21 Сообщение удалено

1. Информация о сообщении - при нажатии на данное действие показывается информация отправителя сообщения и дата время отправки Рис. 20.

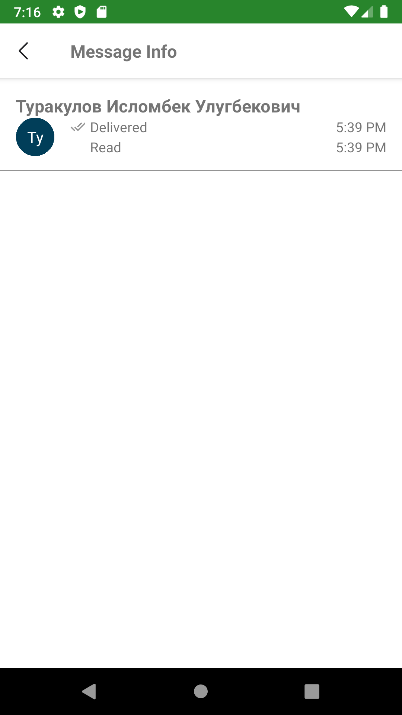


Рисунок 22 Информация о сообщении

При нажатии на кнопку “Send”, отправляется сообщение на все выбранные чаты Рис. 26.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 23 Пересланное сообщение

Кнопкой возвращения в главную страницу, является кнопка, которая расположена в левом верхнем углу экрана.

1. Страница «Тесты»

На данной странице показываются текущие и завершённые тесты пользователя. Если текущие задания отсутствуют, тогда показывается подсказка с изображением и текстом, аналогично с завершёнными заданиями. Отображение данных подсказок изображена на Рис. 24-25. Задания со статусом “Завершённое” показывают оценку за задание пользователя.

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание

Рисунок 24 Образец завершённых и текущих тестов

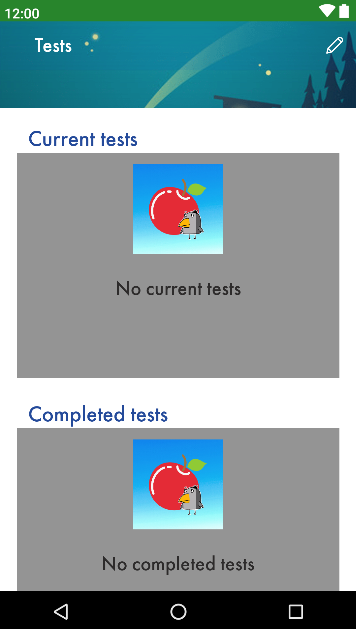


Рисунок 25 Отсутствуют тесты

Кнопка, которая расположена в правом верхнем углу доступна только пользователю с ролью “Преподаватель”. При нажатии на данную кнопку происходит переход на страницу «Создать задание» Рис. 26.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 26 Создание задания

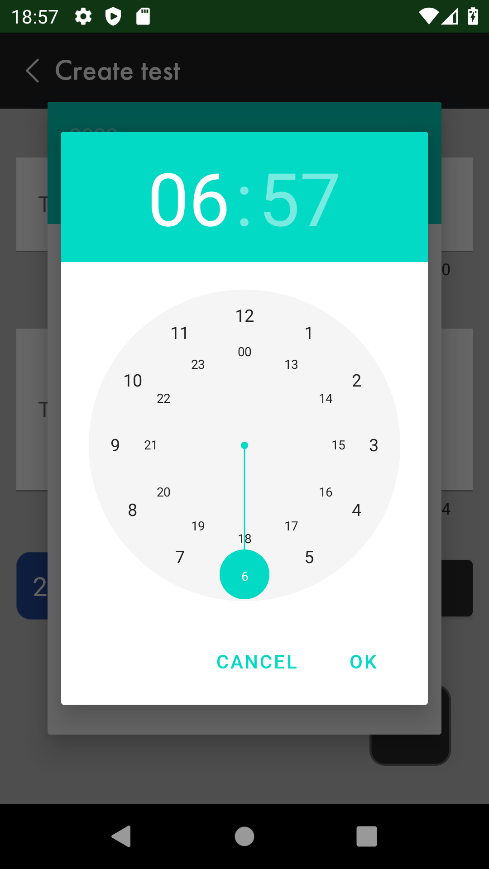


Рисунок 27 Выбор даты и времени дедлайна задания

При нажатии на кнопку “Change Deadline”, происходит системный вызов выбора даты и времени дедлайна для задания. Для корректной работы выбора используется библиотека MaterialDateTimePicker (<https://github.com/wdullaer/MaterialDateTimePicker>) Рис. 31.

Вводимые данные пользователем проверяются на корректность и дедлайн даты и времени.

fun TextInputEditText.validateDescription(): Boolean {  
 if (!this.*text*.*isNullOrEmpty*()) {  
 this.setText(this.*text*?.*trim*())  
 return this.*text*!!.length in 0..256  
 }  
 return false  
}  
  
fun TextInputEditText.validateTitle(): Boolean {  
 if (!this.*text*.*isNullOrEmpty*()) {  
 this.setText(this.*text*?.*trim*())  
 return this.*text*!!.length in 0..50  
 }  
 return false  
}

Листинг 3. Проверка данных на ввод.

Кнопкой сохранения данных является кнопка, которая расположена в правом нижнем углу. После нажатия на данную кнопку показывается Toast об информации отправки и закрывается страница.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 28 Информация о задаче (режим чтения)

При нажатии на тест, выбранный из списка доступных Рис. 24, происходит отправка объекта TaskEntity на страницу «Просмотр задания» Рис. 28.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 29 Информация о задаче (режим редактирования)

Кнопки, которые расположены в компоненте Toolbar (<https://developer.android.com/reference/android/widget/Toolbar>) , а именно функции “Изменение” и “Удаление” доступны для пользователей с ролью “Ассистент” и “Преподаватель” и во время просмотра теста после создания. При нажатии на кнопку “Изменение”, открывается доступ к редактированию текстовых полей и даты времени Рис. 29. При сохранении данных является кнопка, которая расположена в нижнем центре экрана.

Изображение выглядит как текст, снимок экрана, монитор, электроника

Автоматически созданное описание

Рисунок 30 Образец отображения события

При нажатии на данную кнопку выводится Toast с информацией об ошибке. При нажатии на кнопку “Удалить” происходит удаление поста с последующим закрытием страницы.

1. Страница «Новости»

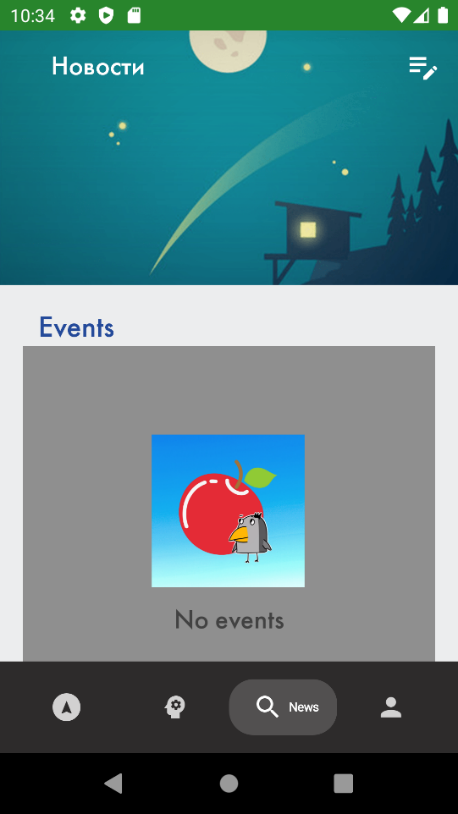


Рисунок 31 Отсутствуют новости

При открытии страницы показывается все доступные новости для определенного курса. При нажатии на новостной компонент со списка новостей, происходит открытие страницы «Информация о событии». Где показывается заголовок, описание и изображение новостного события Рис. 32.

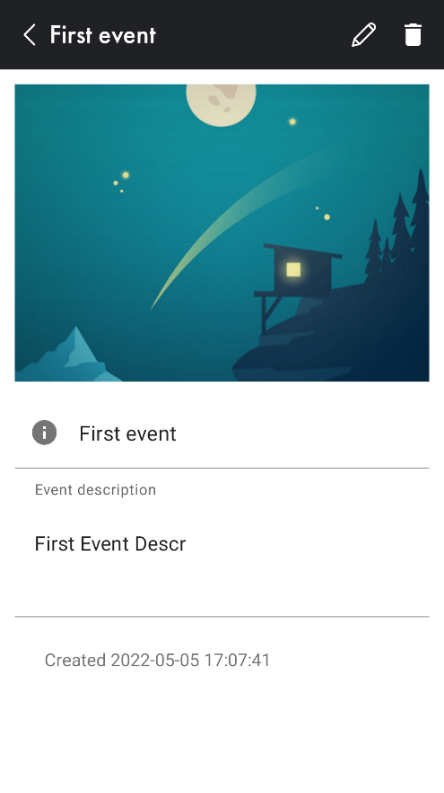


Рисунок 32 Информация новостного события

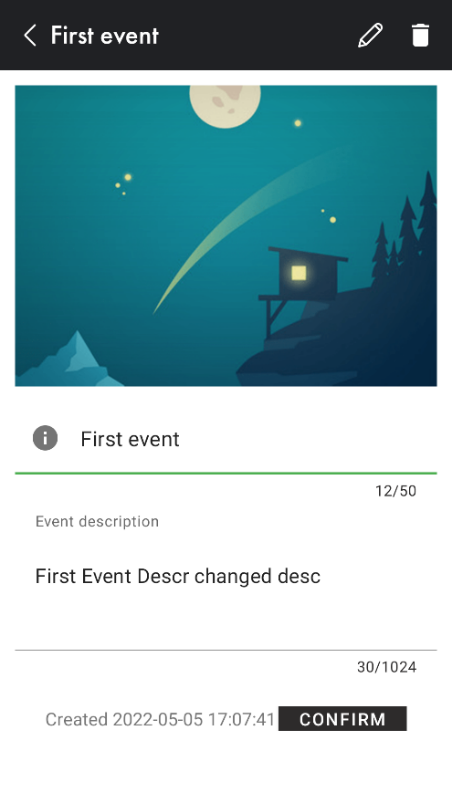


Рисунок 33 Редактирование события

Для пользователя с ролью “Ассистент” или “Преподаватель” доступны следующие функции:

1. Изменение - приложение даёт доступ к редактированию заголовка и описания новостного события Рис 33.
2. Удаление – показывается всплывающая панель с дополнительным выбором удаления. При подтверждении удаляется пост и закрывается страница Рис. 33.

private fun fieldOptionsInitializer(flag: Boolean)

editMode = flag  
newsTitleItem.isClickable = flag  
newsDescriptionItem.isClickable = flag  
newsTitleItem.isFocusable = flag  
newsDescriptionItem.isFocusable = flag  
newsTitleItem.isFocusableInTouchMode = flag  
newsDescriptionItem.isFocusableInTouchMode = flag  
if (!flag) {  
 newsTitleItem.tag = newsTitleItem.keyListener  
 newsDescriptionItem.tag = newsDescriptionItem.keyListener  
}  
textInputLayoutDesc.isCounterEnabled = flag  
textNewsTitleCreate.isCounterEnabled = flag  
confirmButtonEdit.visibility = if (flag) View.VISIBLE else View.GONE

Листинг 4. Проверка на роль пользователя

Кнопка, которая расположена в правом верхнем углу Рис. 32, видна только пользователю с ролью “Преподаватель”. При нажатии на данную кнопку происходит переход на новую страницу «Создать событие» Рис. 34.

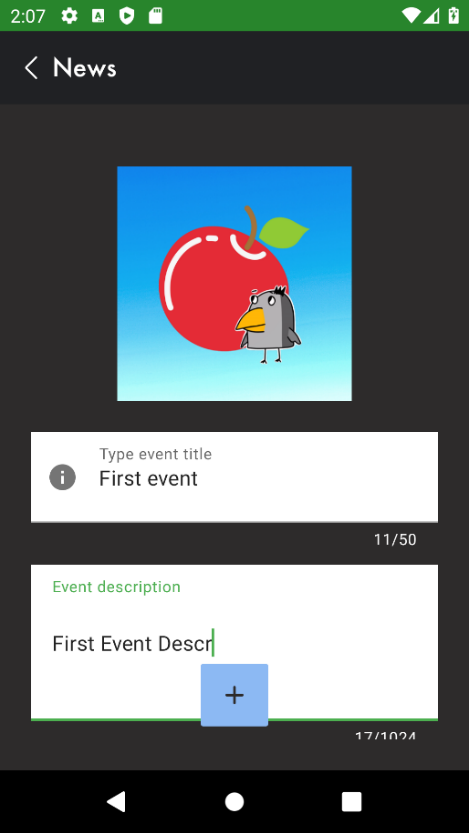


Рисунок 34 Создание события

Изображение выглядит как текст, снимок экрана, монитор

Автоматически созданное описание

Рисунок 35 Выбор изображения

При выборе изображения показывается DialogWindow с выбором изображения Рис. 35. Для данного выбора используется библиотека ImagePicker (<https://github.com/Dhaval2404/ImagePicker>). При вводе данных и сохранения на кнопку, которая расположена в нижнем центре экрана, проверяет вводимый текст Листинг 3. При нажатии на кнопку (плюсик) пост сохраняется и закрывается страница.

1. **Страница «Профиль»**

При переходе на данную страницу происходит инициализация профиля и выдача информации, которая была заранее взята во время авторизации пользователя.

Изображение выглядит как текст, электроника, снимок экрана

Автоматически созданное описание

Рисунок 36 Вид от студента/ассистента

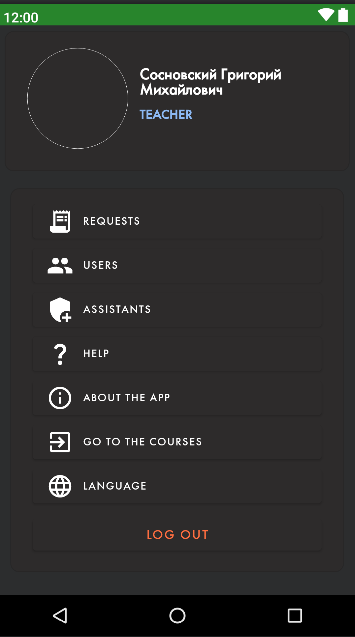


Рисунок 37 Вид от преподавателя

Для пользователя с ролью “Преподаватель” отображаются расширенные возможности приложения такие как: «Заявки», «Пользователи», «Ассистенты». Для пользователя с ролью “Студент” или “Ассистент”, скрываются расширенные настройки.

При нажатии на кнопку “Заявки”, показывается страница «Список заявок на курс», где списком отображаются доступные заявки на курс.

При действии swipeRight (<https://material.io/design/interaction/gestures.html>) происходит событие отклонения заявки на курс, пользователю высвечивается подсказка о том, что можно отменить данное действие, аналогично с действием swipeLeft(<https://material.io/design/interaction/gestures.html>) при выполнении данного события подтверждается заявка пользователя и для этого используется библиотека RecyclerViewSwipeDecorator (<https://github.com/xabaras/RecyclerViewSwipeDecorator>). Аналогично при нажатии на кнопку “Ассистенты”, показывается страница «Список ассистентов».

При нажатии на кнопку “Help”, происходит переход на почту тех. поддержки приложения, а именно в [letter.hseapple@gmail.com](mailto:letter.hseapple@gmail.com) .

При нажатии на кнопку “Go to the courses”, происходит переход на страницу «Выбор курса».

При нажатии на кнопку “Log out”, происходит переход на страницу «Авторизация» и зануляется Access-токен.

#### **Взаимодействие и обновление токенов**

Чтобы взаимодействовать с защищённым локальным хранилищем токенов, используется библиотека HSEAuth. Значения хранятся в формате ключ-значение, где в классе библиотеки AccountManager происходит краткое обновление refresh\_token и access\_token.

#### **Получение данных из локального хранилища**

Если приложение находится в оффлайн режиме, подгружаются все ранее скачанные данные. Для кодирования и декодирования изображений используется формате base64.

## **Описание и обоснование выбора состава технических и программных средств**

### **Состав технических и программных средств**

* Дисплей: Разрешение 1280 x 720
* Операционная система Android 9.0 или выше
* Оперативная память более 4 GB и портативная память более 3 GB

### **Обоснование выбора технических и программных средств**

Приложение разрабатывалось под оптимальную и рекомендованную версию операционной системы Android 9.0 «Pie» API 28. Так как, исходя из официального сайта компании Google API Levels (на 01.05.2022) процент пользователей, кто использует версию 9.0 и выше составляют 81.2%.

# **ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

## **Ориентировочная экономическая эффективность**

В рамках данной работы расчёт экономической эффективности не предусмотрен.

## **Предполагаемая потребность**

Данное мобильное приложение будет удобно совмещать чаты, тесты и новостную ленту в едином приложении. Предоставить единый доступ к материалам пользователям из-за этого преподавателю удобно проводить курсы НИСа в данном мобильном приложении.

## **Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами**

В рамках анализа конкурентов настоящего приложения были выделены следующие основные характеристики для сравнения:

1. Функциональность – разнообразие ключевых функций приложения.
2. Кроссплатформенность – портируемость приложения на разные платформы.
3. Удобство использования социальным работником – удобство взаимодействия с приложением целевой аудиторией пользователей.
4. Безопасность данных – защищенность персональных данных бенефициаров.
5. Пользовательский интерфейс (UI) – интуитивность пользовательского интерфейса.
6. Доступность – ценовая политика приложения.
7. Объем используемой базы – количество потенциально зарегистрированных бенефициаров.
8. Русская локализация

Указанные выше характеристики ранжированы. Это означает, что «Функциональность» имеет наивысший приоритет – 7, «Объем используемой базы» - наименьший, 1.

Процесс поиска аналогов в сети Интернет позволил выявить следующих прямых конкурентов:

* ВКонтакте (международное название — VK) — Российская социальная сеть [10]
* Campus Group - Платформа управления опытом кампуса "все в одном" [11]
* Society App – Приложение для управления событиями университета и организация задач. [12]
* LearningApps - платформа для поддержки обучения и преподавания с помощью небольших общедоступных интерактивных модулей. [13]
* Online Test Pad – платформа для обучения студентов и поддержки интерактивных упражнений. [14]
* ClassMarker – платформа для прохождения тестирований. [15]
* Quizziz – платформа для прохождения головоломок, тестов и задач и общение с преподавателями. [16]
* Coursera - проект в сфере массового онлайн-образования, основанный профессорами информатики Стэнфордского университета. [17]

Далее приведена таблица анализа указанных конкурентов по описанным характеристикам. По каждому критерию выставляется оценка по пятибалльной шкале.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Критерии / Название приложения | ВКонтакте [10] | Campus Group [11] | Society App [12] | LearningApps [13] | Online Test Pad [14] | ClassMarker [15] | Quizizz [16] | Coursera [17] | HSE Apple |
| Функциональность | 5 | 3 | 3 | 4 | 4 | 2 | 4 | 5 | 4 |
| Кроссплатформенность | 5 | 4 | 4 | 2 | 2 | 3 | 3 | 5 | 5 |
| Удобство использования | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 5 | 5 |
| Безопасность данных | 4 | 4 | 4 | 3 | 3 | 3 | 4 | 5 | 4 |
| UI | 5 | 4 | 3 | 4 | 4 | 2 | 4 | 5 | 5 |
| Доступность | 5 | 2 | 2 | 3 | 3 | 3 | 4 | 5 | 4 |
| Объем базы данных | 4 | 3 | 2 | 3 | 3 | 1 | 2 | 4 | 3 |
| Русская локализация | 5 | 0 | 0 | 0 | 3 | 2 | 0 | 5 | 5 |
| Оценка: | **4,63** | **3,00** | **2,75** | **2,87** | **3,25** | **2,50** | **3,00** | **4,88** | **4,38** |

**ПРИЛОЖЕНИЕ 1**

1. **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**
2. ГОСТ 19.101–77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.102–77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.103–77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.104–78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.105–78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.106–78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.404–79 Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.603–78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.604–78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. ВКонтакте [Электронный ресурс]. Режим доступа: <https://vk.com/> (Дата обращения 01.01.2022)
12. The all-in-one campus experience management platform // Campus Group [Электронный ресурс]. Режим доступа: <https://www.campusgroups.com/> (Дата обращения 01.01.2022).
13. Skyrocket your society // Society App [Электронный ресурс]. Режим доступа: <https://www.createyoursociety.com/> (Дата обращения 01.01.2022).
14. LearningApps [Электронный ресурс]. Режим доступа: <https://learningapps.org/>

(Дата обращения 01.01.2022)

1. Online test pads [Электронный ресурс]. Режим доступа: <https://onlinetestpad.com/> (Дата обращения 01.01.2022)
2. ClassMarker [Электронный ресурс]. Режим доступа: <https://www.classmarker.com/online-testing/faq/> (Дата обращения 01.01.2022)
3. Quizziz [Электронный ресурс]. Режим доступа:<https://quizizz.com>(Дата обращения 01.01.2022)
4. Coursera [Электронный ресурс]. Режим доступа: <https://www.coursera.org/> (Дата обращения 02.02.2022)
5. Системные требования Java [Электронный ресурс] // URL <https://www.java.com/ru/download/help/sysreq.html> (Дата обращения: 01.02.2022, режим доступа: свободный).
6. API levels and Android versions [Электронный ресурс]// URL: <https://apilevels.com/> (Дата обращения 1.05.2022, режим доступа свободный).
7. Wikipedia. JAR [Электронный ресурс]// URL: <https://ru.wikipedia.org/wiki/JAR> (Дата обращения 6.05.2022, режим доступа свободный)

# **ПРИЛОЖЕНИЕ** **2**

**ТЕРМИНОЛОГИЯ**

1. **Access token –** кратковременный токен для идентификации подлинности пользователя.
2. **Activity** – является отдельным экраном в Android.
3. **Android SDK -** универсальное средство разработки мобильных приложений для операционной системы Android.
4. **HMAC** (англ. *hash-based message authentication code* – «код аутентификации») – механизм проверки целостности информации.
5. **Fragment** - модульная часть Activity, у которой свой жизненный цикл и свои обработчики различных событий.
6. **Response** – ответ от сервера или объекта
7. **JAR** (англ. *Java Archive)*– это Java-архив. Представляет собой ZIP-архив, в котором содержится часть программы на языке Java.
8. **JSON (**англ. *JavaScript Object Notation*)— Текстовый формат обмена данными.
9. **JWT** (англ. *JSON Web Token*) — это открытый стандарт (RFC 7519) для создания токенов доступа, основанный на формате JSON.
10. **Refresh Token –** долговременный токен, используется для получения новой пары access/refresh токенов.
11. **Авторизация —** это проверка прав пользователя на доступ к определенным ресурсам.
12. **База данных** (сокр. БД) — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

**ПРИЛОЖЕНИЕ 3**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ**

Таблица 3.1

Описание и функциональное назначение классов мобильной части

| Класс | Назначение |
| --- | --- |
| ActivityUtitilities | Класс расширения для методов и полей |
| APIComet | Класс константы, который хранит ключ-значения сервера для общения между участниками |
| App | Класс, который предоставляет для инициализации авторизации |
| AppComponent | Класс, который предоставляет возможность инициализации между компонентами авторизации. |
| AssistantsAdapter | Класс адаптер для работы с инициализацией данных списка ассистентов |
| AvailableCoursesActivity | Класс Activity предоставляет возможность отображения данных о доступных курсах |
| ChatEntity | Класс модель чата для работы с чатом |
| ChatsFragment | Класс фрагмент для работы с отображением групповых чатов и списков |
| CompletedTestsAdapter | Класс адаптер для завершённых тестов для работы с инициализации данных списка тестов |
| ConstantEntities | Класс модель констант для инициализации данных |
| Constants | Класс констант |
| CourseEntity | Класс модель курса для работы с курсом |
| Courses | Класс модель курса для работы с курсом |
| CoursesAdapter | Класс адаптер курса для работы с инициализацией данных списка курсов |
| CreateGroupChatActivity | Класс Activity для группового чата для отображения создания чатов. |
| CreateNewsActivity | Класс Activity для работы с отображений создания новостей |
| CreateTestsActivity | Класс Activity для заданий |
| CurrentTestsAdapter | Класс адаптер для текущих тестов |
| ListOfAssistantsActivity | Класс Activity для отображений списка ассистентов |
| ListOfUsersActivity | Класс Activity для отображений списка участников |
| LoginAuthActivity | Класс Activity для авторизации через ЕЛК |
| MainActivity | Класс Activity для отображения меню страниц |
| MessageEntity | Класс модель сообщений |
| NewsAdapter | Класс адаптер для инициализации новостной ленты |
| NewsFragment | Класс Fragment для инициализации Layout компонент |
| NewsInfoActivity | Класс Activity для инициализации информации о новостях |
| PostEntity | Класс модель для событий |
| PreferenceManager | Класс Manager для контролирования данными |
| ProfileFragment | Класс Fragment для инициализации профиля пользователя |
| RequestEntity | Класс модель для заявок и дальнейшей работы с ним. |
| RequestsActivity | Класс Activity для инициализации списка заявок |
| RequestsAdapter | Класс адаптер для инициализации заявок курса |
| RoleOfUsers | Enum класс для обработки пользователей |
| SplashActivity | Класс Activity для отображения загрузочного экрана и инициализации предварительных данных |
| TaskEntity | Класс модель для выполнения/создания заданий |
| TaskInfoActivity | Класс Activity для отображения информаций заданий |
| TestsFragment | Класс Fragment для инициализации текущих и завершённых заданий |
| UserEntity | Класс модель для инициализации пользователей |
| UserTaskEntity | Класс модель для ответа заданий |

# **ПРИЛОЖЕНИЕ 4**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ И ФУНКЦИЙ**

Таблица 4.1

Описание методов и свойств класса ActivityUtilities.kt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| Activity.toast | internal | Расширение | message: CharSequence | | Метод для показа сообщения или подсказок |
| showToast | public | Метод | message: String | | Метод для показа сообщения в определенной странице |
| String.asDateTime | public | Расширение | Объект String | | Метод, который переводит дата и время в строку |
| TextInputEditText.validateTitle | public | Расширение | Объект TextInputEditText | | Метод для проверки ввода на корректность |
| TextInputEditText.validateDescription | public | Расширение | Объект TextInputEditText | | Метод для проверки ввода на корректность |
| TextInputEditText.validateNews | public | Расширение | Объект TextInputEditText | | Метод для проверки ввода на корректность |
| Calendar.formatDateTime | internal | Расширение | Объект Calendar | | Метод, который переводит дата и время в строку |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| APP\_ACTIVITY | lateinit public | Activity | get, set | Назначение главного Activity | |
| toast | private | Toast | get, set | Объект для работы с Toast информациями. | |

Таблица 4.2

Описание методов и свойств класса APIComet.kt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| Отсуствуют |  |  |  | |  |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| APP\_ID | public const | String | get | Уникальный идентификатор приложения | |
| API\_KEY | public const | String | get | Уникальный идентификатор API | |
| AUTH\_KEY | public const | String | get | Уникальный идентификатор авторизации | |
| API\_COUNTRY | public const | String | get | Уникальный идентификатор страны | |

Таблица 4.3

Описание методов и свойств класса App.kt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| cometInitialization | private | Метод |  | | Метод для выставления настроек для пользователя. |
| createNotificationChannel | private | Метод |  | | Метод для инициализации уведомления |
| provideAuthComponent | override | Метод |  | | Метод для инициализации компонент авторизации |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| TAG | private | String | get | Название класса | |

Таблица 4.4

Описание методов и свойств интерфейса AppComponent.kt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| inject | public | Метод | app: App или activity: LoginAuthActivity | | Провести вставку Activity в компоненту |
| loginComponent | public | Метод | Отсутствуют | | Получить компоненту авторизации |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Builder | public | Interface | get | Builder | |

Таблица 4.5

Описание методов и свойств класса AssistantsAdapter.kt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| reloadItems | private | Конструктор |  | | Метод для GET-запроса для получения данных об ассистентах и инициализации |
| onCreateViewHolder | private | Метод | parent: ViewGroup, viewType: Int | | Метод для инициализации компонента для ViewHolder |
| onBindViewHolder | private | Метод | holder: ViewHolder, position: Int | | Метод для закрепления объектов в ViewHolder |
| getItemCount | public | Метод |  | | Метод для получения количества объектов в списке |
| addItem | public | Метод | item: User, position: Int | | Метод для добавления User объекта в список |
| removeItem | public | Метод | position: Int | | Метод для удаления User объекта из списка |
| getAllItems | public | Метод |  | | Метод для получения списка из User объектов |
| init | public | Конструктор | context: Context | | Инициализация и вызов методов с контекстом |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| context | private | Context | get, set | Контекст. | |
| mItems | private | ArrayList<User> | get, set | Список для взаимодействия с объектами | |

Таблица 4.6

Описание методов и свойств класса AvailableCoursesActivity.kt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| updateItems | private | Метод | list: ArrayList<Courses> | | Метод для инициализации компонент для отображения |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Отсутствуют |  |  |  |  | |

Таблица 4.7

Описание методов и свойств класса СhatEntity.kt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| ChatEntity | public | Конструктор | val id: Long? = null, var title: String? = null, var description: String? = null, var courseID: Long? = null, var updatedBy: Long? = null, var group\_avatar: String? = null, var createdAt: Timestamp? = null, var updatedAt: Timestamp? = null | | Инициализация объектов |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| id | private | Long | get, set | Уникальный идентификатор чата | |
| title | private | String? | get, set | Заголовок чата | |
| description | private | String? | get, set | Описание чата | |
| courseId | private | Long? | get, set | Уникальный идентификатор курса | |
| updatedBy | private | Long? | get, set | Кем был обновлен | |
| group\_avatar | private | String? | get, set | Фото чата | |
| createdAt | private | LocalDateTime? | get, set | Дата и время создания | |
| updatedAt | private | LocalDateTime? | get, set | Дата и время обновления | |

Таблица 4.8

Описание методов и свойств класса ChatsFragment.kt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| initializeListeners | private | Метод |  | | Метод для инициализации и объявления чата |
| loadFragment | private | Метод | fragment: Fragment | | Метод для загрузки новой страницы-фрагмента |
| startGroupIntent | private | Метод | group: Group? | | Метод для запуска новой страницы с чатом |
| joinGroup | private | Метод | group: Group? | | Метод для принятия подтверждения в группу |
| onButtonClick | private | Метод | alertDialog: AlertDialog? v: View?, which: Int, popudId: Int | | Метод для инициализации события нажатия кнопки |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| progressDialog | private | ProgressDialog | get, set | Окно с информацией о загрузке | |
| groupPassword | private | String? | get, set | Вводимый пароль группы | |
| group | private | Group? | get, set | Объект группа | |

Таблица 4.9

Описание методов и свойств класса CompletedTestsAdapter.kt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| reloadItems | private | Метод |  | | Метод для инициализации уже раннее объявленного списка с CurrentTestAdapter |
| onCreateViewHolder | private | Метод | parent: ViewGroup, viewType: Int | | Метод для инициализации компонента для ViewHolder |
| onBindViewHolder | private | Метод | holder: ViewHolder, position: Int | | Метод для закрепления объектов в ViewHolder |
| getItemCount | public | Метод |  | | Метод для получения количества объектов в списке |
| getAllItems | public | Метод |  | | Метод для получения списка из User объектов |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| completedTestsList | private | ArrayList<TaskEntity> | get, set | Список из TaskEntity. | |

Таблица 4.10

Описание методов и свойств класса ConstantEntities.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| arrayOfRequestCourses | public | ArrayList<RequestEntity> | get, set | Список из заявок |
| allTests | public | ArrayList<TaskEntity> | get, set | Список из заданий |
| USER | public | UserEntity | get, set | Класс модель пользователя |
| postInfo | public | PostEntity | get, set | Класс модель новостного события |
| taskInfo | public | TaskEntity | get, set | Класс модель задания |

Таблица 4.11

Описание методов и свойств кдасса Constants.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| preferences | public | SharedPreferences | get, set | Управление локальными данными пользователя |
| ACCESS\_TOKEN | public | String | get, set | Access токен пользователя после авторизации и дальнейшей работы с приложением |
| REFRESH\_TOKEN | public | String | get, set | Refresh токен для контролирования обновления пользователя |
| USER\_CHAT | public | User | get, set | Объект – пользователь для работы с чатом |
| IP\_ADDRESS | public | String | get | Адрес сервера для работы с запросами |
| REQUEST\_LOGIN | public | Long | get | Код запроса авторизации |
| TEMP\_TOKEN | public | String | get | Временный токен для показа примеров работы приложения |
| SELECTION | public | CourseSelection | get, set | Для определения выбранного курса |
| CourseSelection | public | Enum класс | get, set | Enum класс для работы с выбором курса |

Таблица 4.12

Описание методов и свойств интерфейса CourseEntity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| id | public | Long? | get, set | Уникальный идентификатор курса |
| title | public | String? | get, set | Заголовок курса |
| description | public | String? | get, set | Описание курса |

Таблица 4.13

Описание методов и свойств интерфейса Courses.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| nameOfCourse | public | String | get, set | Название курса |
| image | public | String? | get, set | Изображение курса |

Таблица 4.14

Описание методов и свойств интерфейса CoursesAdapter.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| CoursesAdapter | public | Конструктор | courses: ArrayList<Courses> | Список курсов |
| updateData | public | Метод | courses: ArrayList<Courses> | Обновление курса |
| initializePreRequests | private | Метод | position: Courses, holder: View | Инициализация заявок |
| initializeButtons | private | Метод | courseId: Int | Инициализация посредством GET-запроса |

Таблица 4.15

Описание методов и свойств интерфейса CreateNewsActivity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| setListeners | public | Метод |  | Инициализация компонент Layout |
| createEvent | private | Метод |  | Отправка POST-запроса на создание новостного события |
| encodeUri | private | Метод | imageUri: Uri | Кодирование ссылки в Base64-строку |
| checkStoragePermissions | private | Метод |  | Проверка доступа к хранилищу |
| requestStoragePermissions | private | Метод |  | Получение доступа к хранилищу |
| checkCamerPermissions | private | Метод |  | Проверка доступа к камере |
| requestCamerPermissions | private | Метод |  | Получение доступа к камере |
| showImagePickerDialog | private | Метод |  | Показ диалогового окна выбора изображения |
| validateInputFields | private | Метод |  | Проверка вводимых данных |

Таблица 4.16

Описание методов и свойств интерфейса CreateTestsActivity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| setListeners | private | Метод |  | Инициализация компонент Layout |
| createTest | private | Метод |  | Отправка POST-запроса на создание задания |
| onTimeSet | private | Метод |  | Переопределение выбранного времени |
| onDataSet | private | Метод |  | Переопределение выбранной даты |

Таблица 4.17

Описание методов и свойств класса CurrentTestsAdapter.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Метод** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| reloadItems | private | Метод |  | Метод для инициализации уже раннее объявленного списка с CurrentTestAdapter |
| onCreateViewHolder | private | Метод | parent: ViewGroup, viewType: Int | Метод для инициализации компонента для ViewHolder |
| onBindViewHolder | private | Метод | holder: ViewHolder, position: Int | Метод для закрепления объектов в ViewHolder |
| getItemCount | public | Метод |  | Метод для получения количества объектов в списке |
| getAllItems | public | Метод |  | Метод для получения списка из TaskEntity объектов |
| **Свойства** |  |  |  |  |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| completedTestsList | private | ArrayList<TaskEntity> | get, set | Список из TaskEntity. |

Таблица 4.18

Описание методов и свойств класса ListOfAssitantsActivity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| initializeCallBacks | private | Метод |  | Инициализация логики добавления и удаления ассистента со списка |

Таблица 4.19

Описание методов и свойств класса ListOfUsersActivity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| loadFragment | private | Метод | fragment: Fragment? | Открытие и переход на новый фрагмент |

Таблица 4.20

Описание методов и свойств класса LoginAuthActivity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| authLoginCredentials | private | Метод |  | Получение или регистрация пользователя в приложение |
| createUser | private | Метод |  | Создание пользователя для чата |
| login | private | Метод | user: User | Авторизация в приложение |

Таблица 4.21

Описание методов и свойств класса MainActivity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| setupSmoothBottomMenu | private | Метод |  | Инициализация нижнего меню |

Таблица 4.22

Описание методов и свойств класса NewsAdapter.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| reloadItems | private | Метод |  | Метод для инициализации списка |
| onCreateViewHolder | private | Метод | parent: ViewGroup, viewType: Int | Метод для инициализации компонента для ViewHolder |
| onBindViewHoder | private | Метод | holder: ViewHolder, position: Int | Метод для закрепления объектов в ViewHolder |
| getItemCount | public | Метод |  | Метод для получения количества объектов в списке |
| getAllItems | public | Метод |  | Метод для получения списка из PostEntity объектов |
| **Свойства** |  |  |  |  |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| news | private | ArrayList<PostEntity> | get, set | Работа с объектами PostEntity для отображения и инициализации |

Таблица 4.22

Описание методов и свойств класса NewsFragment.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| initializeNewsAdapter | private | Метод |  | Инициализация новостной ленты |
| initializeGradesAdapter | private | Метод |  | Инициализация оценок за тесты |

Таблица 4.23

Описание методов и свойств класса NewsInfoActivity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| initializeTopAppBar | private | Метод | - | Обновление и инициализация верхнего меню |
| initializeFields | private | Метод | - | Инициализация полей |
| editDataNews | private | Метод | - | Запрос на изменение данных новости |
| validateInputFields | private | Метод | - | Метод для проверки вводимых данных |
| fieldOptionInitializer | private | Метод | - | Метод для предоставления характеристик компонент класса |

Таблица 4.24

Описание методов и свойств класса PostEntity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| id | public | Long? | get, set | Уникальный идентификатор новостного события |
| courseId | public | Long? | get, set | Уникальный идентификатор курса |
| createdBy | public | Long? | get, set | Кем был создан |
| updatedBy | public | Long? | get, set | Кем был обновлен |
| title | public | String? | get, set | Заголовок новостного события |
| content | public | String? | get, set | Описание новостного события |
| mediaLink | public | String? | get, set | Ссылка на документ |
| docLink | public | String? | get, set | Ссылка на документ |
| createdAt | public | String? | get, set | Дата и время создания |
| updatedAt | public | String? | get, set | Дата и время изменения |

Таблица 4.25

Описание методов и свойств класса ProfileFragment.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| initializeDefaultFeatures | public | Метод | - | Инициализация для пользователей с ролью “Студент” и “Ассистент’ |
| initializeTeacherFeatures | public | Метод | - | Инициализация для пользователей с ролью “Преподаватель” |

Таблица 4.26

Описание методов и свойств класса RequestEntity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| id | public | Long | get, set | Уникальный идентификатор заявки |
| userId | public | Long | get, set | Уникальный идентификатор пользователя |
| courseId | public | Long | get, set | Уникальный идентификатор курса |
| roleId | public | Long | get, set | Роль пользователя |
| approved | public | Boolean | get, set | Принята ли заявка или нет |
| createdAt | public | LocalDateTime | get, set | Дата и время создания |
| updatedAt | public | LocalDateTime | get, set | Дата и время изменения |

Таблица 4.27

Описание методов и свойств класса RequestsActivity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| initializeCallBacks | private | Метод | - | Инициализация логики добавления и удаления ассистента со списка |

Таблица 4.28

Описание методов и свойств класса RequestsAdapter.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| reloadItems | private | Метод | - | Метод для инициализации списка |
| onCreateViewHolder | private | Метод | parent: ViewGroup, viewType: Int | Метод для инициализации компонента для ViewHolder |
| onBindViewHoder | private | Метод | holder: ViewHolder, position: Int | Метод для закрепления объектов в ViewHolder |
| getItemCount | public | Метод | - | Метод для получения количества объектов в списке |
| getAllItems | public | Метод | - | Метод для получения списка из RequestEntity объектов |
| addItem | public | Метод | item: RequestEntity, position: Int | Добавление объекта в список |
| removeItem | public | Метод | position: Int | Удаление объекта из списка |
| **Свойства** |  |  |  |  |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| news | private | ArrayList<PostEntity> | get, set | Работа с объектами RequestEntity для отображения и инициализации |

Таблица 4.29

Описание методов и свойств Enum-класса RoleOfUsers.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| STUDENT |  | Int | get | Роль пользователя |
| ASSISTANT |  | Int | get | Роль пользователя |
| TEACHER |  | Int | get | Роль пользователя |

Таблица 4.30

Описание методов и свойств класса SplashActivity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| isLoggedIn | private | Метод | - | Проверка, проходил ли пользователь авторизацию ранее |

Таблица 4.31

Описание методов и свойств класса TaskEntity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| id | public | Long? | get, set | Уникальный идентификатор задания |
| courseId | public | Long? | get, set | Уникальный идентификатор курса |
| form | public | String? | get, set | Тип задания “lab” или “test” |
| title | public | String? | get, set | Заголовок задания |
| description | public | String? | get, set | Описание задания |
| task\_content | public | String? | get, set | Информация о задании |
| deadline | public | String? | get, set | Дедлайн задания |
| status | public | Boolean? | get, set | Статус задания “Завершенное” или “Текущее” |
| createdBy | public | Long? | get, set | Кем был создан |
| updatedBy | public | Long? | get, set | Кем был изменен |
| createdAt | public | String | get, set | Дата и время создания |
| updatedAt | public | String? | get, set | Дата и время изменения |

Таблица 4.32

Описание методов и свойств класса TaskInfoActivity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| setListeners | private | Метод | - | Инициализация компонент Layout. |
| validateInputFields | private | Метод | - | Проверка на корректность данных |
| initializeTopAppBar | private | Метод | - | Инициализация верхнего меню |
| fieldOptionsInitializer | private | Метод | - | Инициализация характеристик для компонент |
| createTest | private | Метод | - | Метод для отправки POST-запроса на создание задания |

Таблица 4.33

Описание методов и свойств класса TestsFragment.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| initializeCompletedTests | private | Метод | - | Инициализация текущих тестов |
| initializeCurrentTests | private | Метод | - | Инициализация завершённых тестов |

Таблица 4.34

Описание методов и свойств класса UserEntity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| id | public | Long? | get, set | Уникальный идентификатор пользователя |
| fullName | public | String? | get, set | ФИО пользователя |
| email | public | String? | get, set | Почта пользователя |
| createdAt | public | LocalDateTime | get, set | Дата и время создания аккаунта |

Таблица 4.35

Описание методов и свойств класса UserTaskEntity.kt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| id | public | Long? | get, set | Уникальный идентификатор задания пользователя |
| answer | public | String? | get, set | Ответ пользователя |
| score | public | Int? | get, set | Оценка пользователя |
| status | public | Boolean? | get, set | Статус задания “Завершенный” или “Текущий” |
| taskID | public | Long? | get, set | Уникальный идентификатор задания |
| createdBy | public | Long? | get, set | Кем был создан |
| updatedBy | public | Long? | get, set | Кем был обновлен |
| createdAt | public | LocalDateTime? | get, set | Дата и время создания ответа |
| updatedAt | public | LocalDateTime? | get, set | Дата и время изменения ответа |

# **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 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |