**Постановка задачи**

**1. Организационно-экономическая сущность задачи**

Наименование задачи: мобильное приложение «SkillThread»;

Цель разработки: расширение возможностей обучения учащихся учебных заведений;

Назначение: данный программный продукт разрабатывается для студентов и преподавателей, желающим использовать конструктор текстовых материалов, задач и тестов для проверки и обучения учащихся;

Периодичность использования: во время обучения или по мере необходимости;

Источники и способы получения данных: документации Python, Kivy, Flask;

Обзор существующих аналогичных ПП: Рассмотрим приложение Stepik. В этом приложении реализованы проверка тестов, возможность создания теоритических блоков и проверки написанного кода программ. Все эти основные функции так же будут присутствовать в новом ПП, но будут исключена возможность проверки кода.

Ещё одним аналогичным ПП является GoogleClass, в котором реализована отправка заданий. Эта функция так же будет присутствовать в новом ПП, но получаемые задания будут зависеть от уровня учащихся, заранее установленным преподавателем.

**2. Функциональные требования**

Описание перечня функций и задач, которые должен выполнять будущий ПП:

Гость:

1. авторизация
2. смена темы приложения
3. просмотр информации об авторе

Пользователь: все те же функции, что может выполнять гость, но также появляется:

1. деавторизация
2. просмотр списка учебных каналов, участником которых он является
3. просмотр информации учебных каналов
4. просмотр списка публикаций учебных каналов

Участник учебного канала: все те же функции, что может выполнять пользователь, но также появляется:

1. просмотр публикаций
2. просмотр и выполнение заданий публикаций

Преподаватель: все те же функции, что может выполнять пользователь, но также появляется:

1. установка уровня учащихся в их профиле учебного канала
2. создание учебных каналов
3. создание/удаление публикаций
4. комментирование публикаций
5. создание и отправка заданий

Учащийся: все те же функции, что может выполнять участник учебного канала, но также появляется:

1. просмотр публикаций
2. прохождение заданий публикаций
3. просмотр своего уровня
4. комментирование публикаций

**3. Описание процессов с входной, выходной и условно-постоянной информацией:**

Гость:

При регистрации в окне авторизации, будет создан новый аккаунт учащегося с настроенным профилем в соответствии с желанием ностя. Входная информация – данные нового пользователя (логин, пароль, данные профиля), выходная информация – переход на домашний экран, условно-постоянная информация – данные о созданных аккаунтах

При входе в аккаунт входной информацией будет логин и пароль аккаунта, выходной – переход на домашний экран, условно-постоянная информация – данные аккаунтов.

При нажатии на кнопку смены приложения, оформление изменяется со светлой на темную и наоборот. Входная информация – текущая тема приложения, выходная информация – смена темы приложения, условно-постоянная информация – разметка оформления тем.

При нажатии на кнопку просмотра информации об авторе, пользователь попадает на экран с данными об авторе и его контактной информацией. Входная информация – отсутствует, выходная информация – данные об авторе и его контактной информации, условно-постоянная информация – отсутствует.

Участник учебного канала:

При нажатии на кнопку учебного канала на домашнем экране, пользователь перейдет на экран канала – списка публикаций. Входная информация – выбор учебного канала на домашней странице, выходная информация – отображение списка публикаций выбранного учебного канала, условно-постоянная информация – информация о публикациях в выбранном учебном канале

При нажатии на кнопку с иконкой «Комментарий», на экране пользователя появится поле ввода текста комментария и откроется экранная клавиатура. Входная информация – текст комментария, выходная информация – обновленный список комментариев публикации, условно-постоянная – данные о списке комментариев публикации.

При нажатии на иконку учебного канала на его экране, произведется переход на экран просмотра информации учебного канала. Входная информация – ID канала, выходная информация – переход на экран с информацией выбранного учебного канал, условно-постоянная информация – информация о выбранном учебном канале.

Преподаватель:

При установке уровня учащихся в их профиле учебного канала: входная информация – ID учебного канала и учащегося, выходная информация – обновленный уровень учащегося в виде, условно-постоянная – данные учащихся учебных каналов.

При нажатии на кнопку создания учебного канала, преподавателю будет предложено ввести данные нового учебного канала (иконка, название, описание, список учащихся, которых необходимо добавить в канал): входная информация – данные канала, выходная информация – новый учебный канал в списке каналов учащихся и преподавателей, условно-постоянная – данные существующих каналов.

При нажатии кнопки «Создать задание» в учебном канале, преподавателю будет предложено ввести данные задания и с помощью панели создания заданий. При нажатии на кнопку отправки, задание публикуется в учебном канале. Входная информация – данные задания, выходная информация – новое задание в списке публикаций учебного канала, условно-постоянная информация – данные задания и списка публикаций.

Учащийся:

При нажатии на кнопку деавторизации в настройках приложения, пользователь выйдет из своего аккаунта. Входная информация – выбор опции выхода из аккаунта, выходная – сообщение о выходе из аккаунта, переход на экран авторизации, условно-постоянная – отсутствует.

При нажатии на задание, пользователь попадает на экран с содержимым задания с возможностью взаимодействовать с его содержимым. Входная информация – выбор задания в учебном канале, выходная информация – отображение содержимого задания, условно-постоянная информация – данные о содержимом задания.

При нажатии на публикацию, пользователь попадет на экран с содержимым публикации. Входная информация – выбор учебного канала и публикации, выходная информация – отображение выбранной публикации, условно-постоянная информация – данные о содержимом публикаций.

**4. Эксплуатационные требования**

Требования к применению: Приложение должно работать на мобильных устройствах под управлением опирационных систем Android и iOS.

Требования к реализации: Для реализации мобильного приложения будет использоваться язык программирования Python и библиотека Kivy для поддержки кросс-платформенного приложения и фреймворк Flask для создания достаточной функциональности сервера.

Требования к надежности приложения: Система должна работать безперерывно. У администратора сервера должна быть возможность выгрузить и загрузить копию сайта.

Требования к интерфейсу приложения: При разработке приложения должны быть использованы преимущественно голубые/фиолетовые оттенки. Приложение должно содержать оформление в темной и светлой теме и следующий набор экранов: экран входа, домашний экран, экран настроек, профиля,

Требования к хостингу:

Аппаратные характеристики:

* Оперативная память: не менее 1 ГБ;
* Хранение данных: SSD не менее 40 ГБ;
* Скорость интернет-соединения: не менее 10 Мбит/с.

Программные характеристики:

* Операционная система: Linux;
* База данных: SQLite;
* Язык программирования: Python.

Автоматическая установка необходимых программных компонентов и регулярного обновления системы без простоев и сбоев в работе серверной части.

Надёжность и безопасность хостинга, включая защиту от взломов, резервное копирование данных и поддержка SSL-сертификатов для защищенного соединения с сервером.

* Круглосуточный мониторинг серверов 24/7: сбои устраняются раньше, чем их можно заметить
* Эффективная защита от спама и взлома.
* База данных хранятся на SSD-дисках.
* Все данные проходят через процедуру ежедневного резервного копирования. Копии хранятся в течении 31 дня.

**5 Выбор инструментов разработки**

Для разработки будет выбран язык программирования, который наилучшим образом соответствует потребностям и возможностям проекта. Для такой задачи лучше всего подойдёт язык программирования Python, как для серверной, так и для клиентской части.

Выбор библиотек Flask, Kivy и KivyMD обоснован легкостью реализации обработки запросов на серверной стороне, быстротой клиентской на фреймворке Kivy за счёт большого количества готовых и легких в использовании виджетов и возможностью использования современных элементов интерфейса стиля Material Design расширения KivyMD.

Flask – это фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2.

Kivy — это бесплатная платформа Python с открытым исходным кодом для разработки мобильных приложений и другого мультисенсорного прикладного программного обеспечения с естественным пользовательским интерфейсом.

KivyMD (Kivy Material Design) – это набор виджетов и стилей для фреймворка Kivy, который предоставляет компоненты в стиле Material Design для разработки мобильных приложений.

Преимущества Python:

Простой и читаемый синтаксис: Python разработан так, чтобы код был легко читаемым и понятным. Это делает его отличным выбором для начинающих программистов и способствует повышению производительности разработчиков.

Кроссплатформенность: Python поддерживается на множестве операционных систем, что позволяет запускать код на различных платформах без изменений.

Большое сообщество: Python имеет активное сообщество разработчиков, что означает, что вы можете легко найти поддержку, библиотеки и ресурсы для решения задач.

Множество библиотек и фреймворков: Python предоставляет огромное количество библиотек и фреймворков, которые упрощают разработку различных приложений, включая веб-сайты, научные вычисления, искусственный интеллект и многое другое.

Широкое использование: Python применяется в различных областях, включая веб-разработку, анализ данных, машинное обучение, научные исследования, автоматизацию задач, игры и многое другое.

Открытый исходный код: Python – это свободное программное обеспечение, что означает, что вы можете использовать его бесплатно и иметь доступ к исходному коду для внесения изменений.

Хорошая документация: Python имеет хорошо разработанную и поддерживаемую документацию, что облегчает изучение языка и решение проблем.

Интеграция с другими языками: Python легко интегрируется с языками программирования, такими как C, C++, и Java, что позволяет использовать существующий код и библиотеки. В дальнейшем это может поспособствовать ускорения медленных участков кода за счёт создания аналогичных модулей на C++.

Широкая поддержка сторонних разработчиков: существует множество сторонних инструментов и ресурсов, которые расширяют возможности Python.

Продвинутые возможности: Python поддерживает асинхронное программирование, многопоточность, многозадачность и другие продвинутые функции для разработки сложных приложений.

**6 UX-проектирование пользовательского интерфейса**

Ссылка на проект с дизайном интерфейса: https://www.figma.com/file/ZIXipHfSx4AAfIOm4rZXM0/Untitled?type=design&node-id=0%3A1&mode=design&t=9J0C5vJsoa8m5ym0-1.

**7 UI-проектирование пользовательского интерфейса**

UI-макет прототипа разработан и реализован уже в самом приложении. Ссылка на репозиторий с прототипом: https://github.com/MikutsevichWadim/SkillThread

**8 Диаграмма компонентов**

Диаграмма компонентов (рисунок А.9) была разработана для курсовой работы по предмету «Основы алгоритмизации и программирования».

**9 Реализация**

Исходные файлы проекта доступны в репозитории по ссылке: https://github.com/MikutsevichWadim/SkillThread

**10 Тестирование**

Таблица 10.1 – Отчёт результатах тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тест | Ожидаемый результат | Физический результат | Результат  тестирования |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Проверка полей ввода экрана «Вход» | Заполнение данных | Заполнение данных | Выполнено |
| 2 | Проверка кнопки «Регистрация» экрана «Вход» | Переход на экран «Вход» | Переход на экран «Вход» | Выполнено |
| 3 | Проверка полей ввода экрана «Регистрация» | Заполнение данных | Заполнение данных | Выполнено |
| 4 | Проверка кнопки «Вход» экрана «Регистрация» | Переход на экран «Регистрация» | Переход на экран «Регистрация» | Выполнено |
| 5 | Проверка кнопки «Зарегестрироваться» | Переход домашний экран | Переход домашний экран | Выполнено |
| 6 | Проверка кнопки «Меню» домашнего экрана | Открытие меню | Открытие меню | Выполнено |
| 7 | Проверка сворачиваемости меню | Закрытие меню | Закрытие меню | Выполнено |

**11 Руководство**

Мобильное приложение «SkillThread». Данный программный продукт разрабатывается для студентов и преподавателей, желающим использовать конструктор текстовых материалов, задач и тестов для проверки и обучения учащихся. Периодичность использования: во время обучения или по мере необходимости.

Требования:

* Android 9 и выше
* 1 GB оперативной памяти
* 500 MB свободного пространства
* наличие доступа в интернет
* разрешение на запись/чтения накопителя

**11.2 Руководство программиста**

**11.2.1 Организация данных**

~/assets/icons/ - содержит иконки

~/screens/\*/ - директории экранов, содержит \*.py, \*.kv файлы, \* - название экрана

~/widgets/\*/ - директории переопределенных виджетов, также содержит \*.py, \*.kv файлы, \* - название виджетов

**11.2.2 Структура программы**

~/screens/{screen\_name}/{screen\_name}.py – содержат классы экранов

~/screens/{screen\_name}/{screen\_name}.kv – содержат KV-классы разметки

~/widgets/{widget\_name}/{screen\_name}.py - содержат классы виджетов

~/widgets/{widget\_name}/{screen\_name}.kv - содержат KV-классы виджетов

**11.2.3 Структура и описание процедур и функций пользователя**

Таблица 11.2.3 – структура и описание процедур и функций

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Имя функции | В каком модуле находится | За каким элементов управления закреплена | Назначение |
| 1 | def build(self) | main.py | STApp (MDApp) | Сборка приложения |
| 2 | def open\_menu (self, \_): | screens/ home/ home\_screen.py | HomeScreen (MDScreen) | Открытие бокового меню |
| 3 | def switch\_screen (self, \_): | screens/ home/ home\_screen.py | HomeScreen (MDScreen) | Анимация и переход на другой экран |
| 4 | def login (self): | screens/ login/ login\_screen.py | LoginScreen (MDScreen) | Функция входа |
| 5 | def go\_to\_signup (self): | screens/ login/ login\_screen.py | LoginScreen (MDScreen) | Переход на экран регистрации |
| 6 | def success\_login (self): | screens/ login/ login\_screen.py | LoginScreen (MDScreen) | Функция, выполняющаяся при успешном входе |
| 7 | def signup (self): | screens/ login/ signup\_screen.py | class SignupScreen (MDScreen): | Функция регистрации |
| 8 | def go\_to\_login (self): | screens/ login/ signup\_screen.py | class SignupScreen (MDScreen): | Переход на экран входа |
| 9 | def success\_signup (self): | screens/ login/ signup\_screen.py | class SignupScreen (MDScreen): | Функция, выполняющаяся при успешной регистрации |

10.2.4 Спецификация программы

Таблица 10.2.4 - Спецификация программы

|  |  |  |
| --- | --- | --- |
| № | Имя файла | Назначение |
| 1 | bin/skillthread-0.1.0.1… | Пакет установки |
| 2-6 | data/icon-\*.png | Иконки приложения в разных разрешениях |
| 7-10 | data/channels/\*.png | Иконки каналов |
| 11 | data/presplash-3000.png | Изображение экрана загрузки |
| 12-14 | screens/{название\_экрана}/ {название\_экрана}.kv | KV-макет экрана |
| 15-17 | screens/{название\_экрана}/ {название\_экрана}.py | Модуль экрана |
| 18 | screens/screens.kv | Модуль импорта разметки экранов |
| 19 | screens/\_\_init\_\_.py | Модуль импорта модулей экранов |
| 12-14 | widgets/{название виджета }/ {название виджета }.kv | KV-макеты виджетов |
| 15-17 | screens/{название виджета}/ {название виджета}.py | Модули виджетов |
| 20 | widgets/widgets.kv | Модуль импорта разметки виджетов |
| 21 | widgets/\_\_init\_\_.py | Модуль импорта модулей виджетов |
| 22 | README.md | Краткое писание проекта |
| 23 | buildozer.spec | Конфигурация компиляции |
| 24 | main.kv | Главный модуль KV-разметки |
| 25 | main.py | Главный модуль приложения |

**10.3. Руководство системного программиста (Инсталляция)**

Сначала необходимо загрузить apk-пакет установки из директории bin проекта:

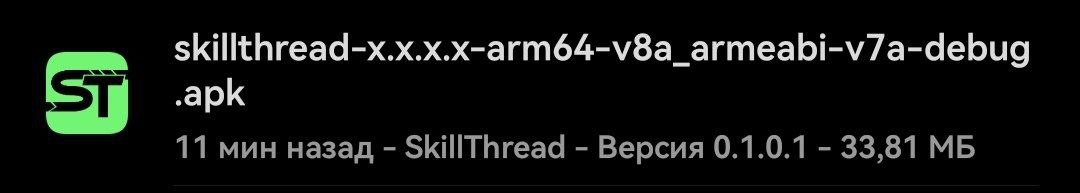


Рисунок – пакет установки

Запустить установщик из файлового менеджера:

Рисунок – подтверждение установки

Подтвердить установку и дождаться её окончания:

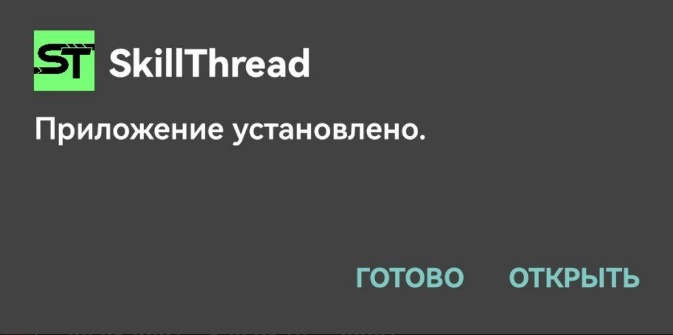


Рисунок – приложение установлено

**10.4 Руководство пользователя**

**10.4.1 Запуск программы по работе с программой**

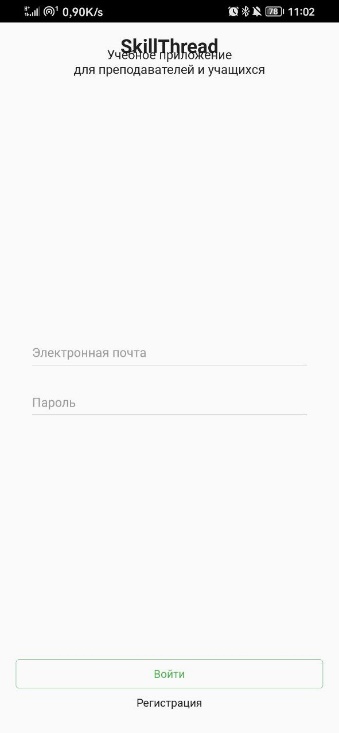
Запуск программы выполняется после установки по клику на иконку приложения:



Рисунок – иконка приложения

**10.4.2 Инструкции по работе с программой**

После запуска появляется окно входа в аккаунт

Рисунок – экран входа

Здесь вводятся данные аккаунта

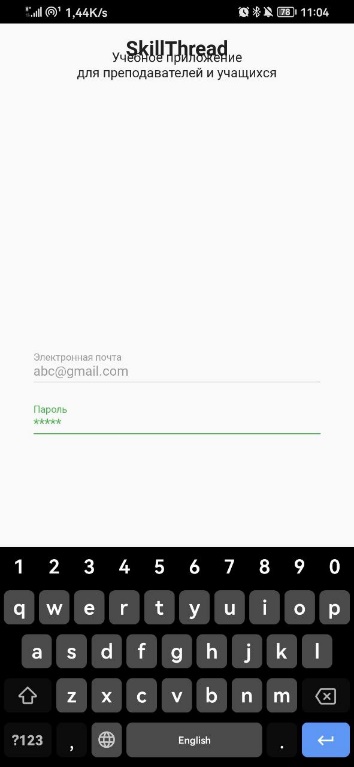


Рисунок – заполнение полей

Если аккаунт не создан, имеется возможность перейти на экран регистрации

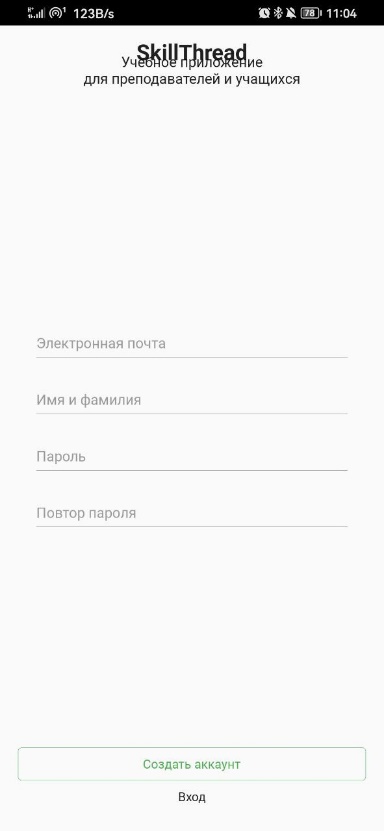


Рисунок – экран регистрации

После завершения входа или регистрации появляется домашний экран со списком каналов

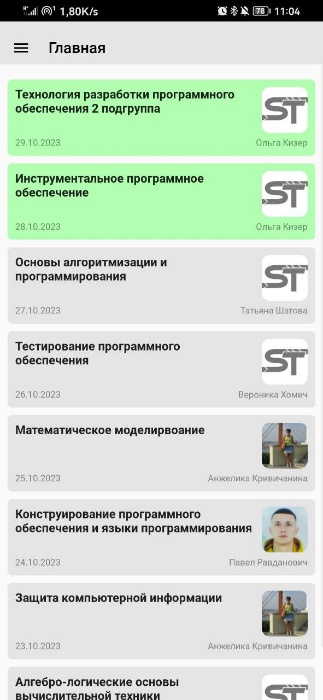


Рисунок – домашний экран

При нажатии на значок в верхнем левом углу появляется меню приложения

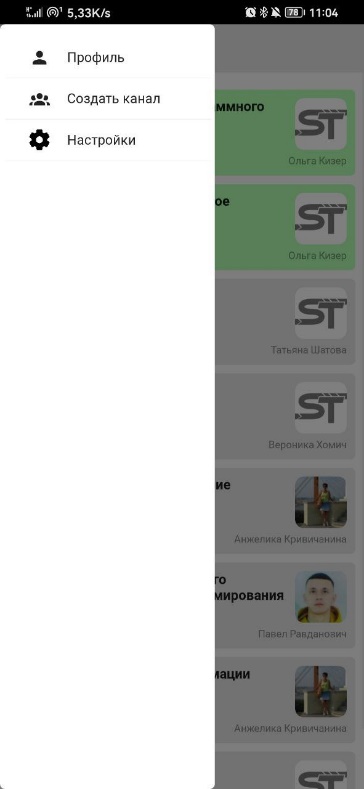


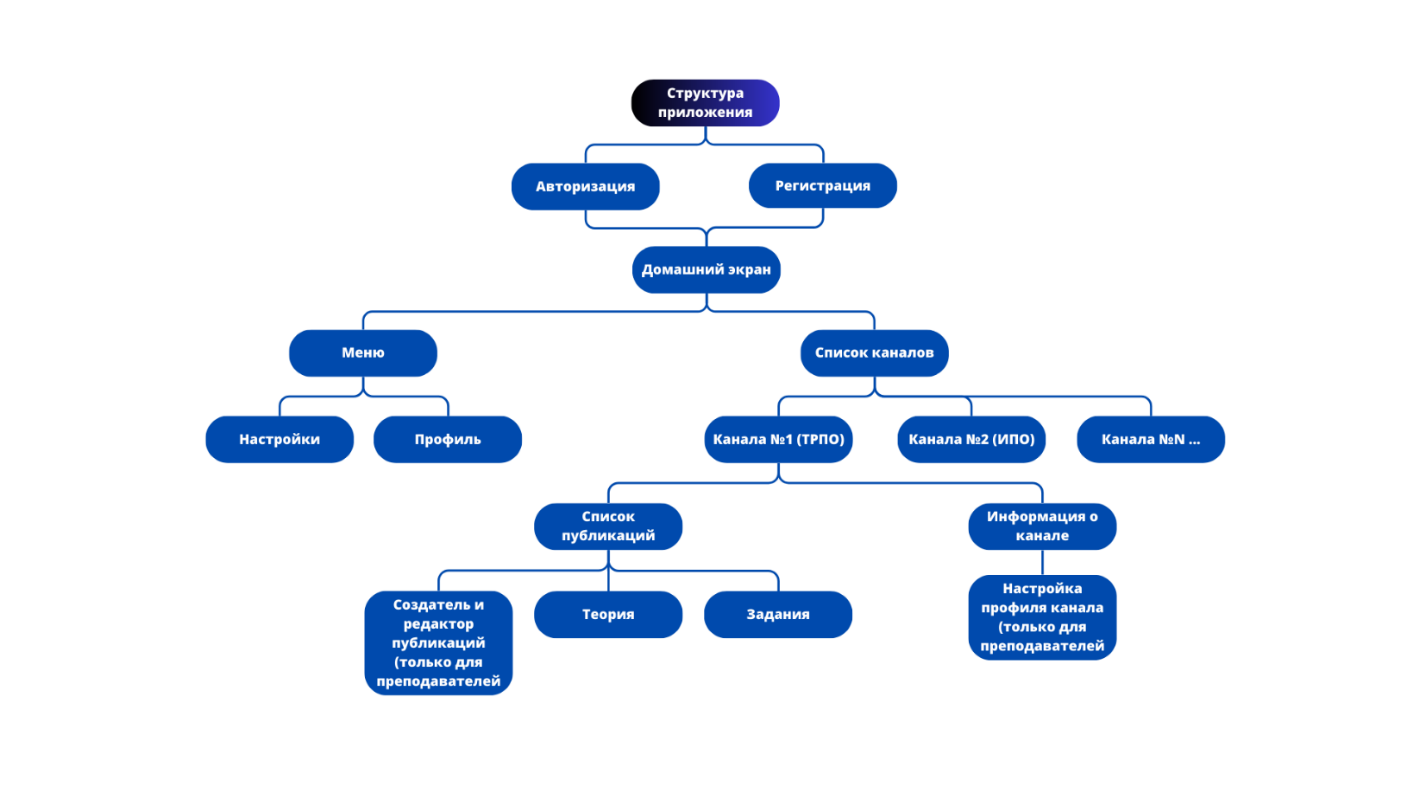
Рисунок – меню приложения

**10.4.3 Завершение работы с программой**

Завершение работы производится системным выходом из приложения

Рисунок – выход через менеджер задач Андроид

**Приложение А**

Рисунок А.1 – Структура приложения

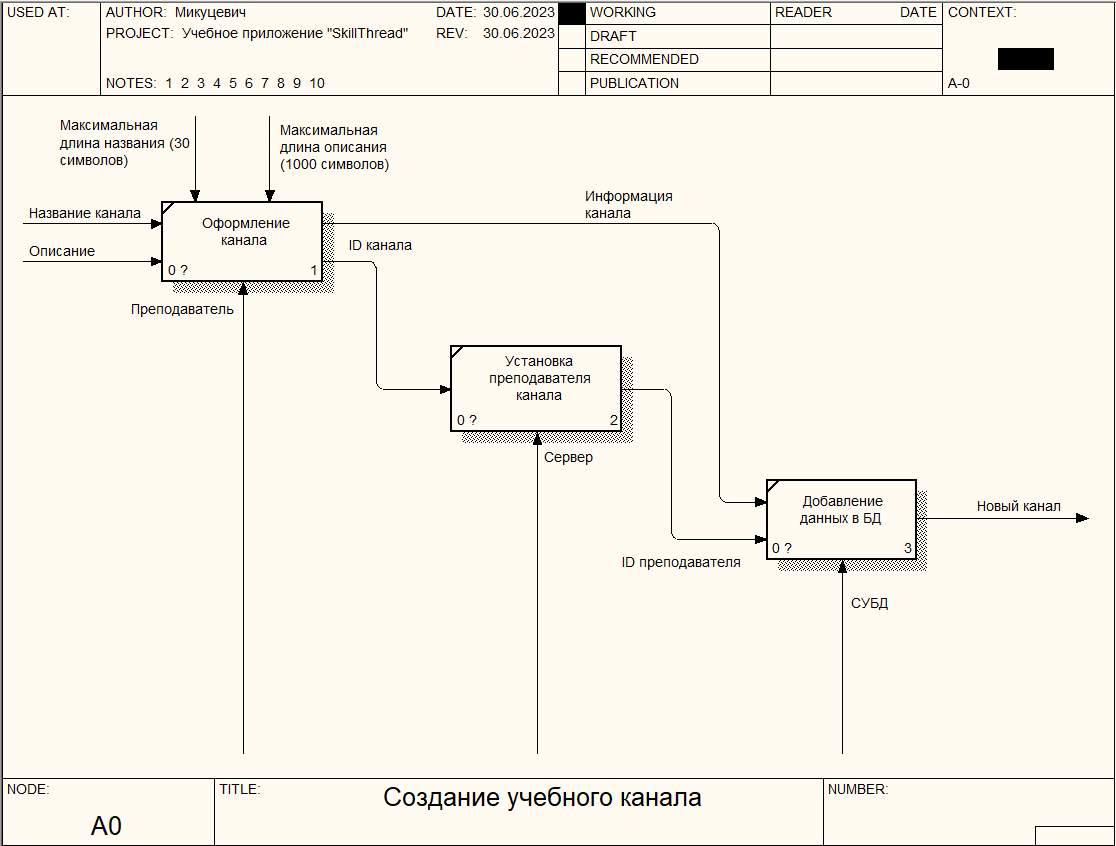
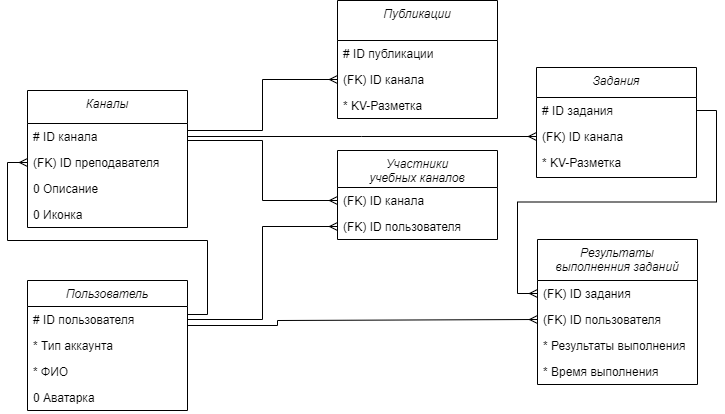
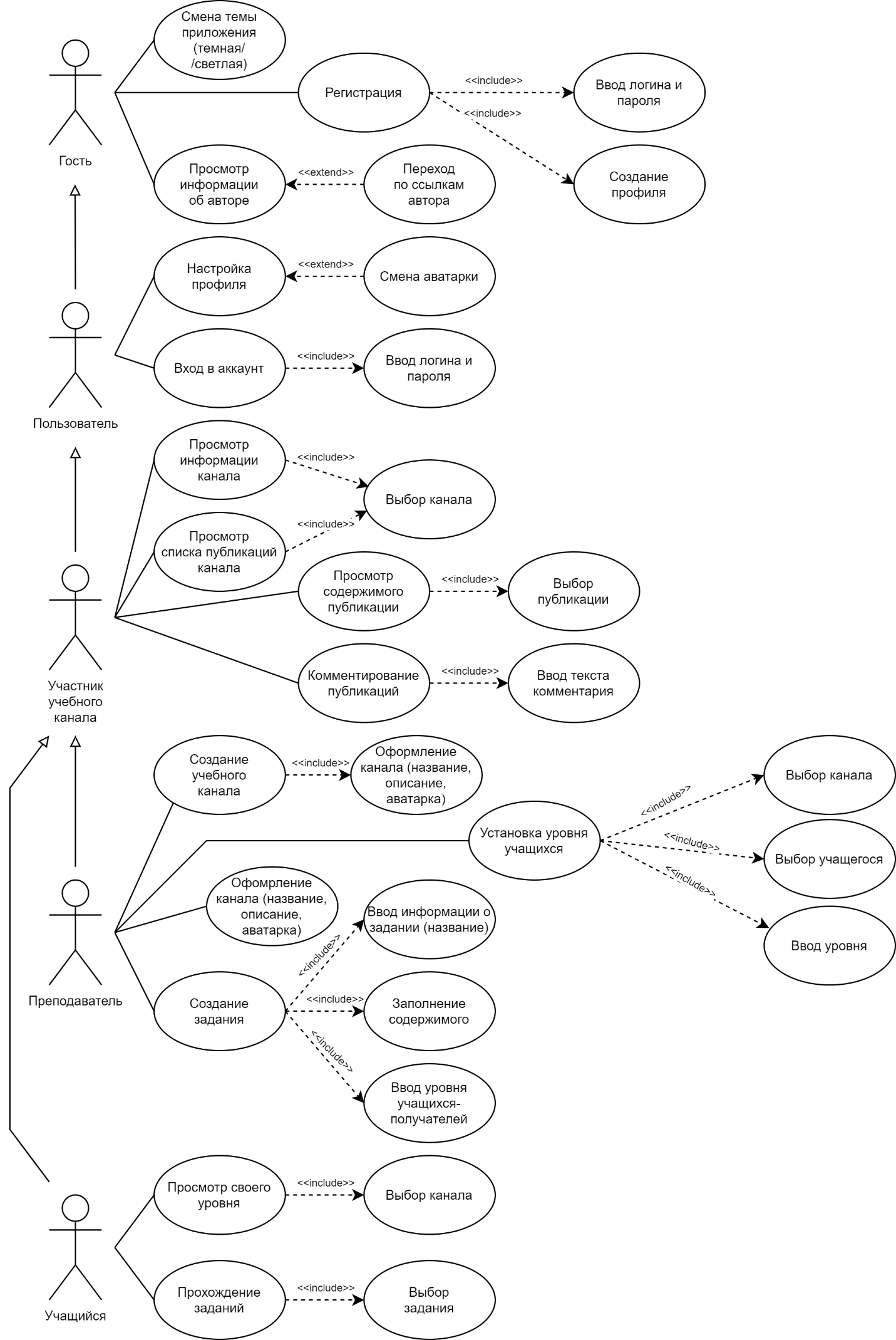
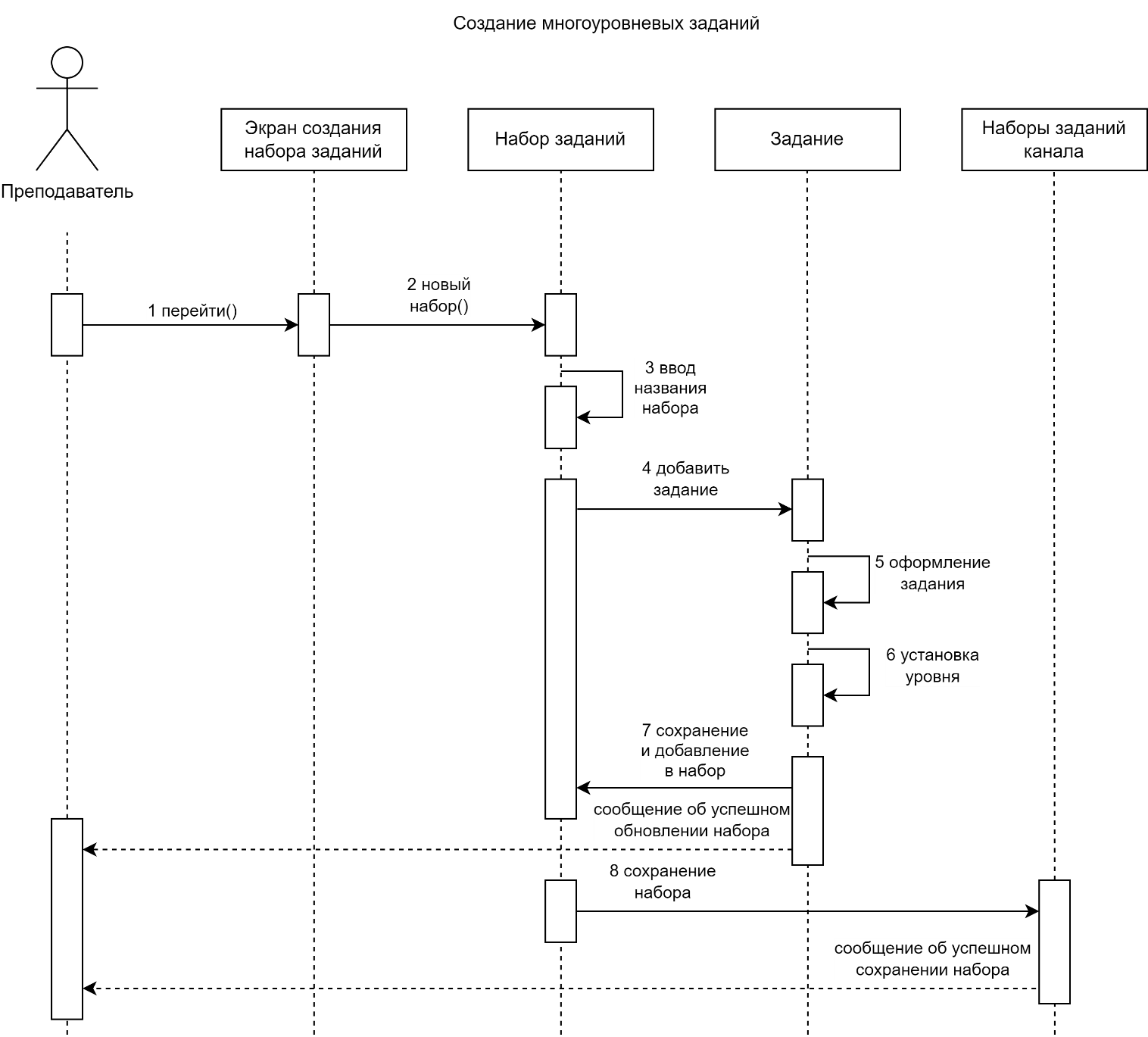
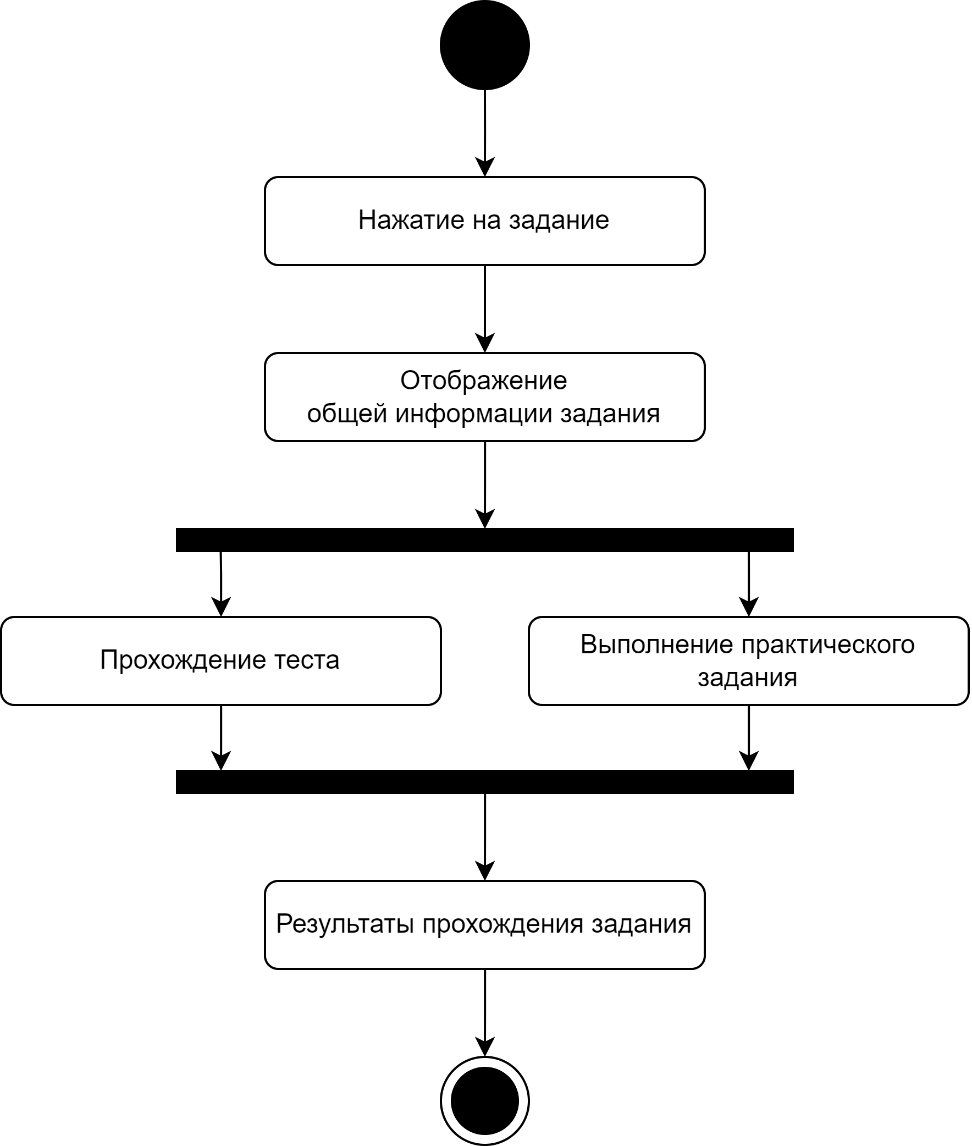


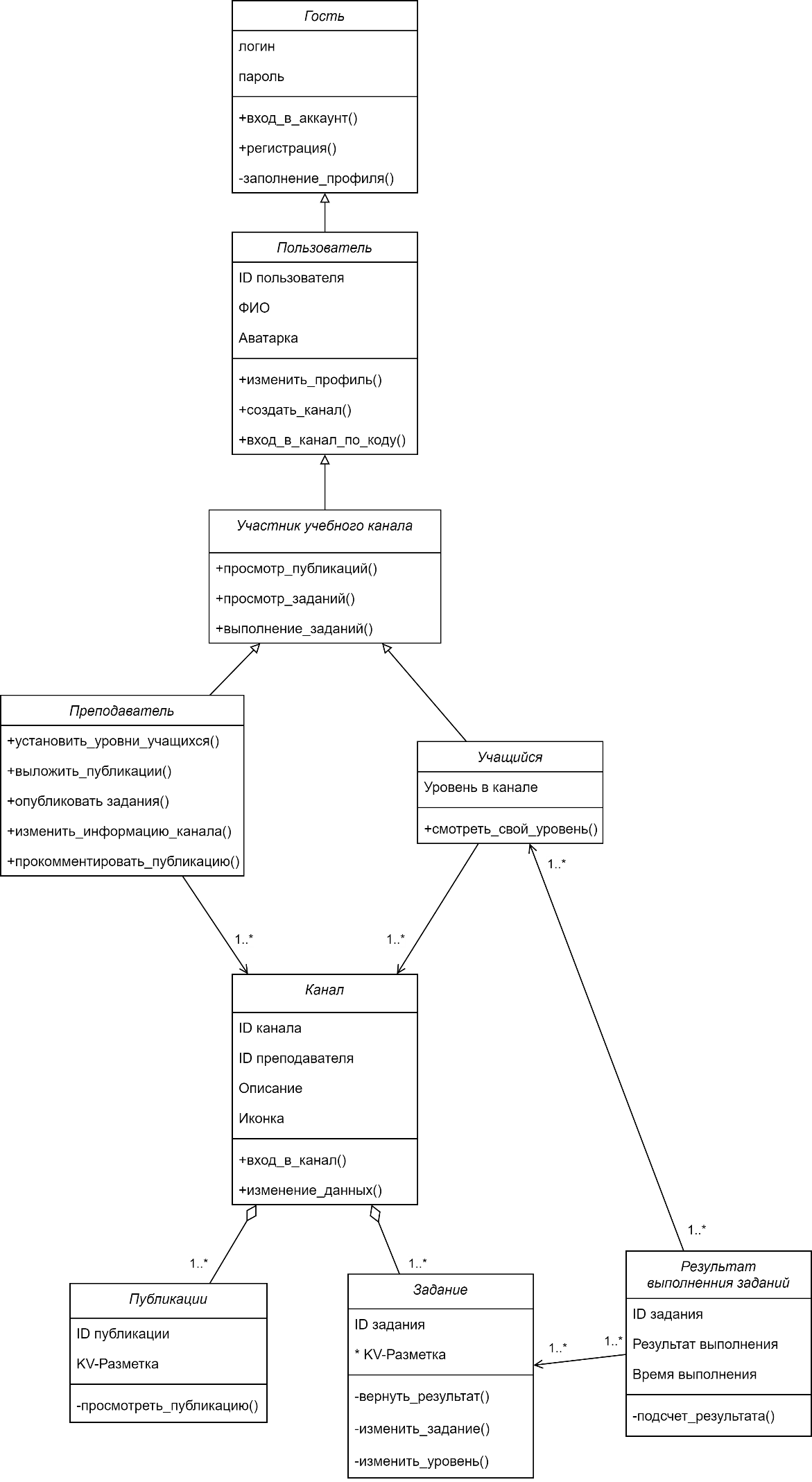
Рисунок А.2 – Функциональная модель создания учебного канала

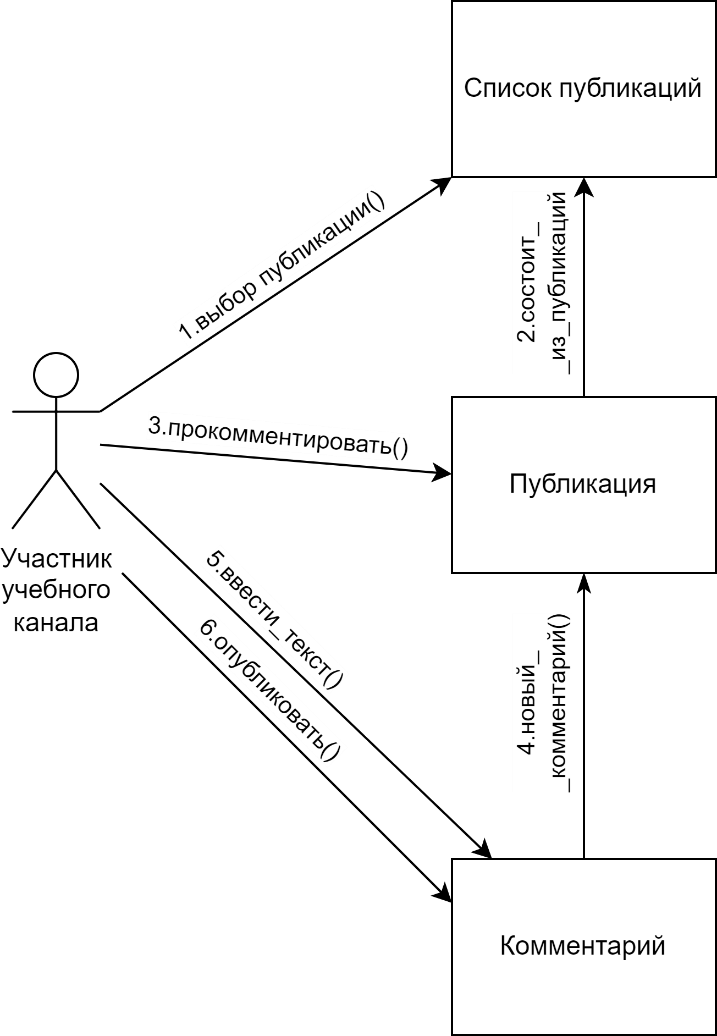
Рисунок А.3 – Модель данных

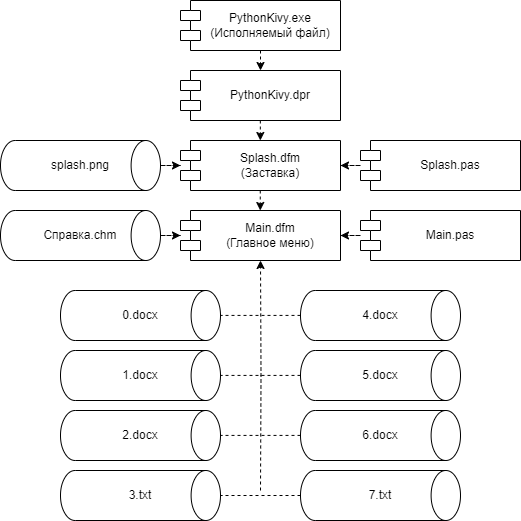
Рисунок А.4 – Диаграмма вариантов использования

Рисунок А.5 – Диаграмма последовательностей процесса создания заданий

Рисунок А.6 – Диаграмма деятельности процесса прохождения задания

Рисунок А.7 – Диаграмма классов

Рисунок А.8 – Диаграмма объектов процесса комментирования публикации

Рисунок А.9 – Диаграмма компонентов