**Лабораторная работа № 24**

**Документирование программного обеспечения (Разработка структуры документации, приложений к ПО, разработка руководства пользователя).**

**Цель работы:**

1. Познакомиться со стандартами ЕСПД .

2. Изучить основные этапы построения руководства пользователя.

3. Научиться разрабатывать руководство пользователя.

**Порядок выполнения работы**

1. Изучить краткие теоретические сведения по следующей теме: «Документирование ПО». Если

информация отсутствует в конспекте, то необходимо ее законспектировать.

2. Изучить стандарты оформления текстовых документов. Познакомиться со стандартами ЕСПД.

Изучить структуру Руководства системного программиста (ГОСТ 19.503-79) и Руководства

программиста (ГОСТ 19.504–79) и Руководство Оператора (.ГОСТ 19.505-79)

3. Разработать руководство пользователя для программного продукта (прототипа),

разработанного по своему индивидуальному заданию .

Руководство пользователя разработать по следующему плану (смотри файл

РуководствоПользователя\_план. Doc).

4. Ответить на контрольные вопросы.

5. Защитить работу и сдать ее преподавателю.

**Ход работы**

**1 Общие сведения о программном продукте**

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

**2 Руководство программиста**

**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

**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

**2.3 Структура и описание процедур и функций пользователя**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | 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 |

2.4 Спецификация программы

Таблица № - 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 |

3. Руководство системного программиста (Инсталляция)

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

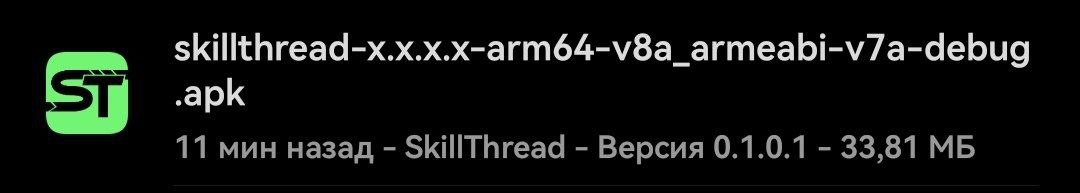


Рисунок – пакет установки

Запустить установщик из файлового менеджера:

Рисунок – подтверждение установки

Подтвердить установку и дождаться её окончания:

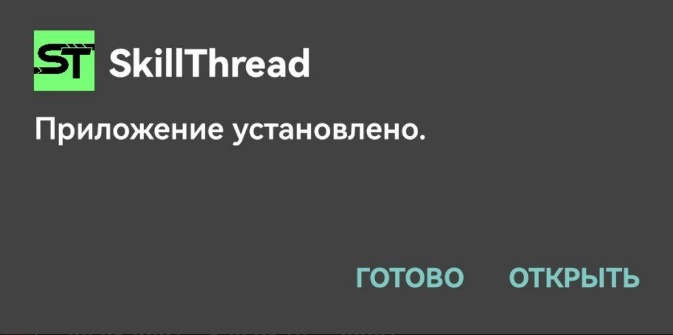


Рисунок – приложение установлено

4 Руководство пользователя

4.1 Запуск программы по работе с программой

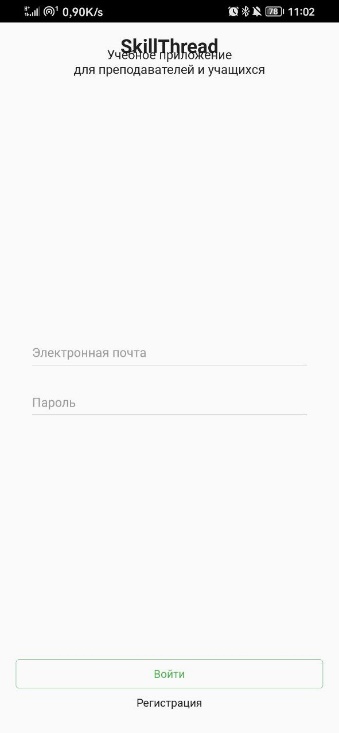
Запуск программы выполняется после установки по клику на иконку приложения:



Рисунок – иконка приложения

4.2 Инструкции по работе с программой

После запуска появляется окно входа в аккаунт

Рисунок – экран входа

Здесь вводятся данные аккаунта

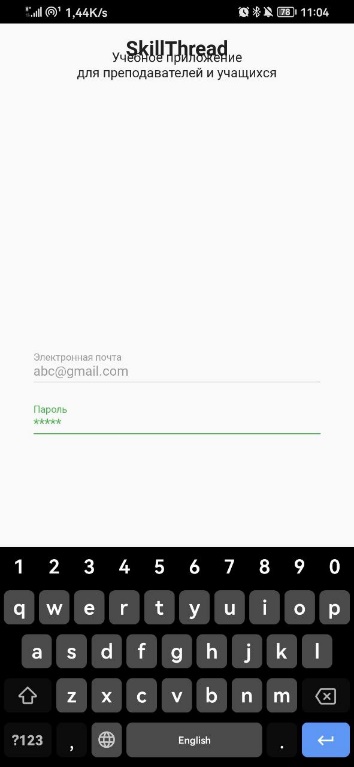


Рисунок – заполнение полей

Если аккаунт не создан, имеется возможность перейти на экран регистрации

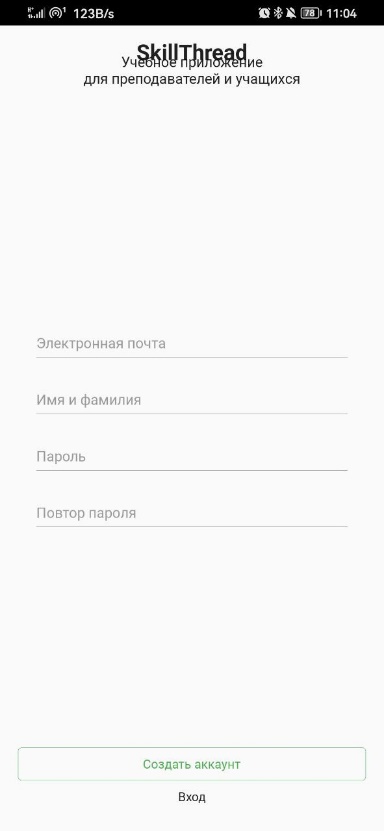


Рисунок – экран регистрации

После завершения входа или регистрации появляется домашний экран со списком каналов

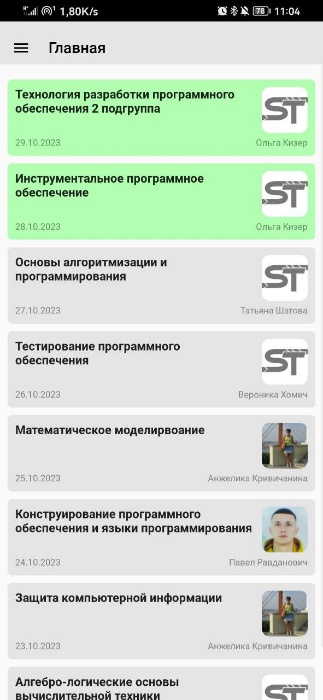


Рисунок – домашний экран

При нажатии на значок в верхнем левом углу появляется меню приложения

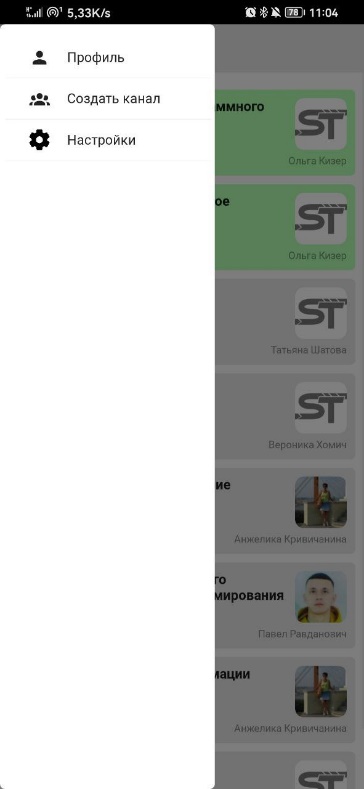


Рисунок – меню приложения

4.3 Завершение работы с программой

Завершение работы производится системным выходом из приложения

Рисунок – выход через менеджер задач Андроид

**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Перечислите виды программной документации и дайте им краткую характеристику.

Техническое задание (ТЗ):

Характеристика: Определяет требования к программному продукту, его функциональность, архитектуру, производительность и другие технические аспекты.

Спецификация требований:

Характеристика: Более детальное описание функциональных и нефункциональных требований к программе. Включает в себя детали по взаимодействию с пользователем, функциональности и производительности.

Программная спецификация:

Характеристика: Подробное техническое описание программы, включая архитектуру, структуру кода, алгоритмы и структуры данных.

Руководство пользователя:

Характеристика: Описание того, как использовать программу, включая инструкции по установке, настройке и основным функциям. Предназначено для конечных пользователей.

Руководство по разработке:

Характеристика: Направлено на разработчиков и содержит информацию о структуре кода, правилах написания, стиле кодирования и другие технические аспекты разработки.

Тестовая документация:

Характеристика: Содержит информацию о тестовых случаях, стратегии тестирования и результаты проведенных тестов.

Документация по API:

Характеристика: Описывает интерфейс программы для взаимодействия с другими программами или компонентами.

Документация по безопасности:

Характеристика: Содержит информацию о мерах безопасности, рекомендации по обеспечению защиты от угроз и восстановлению после инцидентов.

2. Опишите назначение стандартизации в области документирования программных продуктов.

Стандартизация в области документирования программных продуктов служит нескольким целям:

Улучшение качества документации: Стандарты определяют общие требования к структуре, содержанию и форматированию документов, что помогает создавать более четкую, понятную и полезную документацию.

Снижение риска ошибок: Стандартизированные подходы к документированию могут снизить вероятность возникновения ошибок и недоразумений, так как разработчики и другие участники проекта следуют общим правилам.

Обеспечение единообразия: Стандарты позволяют обеспечить единообразие документации в рамках проекта или между различными проектами. Это упрощает понимание и обмен информацией.

Сокращение времени разработки: Использование стандартов ускоряет процесс создания документации, так как разработчики могут следовать уже установленным правилам и шаблонам.

Обеспечение требований заказчика: Стандарты в документировании могут включать требования заказчика, что помогает создать продукт, соответствующий ожиданиям заказчика.

3. Что обозначает аббревиатура ЕСПД и какие документы входят в ее состав?

ЕСПД - это Единая система программной документации.

В состав ЕСПД обычно входят следующие документы:

Техническое задание (ТЗ): Определяет требования к программе.

Программная спецификация: Описывает детали реализации программы.

Руководство пользователя: Инструкции по установке и использованию программы.

Руководство по разработке: Документация для разработчиков.

Документация по тестированию: Информация о тестировании программы.

4. Опишите назначение руководства системного программиста, программиста, оператора и пользователя?

Руководство системного программиста: Ориентировано на архитектурные и технические аспекты разработки системного программного обеспечения.

Руководство программиста: Описывает принципы и правила написания кода, структуру программы.

Руководство оператора: Предоставляет инструкции по установке, настройке и обслуживанию программы.

Руководство пользователя: Содержит информацию для конечных пользователей о использовании программы.

5. Опишите назначение справочной системы программного продукта. К какому виду программной документации она относится? Поясните, почему возникает необходимость в разработке и подключении справочной системы для каждого программного продукта.

Справочная система предназначена для предоставления пользователям подробной информации о программе. Она относится к руководству пользователя и помогает пользователям быстро находить ответы на вопросы, касающиеся использования программного продукта.

Причины разработки и подключения справочной системы включают:

Улучшение опыта пользователя: Пользователи могут легко находить нужную информацию, улучшая их взаимодействие с программой.

Сокращение времени поддержки: Хорошая справочная система может снизить нагрузку на службу поддержки, так как пользователи могут сами находить ответы на свои вопросы.

Повышение эффективности обучения: Новые пользователи могут быстрее освоить программу, обращаясь к справочной системе.

Справочная система относится к виду программной документации, предназначенной для конечных пользователей.