



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных  
технологий

## **Общий отчёт по проектной работе**

по дисциплине «Технологии разработки программных приложений»

**Тема практической работы:** «Мобильное приложение»

**Выполнил:**

Студент группы ИКБО-16-19

Горшенев Л.А.

**Проверил:**

Степанов П.В.

2021 г.

## Часть 1. Система контроля версий и оформление кода

### Пункт 1. Выбор темы

- Мобильная разработка

### Пункт 2. Состав команды

- Горшенев Л.А.
- Салогуб М.А.

### Пункт 3. User Story

Заголовок	Запись информации	
Заказчик	Как	Пользователь
Примечание	Я хочу	Создавать текстовые заметки
Цель	Чтобы	Было удобнее планировать день, составлять списки дел и записывать мысли и идеи

Заголовок	Голосовые заметки	
Заказчик	Как	Пользователь
Примечание	Я хочу	Создавать голосовые заметки и осуществлять голосовой набор
Цель	Чтобы	Быстро записывать идеи

Заголовок	Напоминание и уведомление	
Заказчик	Как	Пользователь
Примечание	Я хочу	Иметь возможность ставить напоминания и получать уведомления
Цель	Чтобы	Вовремя вспоминать о важных делах

Заголовок	Редактирование заметок	
Заказчик	Как	Пользователь
Примечание	Я хочу	Иметь возможность редактировать заметки
Цель	Чтобы	Вносить какие-либо изменения, добавлять/удалять информацию

Заголовок	Цветовая тема	
Заказчик	Как	Пользователь
Примечание	Я хочу	Иметь возможность выбора цветовой темы
Цель	Чтобы	Сделать использование приложения более приятным

## Пункт 4. Макет

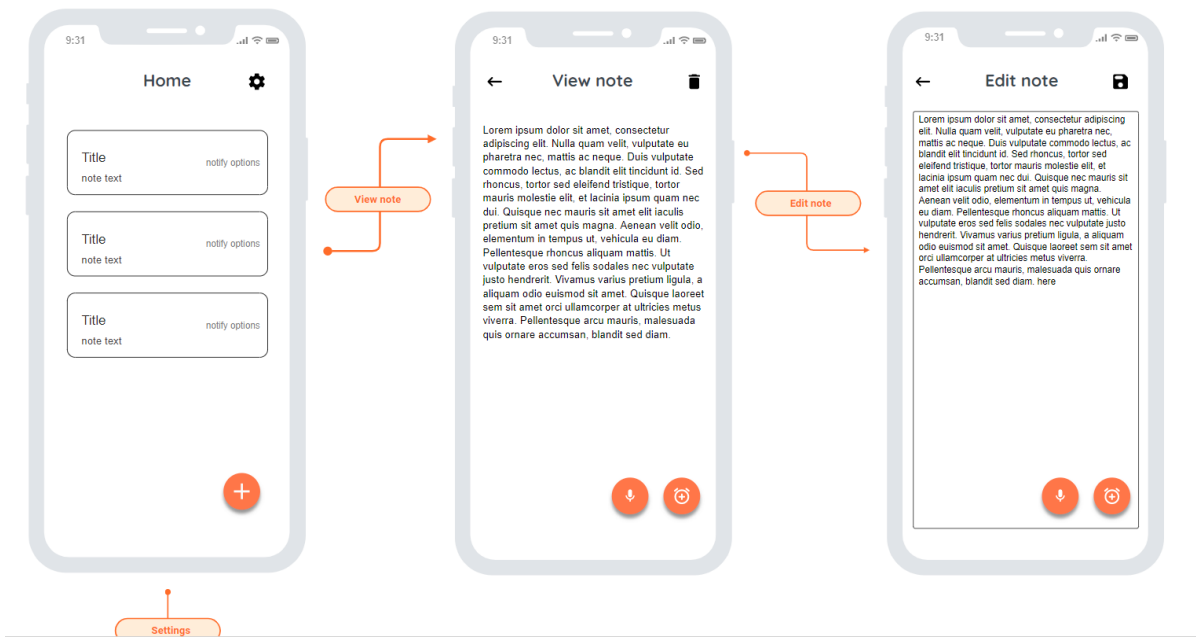


Рисунок 1. Макет



Рисунок 2. Макет

## Пункт 5. Репозиторий

- <https://github.com/trpp-mirea-ikbo-16-19/proekty-mosaic-mob-dev-notes>

## Часть 2. Система сборки Gradle

### Пункт 1. Readme.md

Содержимое файла Readme.md описано в листинге 1.

```
# Notes application with Flutter
```

```
Link go GooglePlay: https://play.google.com/store/apps/details?id=com.company55.notes
```

```
## Screens
```

```
In this app you are able to make notes, delete, edit them using voice input and schedule notifications.
```

```
### Main
```

```
In the Home page you can open or delete existing note, create new or open settings.
```

```
<p align="center" style="">  
    
    
</p>
```

```
### Settings
```

```
Here you can change theme or open this repository
```

```
<p align="center" style="">  
    
</p>
```

```
### View Note
```

```
In this screen you can delete or edit note, use voice input and set notification
```

```
<p align="center" style="">  
    
    
</p>
```

```
### Set notification
```

```
In this screen you can schedule notification.
```

```
<p align="center" style="">  
    
</p>
```

Листинг 1. Файл Readme.md

## Пункт 2. Файл сборки gradle

Содержимое файла build.gradle описано в листинге 2, а результат сборки показан на рис. 3

```
def localProperties = new Properties()
def localPropertiesFile = rootProject.file('local.properties')
if (localPropertiesFile.exists()) {
    localPropertiesFile.withReader('UTF-8') { reader ->
        localProperties.load(reader)
    }
}

def flutterRoot = localProperties.getProperty('flutter.sdk')
if (flutterRoot == null) {
    throw new GradleException("Flutter SDK not found. Define
location with flutter.sdk in the local.properties file.")
}

def flutterVersionCode =
localProperties.getProperty('flutter.versionCode')
if (flutterVersionCode == null) {
    flutterVersionCode = '1'
}

def flutterVersionName =
localProperties.getProperty('flutter.versionName')
if (flutterVersionName == null) {
    flutterVersionName = '1.0'
}

apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply from:
"$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"

def keystoreProperties = new Properties()
def keystorePropertiesFile = rootProject.file('key.properties')
if (keystorePropertiesFile.exists()) {
    keystoreProperties.load(new
FileInputStream(keystorePropertiesFile))
}

android {
    compileSdkVersion 30

    sourceSets {
        main.java.srcDirs += 'src/main/kotlin'
    }
}
```

```

defaultConfig {
    applicationId "ru.smstretching.appstudio"
    minSdkVersion 16
    targetSdkVersion 30
    versionCode flutterVersionCode.toInteger()
    versionName flutterVersionName
    multiDexEnabled true
}

signingConfigs {
    release {
        keyAlias keystoreProperties['keyAlias']
        keyPassword keystoreProperties['keyPassword']
        storeFile keystoreProperties['storeFile'] ?
file(keystoreProperties['storeFile']) : null
        storePassword keystoreProperties['storePassword']
    }
}

buildTypes {
    release {
        signingConfig signingConfigs.release
    }
}

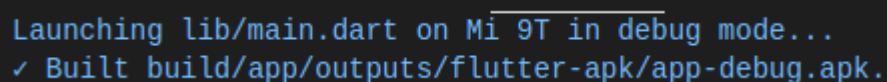
lintOptions {
    checkReleaseBuilds false
    abortOnError false
}

flutter {
    source '../..'
}

dependencies {
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation 'com.yandex.android:maps.mobile:4.0.0-full'
    implementation 'com.android.support:multidex:1.0.3'
}

```

Листинг 2. Файл build.gradle



```

Launching lib/main.dart on Mi 9T in debug mode...
✓ Built build/app/outputs/flutter-apk/app-debug.apk.

```

Рисунок 3. Результат сборки

## Часть 3. Система контейнеризации Docker

### Пункт 1

Написать Dockerfile, собрать образ, запустить контейнер (и записать команду для его запуска). Для монтирования создайте директорию data и в ней файл student.txt, содержащий ФИО, название группы и номер варианта.

Для установки пакетов использовать команду `apt install -y название-пакета`. В качестве примера можно использовать Dockerfile из раздела 7.

Чётные варианты:

- необходимо использовать базовый образ `ubuntu:20.10`
- примонтировать файл `data/student.txt` как `/mnt/files/student.txt` в контейнере.

Нечётные варианты:

- необходимо использовать базовый образ `ubuntu:20.04`
- примонтировать директорию `data` в директорию `/mnt/files/` в контейнере.

Запустить веб-сервер, отображающий содержимое `/mnt/files`, в хостовой системе должен открываться на порту (`8800 + номер варианта`). Например, для 22-го варианта это порт 8822

Установить пакет `jq`.

### Результаты:

```
FROM ubuntu:20.10

RUN apt update \

&& apt install -y jq python3

ADD ./data/Student.txt /mnt/files/Student.txt

EXPOSE 80

RUN echo $(jq --version)

CMD ["python3", "-m", "http.server", "-d", "/mnt/files", "80"]
```

Листинг 3. Содержимое Dockerfile

Сборка образа:

```
sudo docker build -t trpp3 .
```

Запуск контейнера:

```
sudo docker run --rm -it -p8810:80 trpp3
```

Вывод команды сборки:

```
Sending build context to Docker daemon 6.656kB
```

```
Step 1/6 : FROM ubuntu:20.04
```

```
---> 7e0aa2d69a15
```

```
Step 2/6 : RUN apt update && apt install -y jq python3
```

```
---> Using cache
```

```
---> 432c3d706252
```

```
Step 3/6 : ADD ./data/Student.txt /mnt/files/Student.txt
```

```
---> Using cache
```

```
---> 4aa71807e806
```

```
Step 4/6 : EXPOSE 80
```

```
---> Using cache
```

```
---> 023336e7cb9f
```

```
Step 5/6 : RUN echo $(jq --version)
```

```
---> Using cache
```

```
---> 430d0d775c76
```

```
Step 6/6 : CMD ["python3", "-m", "http.server", "-d",  
"/mnt/files", "80"]
```

```
---> Using cache
```

```
---> 1d75a31d4b46
```

```
Successfully built 1d75a31d4b46
```

```
Successfully tagged trpp3:latest
```

Вывод контейнера после запуска:

```
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
172.17.0.1 - - [09/May/2021 07:42:33] "GET / HTTP/1.1" 200 -
```

```
172.17.0.1 - - [09/May/2021 07:42:36] "GET /Student.txt  
HTTP/1.1" 200 -
```

Листинг 4. Команды для работы с докером и их вывод



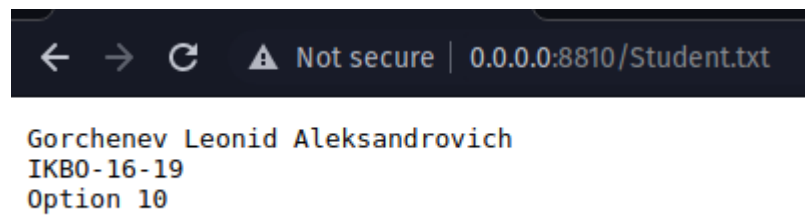


Рис 4. Скриншот файла Student.txt, полученного из контейнера