Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Лабораторная работа №11**

**по дисциплине**

**«Технологии разработки мобильных приложений»**

**Знакомство с Flutter**

**Выполнил**:

ст. гр. ПРИ-120

Д. А. Грачев

**Принял**:

Петрова А. И.

Владимир, 2023

ЦЕЛЬ РАБОТЫ

Реализовать приложение с использованием фреймворка Flutter.

ВЫПОЛНЕНИЕ РАБОТЫ

1. Создадим виджет, который выводит на экран текст

class MyFirstWidget extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return Text("Грачев Даниил Алексеевич", textDirection: TextDirection.ltr);  
 }  
}

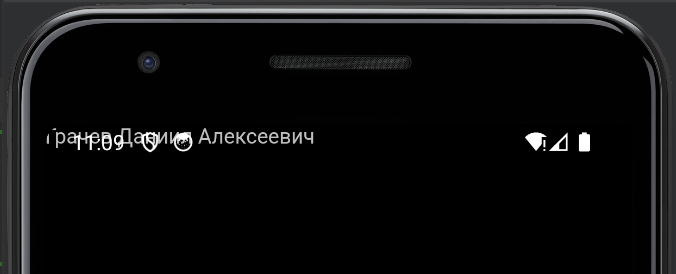


Рисунок 1. Вывод текста на экран

1. Создадим виджет, который отображает картинку из ассетов, для этого добавим ассет в pubspec.yaml и саму картинку в файлы проекта

assets:  
 - assets/dog.jpg

Widget build(BuildContext context) {  
 return Image.asset('assets/dog.jpg');  
}

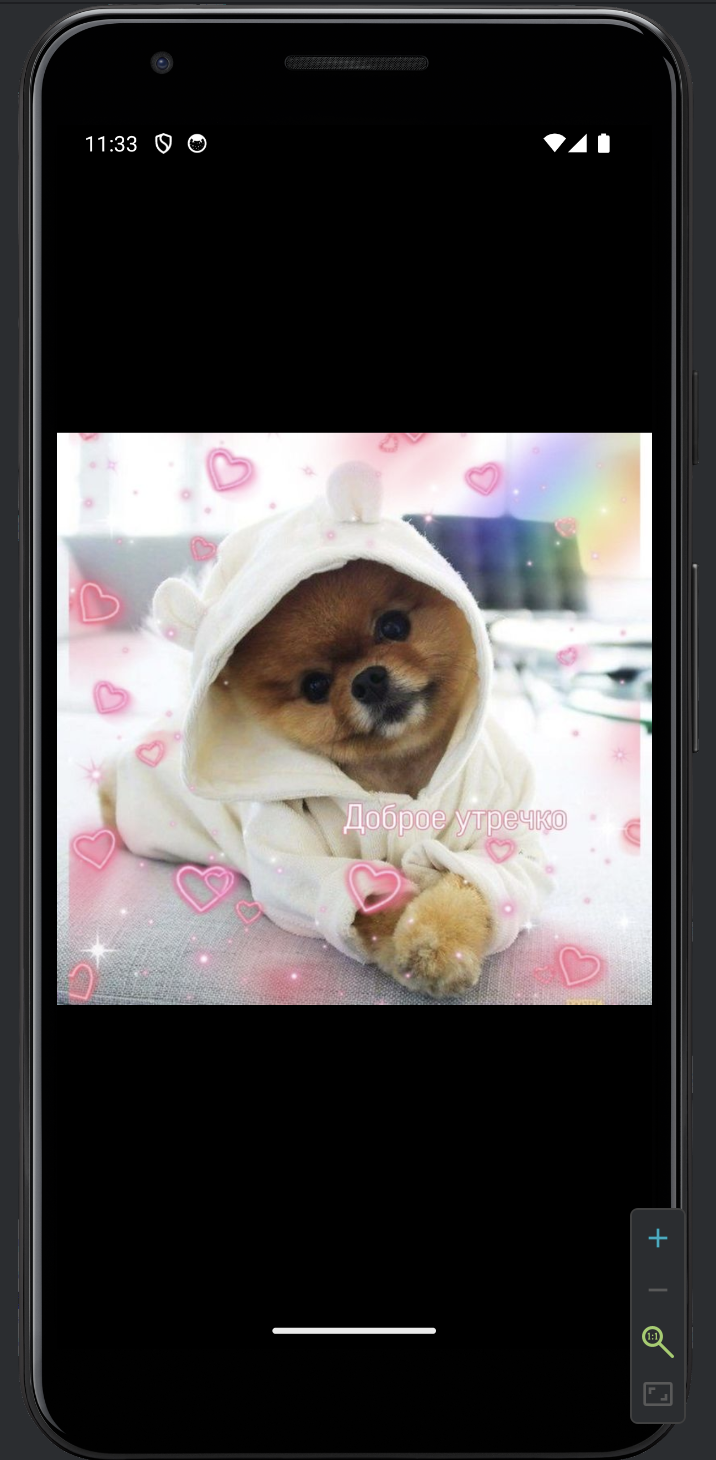


Рисунок 2. Виджет картинки

1. Добавим виджет центрирования, чтобы текст и картинка отображались по центру экрана

return Center(  
 child: Column(  
 children: [title, image],  
 mainAxisSize: MainAxisSize.min,  
 ),  
);

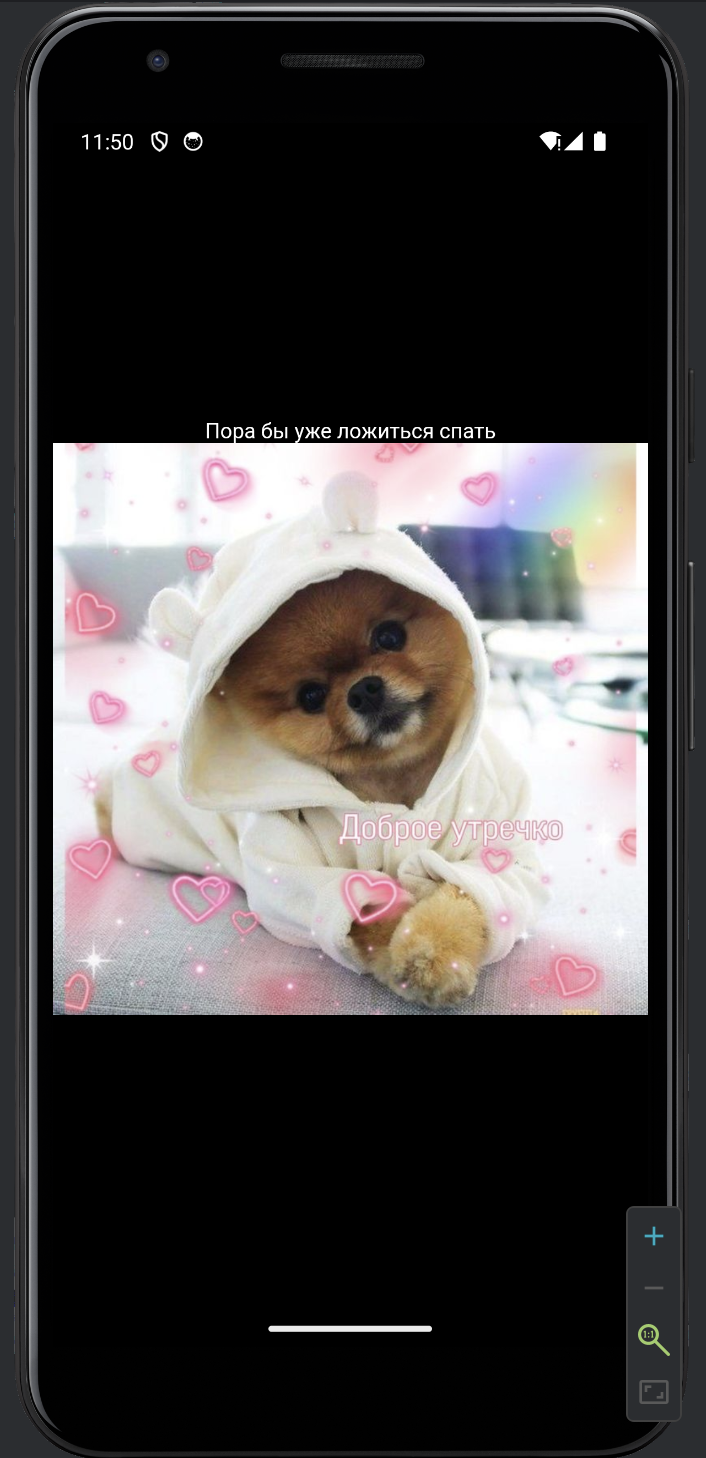


Рисунок 3. Отцентрированный виджет

1. С помощью Material App изменим приложение

return MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text("Grachev`s App"),  
 ),  
 body: Center(  
 child: Column(  
 children: [title, image],  
 mainAxisSize: MainAxisSize.min,  
 ),  
 ),  
 ),  
);

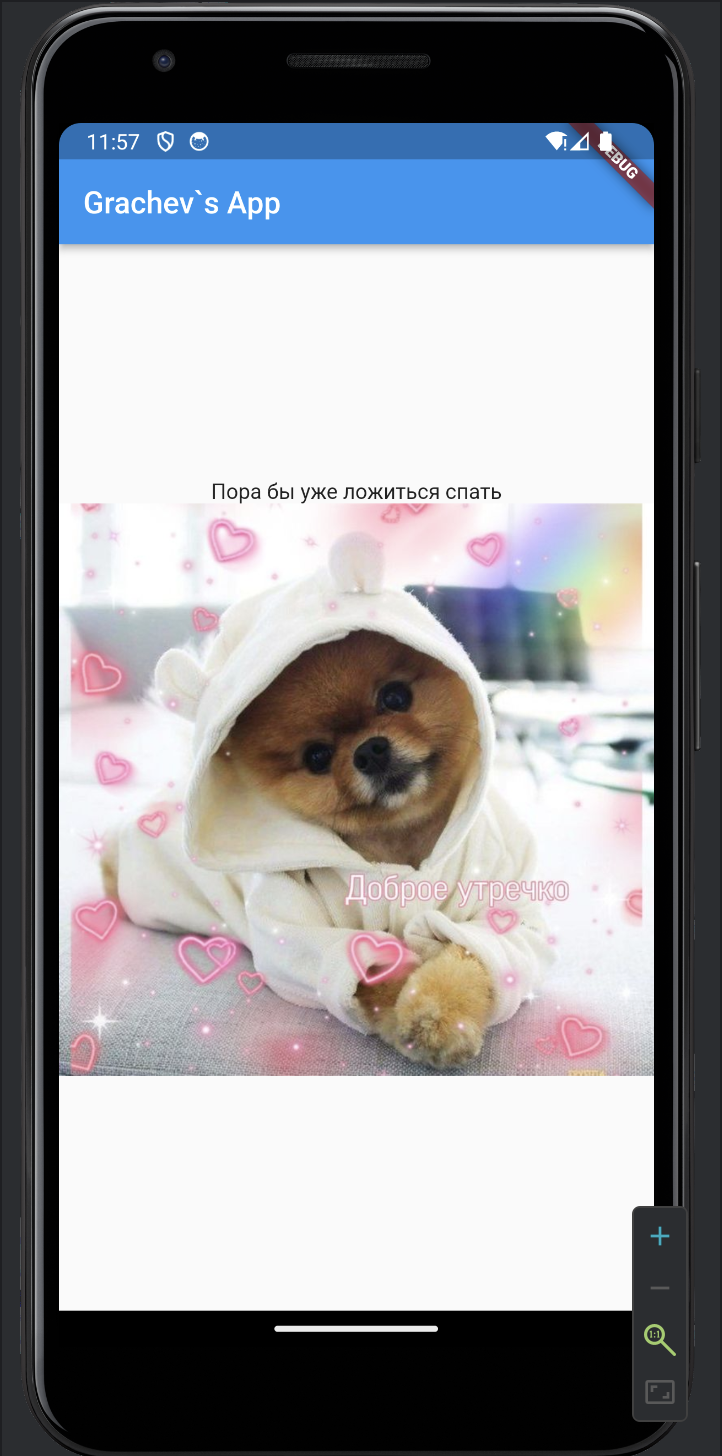


Рисунок 4. Приложение с Material оберткой

1. Создадим приложение с состоянием и сменой картинок по нажатию

class MyState extends State<MySecondWidget> {  
 String url = "https://source.unsplash.com/random/800x600";  
  
 void changeURL() {  
 setState(() {  
 url = "https://source.unsplash.com/random/800x600?q=${new DateTime.now().millisecondsSinceEpoch}";  
 });  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Scaffold(  
 body: Center(  
 child:  
 Column(  
 mainAxisSize: MainAxisSize.min,  
 children: <Widget>[  
 ElevatedButton(  
 child: Text("Press Me"),  
 onPressed: changeURL,  
 ),  
 Image.network(url)  
 ]  
 )  
 )  
 )  
 );  
 }  
}

class MySecondWidget extends StatefulWidget {  
 @override  
 State<StatefulWidget> createState() {  
 return MyState();  
 }  
}

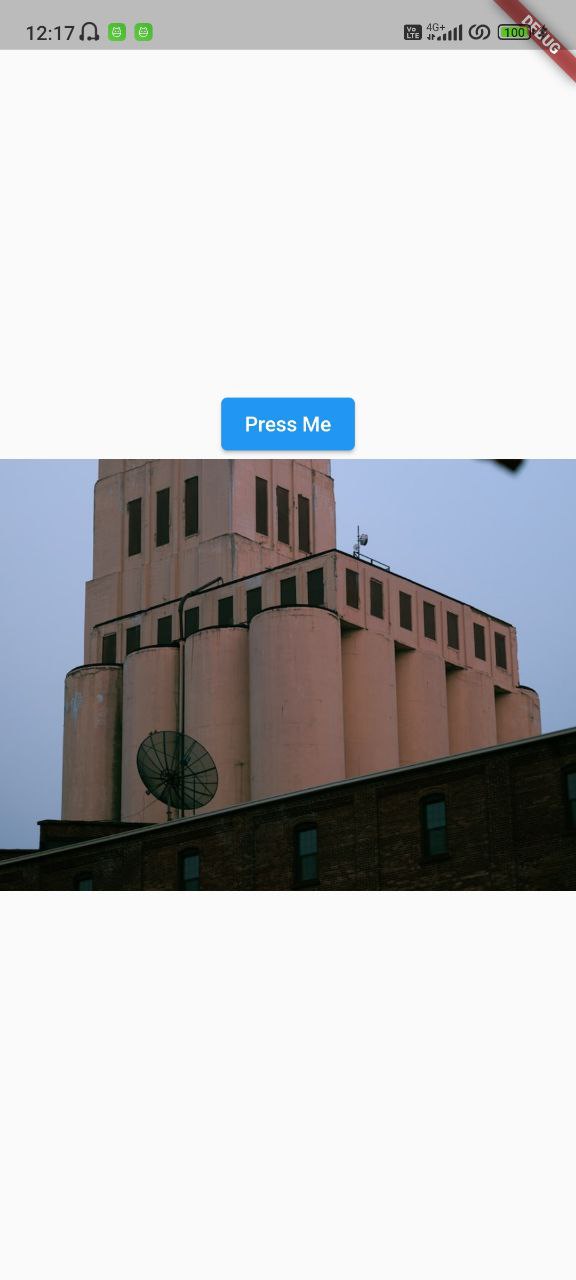


Рисунок 5. Приложение с рандомной картинкой

1. Скопировав код из методических указаний, создадим виджет списка



Рисунок 6. Виджет списка

1. Код отображения списка был изменен так, чтобы вместо бессмысленного текста отображался список книг с их обложками, названиями и авторами. Так же, при нажатии на элемент списка у него будет меняться цвет фона

decoration: BoxDecoration(  
 border:  
 const Border(bottom: BorderSide(color: Colors.*black*, width: 1)),  
 color: bgColor)

void changeColor() {  
 setState(() {  
 bgColor = Color((math.Random().nextDouble() \* 0xFFFFFF).toInt())  
 .withOpacity(1.0);  
 });  
}

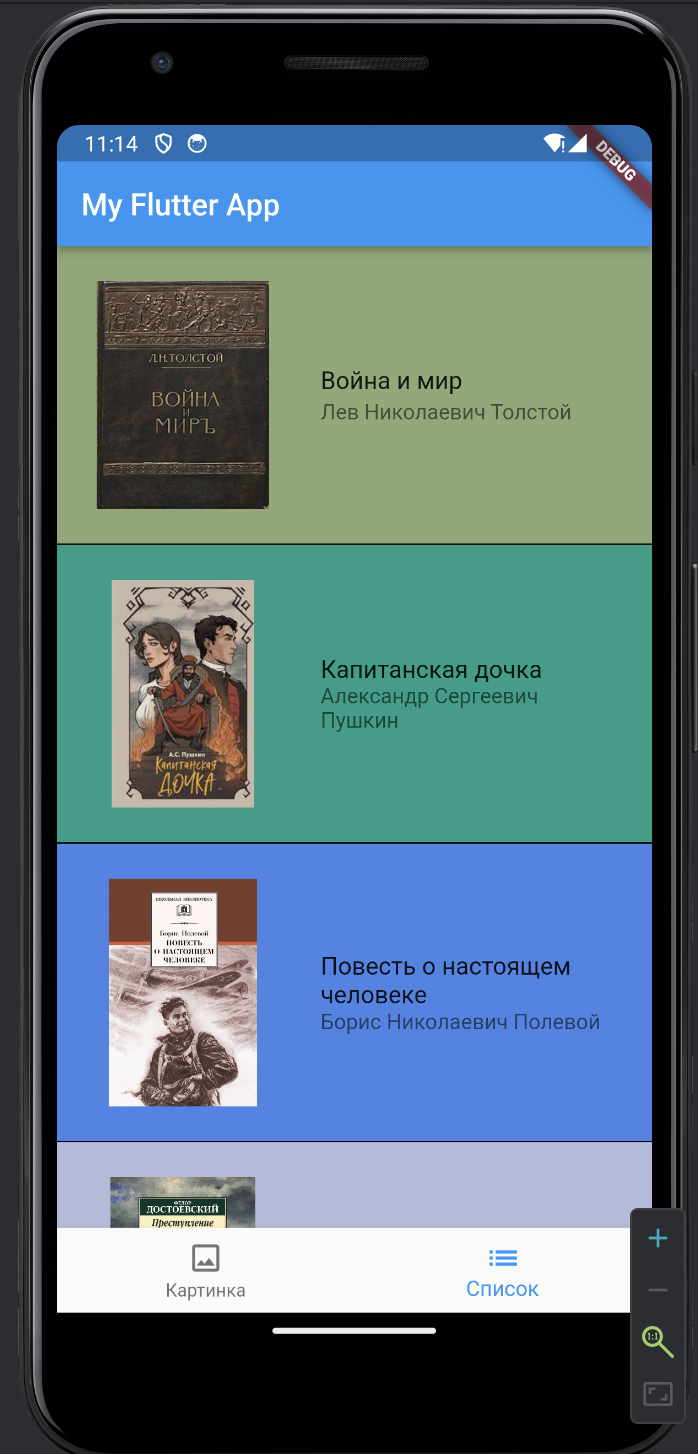


Рисунок 7. Измененный список

1. С помощью таймера реализуем загрузку новых фотографий каждые 10 секунд на странице с фотографиями

@override  
void initState() {  
 Timer.periodic(const Duration(seconds: 10), (timer) {  
 changeURL();  
 });  
 super.initState();  
}



Рисунок 8. Сменяющиеся изображения

1. Была добавлена логика удаления элемента из списка при долгом нажатии

onLongPress: filterItems(item.id),

void filterItems(int id) => () {  
 setState(() {  
 items = items.where((element) => element.id != id).toList();  
 });  
};

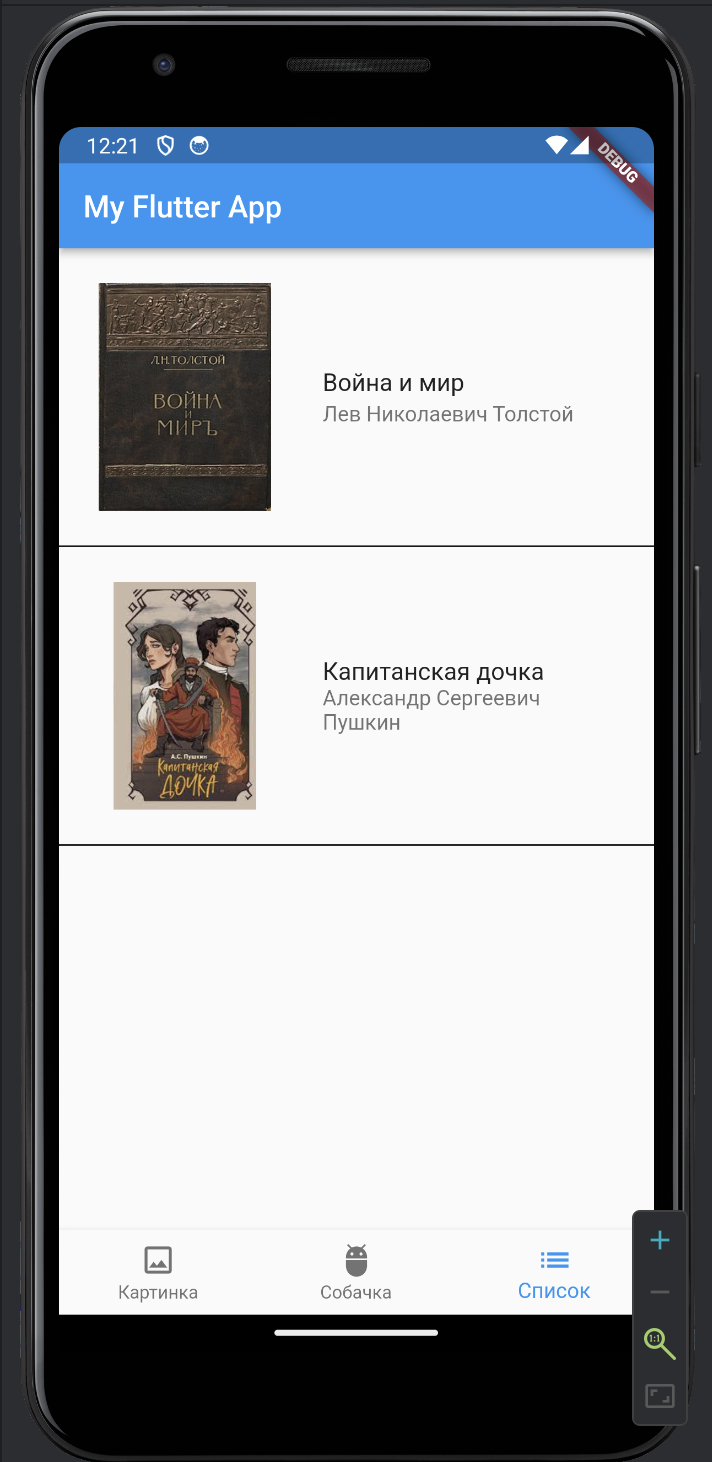


Рисунок 9. Удаление элементов списка

ВЫВОД

В ходе выполнения лабораторной работы были получены практические навыки по реализации приложения с использованием фреймворка Flutter.