

Техническая документация

Структура проекта: папки `android` и `ios` содержат в себе код и дополнительные файлы конфигурации, позволяющие связать наше приложение на `dart` с `android/ios`.

Папка `test` содержит тест для нашего проекта.

Один из важных файлов является `pubspec.yaml`, который является файлом конфигурации нашего приложения. Здесь задаются такие параметры, как имя, описание, версия приложения, версия использованного SDK.

Основная (корневая) папка – `lib`. Далее происходит разделение на `widgets` и `pages`.

`BureauApp` – основной класс, использующийся для запуска приложения. Основным методом является функция `build`, которая используется для построения и отрисовки виджета на экране.

```
class BureauApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return CupertinoApp(
      debugShowCheckedModeBanner: false,
      title: "ПРОФБЮРО",
      home: HomePage(),
    );
  }
}
```

Данный метод возвращает тип `Widget` – `CupertinoApp` (для `ios` разработки). Параметры класса: название приложения, отладочный баннер и рутовый виджет `home`.

`HomePage` – класс, который отображается в самом начале запуска приложения. Он унаследован от `StatefulWidget` для того, чтобы виджет мог изменять состояния приложения.

```
class _HomePageState extends State<HomePage>{
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.lightBlue[100],
      appBar: AppBar(
        title: Text('ПРОФБЮРО'),
        backgroundColor: Colors.blue,
      ),
      bottomNavigationBar: MainButtons(),
    );
  }
}
```

Данный метод возвращает тип `Widget` – `Scaffold`, который применяется для создания интерфейса в стиле `Material Design`. Конструктор класса `Scaffold` также имеет довольно много параметров: `backgroundColor` – цвет фона, `appBar` – верхняя панель с заголовком и возможным цветом и `bottomNavigationBar` – нижняя панель навигации, которая будет вызывать функцию `MainButtons()`.

MainButtons – класс, который будет определять кнопки переходов: “Вступить в профсоюз”, “Инициировать заседание”, “Присоединиться к заседанию”.

```
class MainButtons extends StatelessWidget {
  MainButtons({Key key}): super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: EdgeInsets.symmetric(
        horizontal: 50.0,
        vertical: 200.0,
      ),
      child: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: [
            SizedBox(
              width: 300.0,
              height: 100.0,
              child: RaisedButton(
                shape: RoundedRectangleBorder(borderRadius: new
BorderRadius.circular(50.0)),
                color: Colors.blue,
                child: Text("Вступить в профсоюз", style: new
TextStyle(fontSize: 20.0,color: Colors.white)),),
                onPressed: () {
                  Navigator.push(context, MaterialPageRoute(builder:
(context) => RequestPage()));
                },
              ),
            ),
            SizedBox(
              width: 300.0,
              height: 100.0,
              child: RaisedButton(
                shape: RoundedRectangleBorder(borderRadius: new
BorderRadius.circular(50.0)),
                color: Colors.blue,
                child: Text("Инициировать заседание", style: new
TextStyle(fontSize: 20.0,color: Colors.white)),),
                onPressed: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) =>
InitMeet()),
                  );
                },
              ),
            ),
            SizedBox(
              width: 300.0,
              height: 100.0,
              child: RaisedButton(
                shape: RoundedRectangleBorder(borderRadius: new
BorderRadius.circular(50.0)),
                color: Colors.blue,
```

```

        child: Text("Присоединиться к заседанию", style: new
TextStyle(fontSize: 19.0,color: Colors.white)),
        onPressed: () {},
      ),
    ),
  ],
),
);
}
}

```

Данный метод возвращает тип `Widget` – `Container`, необходимый для содержания только одного вложенного элемента с дополнительным функционалом. Основные параметры: `padding` – настройки отступа вложенного элемента от границ контейнера, `child` – параметр, участвующий в создании `Container`.

`Center()` – это виджет который позволяет разместить другой виджет, переданный в аргументе `child`, в центре по горизонтали и вертикали.

`Column()` – это виджет который позволяет разместить другой виджет, переданный в аргументе `child`, располагающий элементы вертикально.

`Children:[]` – принимает список виджетов. В качестве списка используются `SizedBox`.

`SizedBox` – контейнер, позволяющий дочернему виджету `child` иметь заданный размер.

`RaisedButton` – кнопка, с помощью которой будут осуществляться переходы на другие страницы.

`RequestPage` – класс, в котором происходит регистрация. Для отправки заявки во вступление в профсоюз необходимо ввести имя, фамилию, отчество и контактный телефон.

```

class RequestPage2 extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("ПРОФБЮРО"),
      ),
      backgroundColor: Colors.lightBlue[100],
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: <Widget> [
            Text("СПАСИБО, ВАМ ОБЯЗАТЕЛЬНО ПЕРЕЗВОНЯТ", style:
TextStyle(fontSize: 30.0,color: Colors.black)),
            ElevatedButton(
              onPressed: () {
                Navigator.push(context,

```

```

        MaterialPageRoute(builder: (context) => HomePage()),
    );
  },
  child: Text('НА ГЛАВНУЮ'),
),
],
)
),
);
}
}

```

```

class _RequestPageState extends State<RequestPage>{
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.lightBlue[100],
      appBar: AppBar(
        title: Text('ЗАПОЛНЕНИЕ ЗАЯВКИ'),
        backgroundColor: Colors.blue,
      ),

      body: Center(
        child: Column(
          children: <Widget> [
            TextField(
              decoration: InputDecoration(
                border: OutlineInputBorder(),
                hintText: 'ФАМИЛИЯ'
              ),
              keyboardType: TextInputType.text
            ),
            TextField(
              decoration: InputDecoration(
                border: OutlineInputBorder(),
                hintText: 'ИМЯ'
              ),
            ),
            TextField(
              decoration: InputDecoration(
                border: OutlineInputBorder(),
                hintText: 'ОТЧЕСТВО'
              ),
            ),

            TextField(
              decoration: InputDecoration(
                border: OutlineInputBorder(),
                hintText: 'ТЕЛЕФОН'
              ),
            ),

            ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (context) =>
RequestPage2()),

```

```

        );
      },
      child: Text('ОТПРАВИТЬ'),
    ),
  ],
),
),
);
}
}

```

Данный метод возвращает тип `Widget` – `Scaffold`. Конструктор класса `Scaffold` также имеет довольно много параметров: `backgroundColor` – цвет фона, `appBar` – верхняя панель с заголовком и возможным цветом и `body` – он задает основное содержимое `Scaffold` в виде другого виджета.

`TextField()` – виджет, необходимый для ввода текста пользователем.

`ElevatedButton` – виджет, представляющий кнопку, основная задача которой – реагировать на нажатия и по нажатию выполнять некоторое действие.

`onPressed()` – описывает действие, которое вызывается при нажатии.

`Navigator.push()` – метод для переключения на новый маршрут. Метод `push()` добавляет маршрут в стек маршрутов, управляемых навигатором.

`InitMeet` – класс, в котором происходит инициализация встречи. Для входа необходимо ввести логин и пароль. Далее нажать кнопку «ВОЙТИ» для дальнейших действий.

```

class _InitMeetPage extends State<InitMeet>{
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(""),
      ),
      body: Center(
        child: Column(
          children: <Widget> [
            TextField(
              decoration: InputDecoration(
                border: OutlineInputBorder(),
                hintText: 'ЛОГИН'
              ),
            ),
            TextField(
              decoration: InputDecoration(
                border: OutlineInputBorder(),
                hintText: 'ПАРОЛЬ'
              ),
            ),
            ElevatedButton(
              onPressed: () {
                Navigator.push(

```

```

        context,
        MaterialPageRoute(builder: (context) =>
CurrentPos()),
    );
    },
    child: Text('ВОЙТИ'),
  ),
]
),
),
);
}
}

```

Данный метод возвращает тип Widget – Scaffold. Конструктор класса Scaffold также имеет довольно много параметров: appBar – верхняя панель с заголовком и возможным цветом и body – он задает основное содержимое Scaffold в виде другого виджета.

Center() – это виджет который позволяет разместить другой виджет, переданный в аргументе child, в центре по горизонтали и вертикали.

Column() – это виджет который позволяет разместить другой виджет, переданный в аргументе child, располагающий элементы вертикально.

Children:[] – принимает список виджетов. В качестве списка используются TextField и ElevatedButton.

TextField() – виджет, необходимый для ввода текста пользователем.

ElevatedButton – виджет, представляющий кнопку, основная задача которой - реагировать на нажатия и по нажатию выполнять некоторое действие.