



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ

Институт компьютерных наук  
Кафедра автоматизированных систем управления

**ОТЧЕТ  
о производственной практике  
"Проектно-технологическая практика"  
в "Управление цифрового развития Липецкой области"**

Студент АС-20-1

Эрхов Д.В.

Руководитель от кафедры

Болдырихин О.В.

Руководитель ВКР  
канд. техн. наук

Кургасов В.В.

Руководитель от предприятия

Шапошникова И.А.

Липецк 2024 г.

Липецкий государственный технический университет  
Институт компьютерных наук  
Кафедра автоматизированных систем управления

**ЗАДАНИЕ НА ПРОИЗВОДСТВЕННУЮ ПРАКТИКУ**

Студенту Эрхову Дмитрию Викторовичу группы АС-21-1

Направление (специальность) 09.03.01 "Информатика и вычислительная техника"

**Изучить информацию по общей программе практики:**

1. Структуру, деятельность, бизнес-процессы, задачи отдела разработки.
2. Используемые в отделе разработки средства автоматизации.
3. Действующий порядок разработки и использования средств автоматизации: техническую политику отдела разработки в области систем автоматизации, инструменты реализации технической политики, требования к компонентам систем автоматизации и т.п.

**Выполнить индивидуальное задание** по разработке экранов информационной системы на Flutter:

1. Разработать экран изделий.
2. Разработать экран сотрудников.

Руководитель практики от кафедры

старший преподаватель Болдырихин О.В.

Задание принял к исполнению студент Эрхов Д.В.

## **Аннотация**

**С. 33, Ил. 9, Литература 6 назв. Прил. 2**

Данный документ содержит место прохождения практики, а также задание, выданное руководителем производственной практики, описание и вывод о проделанной работе.

## Оглавление

Введение.....	6
1 Краткое описание компании – места практики.....	7
1.1 Характеристика деятельности организации.....	7
2 Описание выполнения общей программы.....	8
2.1 Описание используемого программного обеспечения для выполнения практики в отделе разработки.....	8
2.1.1 Описание программы Android Studio.....	8
2.1.2 Описание Dart.....	8
2.1.3 Описание Flutter.....	8
3 Описание выполнения индивидуального задания.....	10
4 Постановка задачи ВКР.....	14
5. Описание информации, полученной на практике для выполнения ВКР.....	15
Заключение.....	16
Библиографический список.....	17
ПРИЛОЖЕНИЕ А.....	18
ПРИЛОЖЕНИЕ Б.....	25

## **Введение**

Отчет по производственной практике содержит описание "Управления цифрового развития Липецкой области" и отдела разработки, последовательность действий, совершаемая для выполнения как общей программы, так и индивидуального задания. Также подводится итог проделанной работы, каких результатов удалось достичь в ходе выполнения заданий.

## 1 Краткое описание компании – места практики

### 1.1 Характеристика деятельности организации

Управление цифрового развития Липецкой области — отраслевой исполнительный орган государственной власти Липецкой области. Создан постановлением главы администрации Липецкой области от 8 декабря 2020 года.

Продукты и сервисы помогают улучшать Липецкую область. Отдел разработки создает цифровые сервисы, приложения и сайты, которые помогают жителям нашей области легко и быстро получать нужные им услуги.

В рамках оказания платных услуг в учреждении реализуются задачи по разработке, внедрению, сопровождению и модификации сайтов для органов местного самоуправления и подведомственных им учреждений.

Отдел прохождения практики - отдел разработки.

## **2 Описание выполнения общей программы**

### **2.1 Описание используемого программного обеспечения для выполнения практики в отделе разработки**

#### **2.1.1 Описание программы Android Studio**

Android Studio – интегрированная среда разработки, созданная Google для операционных систем Windows, macOS и Linux. Предназначена для разработки мобильных приложений под платформу Android. Имеет встроенные инструменты для отладки, средства для работы с версионным контролем (Git) и анализатор производительности. Android Studio поддерживает множество плагинов, позволяя разработчикам настраивать среду под свои потребности. Распространяется бесплатно и разрабатывается как программное обеспечение с открытым исходным кодом на базе платформы IntelliJ IDEA.

#### **2.1.2 Описание Dart**

Dart – язык программирования, разработанный компанией Google для создания веб-приложений, мобильных приложений и серверного программного обеспечения. Обладает строгой статической типизацией и ориентирован на оптимизацию работы в современных браузерах и средах выполнения. Язык поддерживает объектно-ориентированное программирование с классами и миксинами. Dart активно используется в разработке с помощью фреймворка Flutter, позволяя создавать кроссплатформенные мобильные приложения с единым кодом. Также последнее время становится более популярным в серверной разработке, чем только после его выхода, несмотря на то, что основное применение в связке с Flutter. Распространяется бесплатно и разрабатывается как программное обеспечение с открытым исходным кодом.

#### **2.1.3 Описание Flutter**

Flutter – фреймворк для разработки кроссплатформенных приложений, созданный компанией Google. Позволяет создавать мобильные, веб и десктопные приложения с использованием языка Dart. У одного приложения для разных платформ один и тот же исходный код. Flutter включает в себя

полный набор виджетов для создания пользовательских интерфейсов, которые можно кастомизировать под любую платформу. Есть как набор виджетов для iOS, так и типичные Android виджеты. Обладает горячей перезагрузкой, позволяя мгновенно вносить изменения в код и видеть результаты без необходимости перезапуска проекта. Распространяется бесплатно и разрабатывается как программное обеспечение с открытым исходным кодом.

### 3 Описание выполнения индивидуального задания

Для реализации информационной системы мной была создана структура проекта представленная на рисунке 2. Проект разделен на директории экранов, виджетов, blocs для дальнейшей более удобной работы. На данный момент для выполнения задания мы будем работать с файлами employees\_screen.dart, materials\_screen.dart.

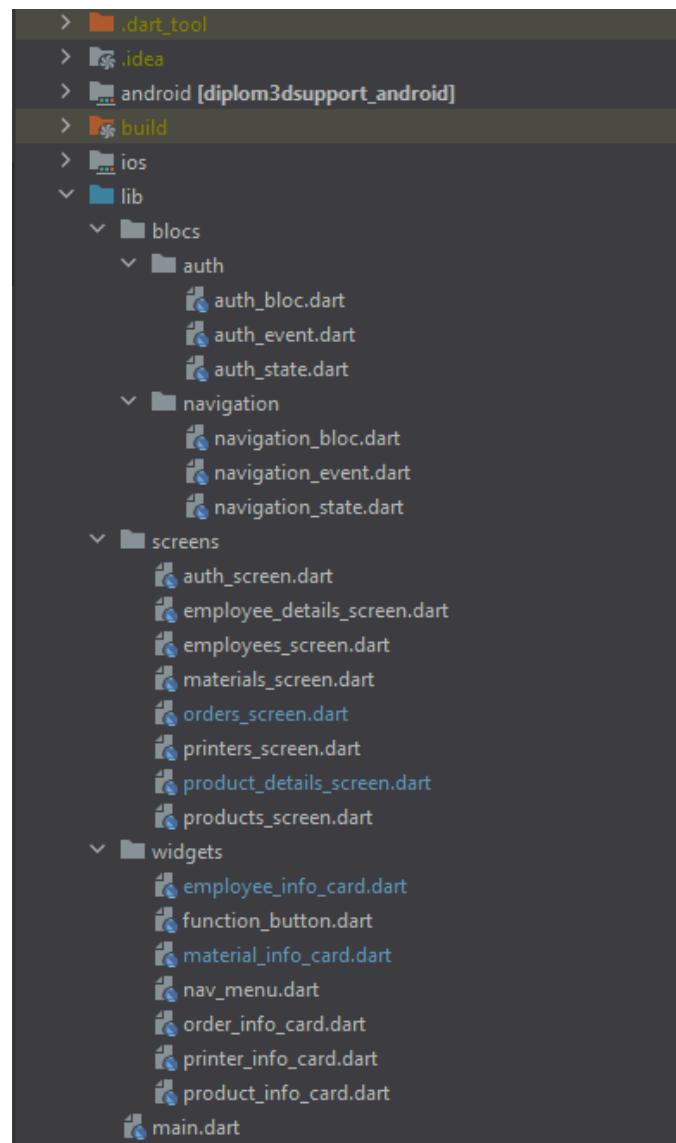


Рисунок 2 – Структура проекта

Перед продолжением работы над кодом необходимо создать макеты для экранов. Для этого опишем необходимую информацию для отображения на экранах, в соответствие с задачей системы.

Для страницы изделий необходимо отображать отдельно изделия, которые находятся на печатном столе 3Д принтера, а также блок с списком всех остальных изделий. Помимо отображения информации необходимо предусмотреть возможность фильтрации отображаемых изделий по их статусу: все, на складе, выданы, списаны. Выделим действия пользователя информационной системы с изделиями - добавить изделие, изменить статус, изменить расположение, списать изделие.

Дизайн страницы изделий с учетом ранее описанных требований представлен на рисунке 3.

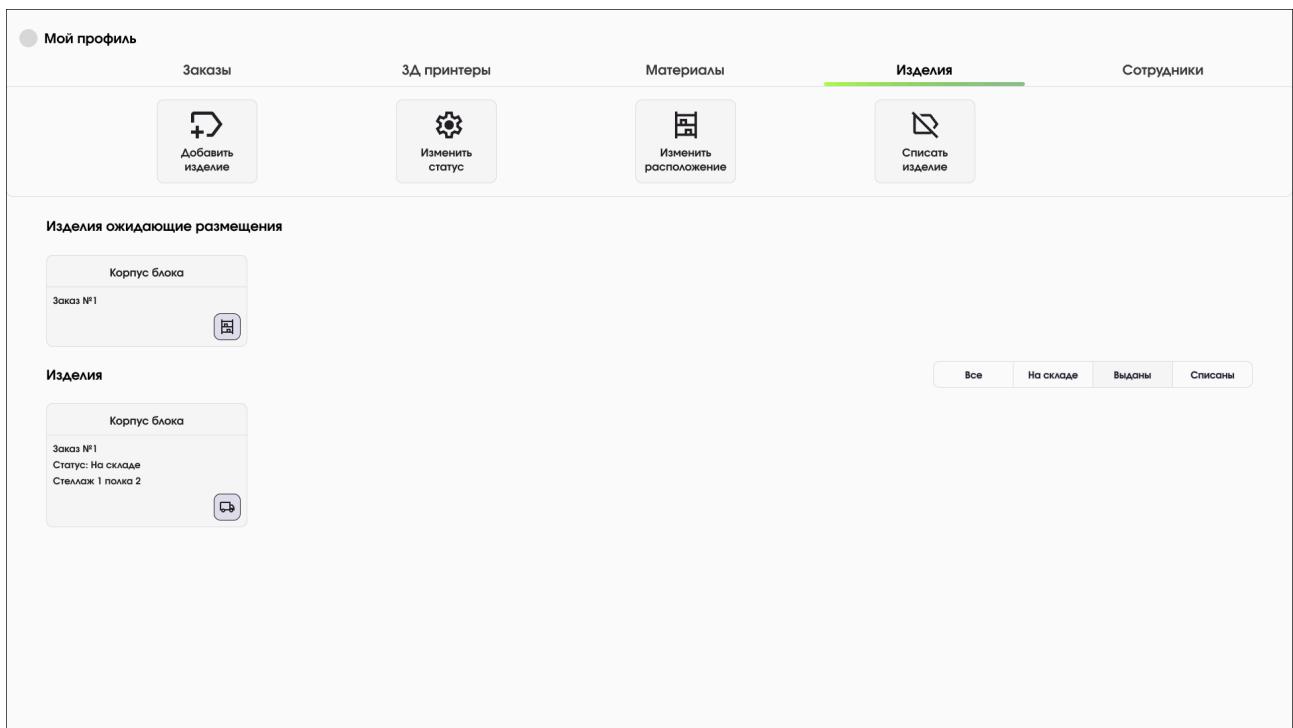


Рисунок 3 – Макет страницы изделий

Для страницы сотрудников необходимо отображать список всех сотрудников. Помимо отображения информации необходимо предусмотреть возможность фильтрации отображаемых сотрудников по их статусу: все, работающие, уволенные, в отпуске. Выделим действия администратора информационной системы с сотрудниками - добавить сотрудника, изменить статус, изменить должность. Результат представлен на рисунке 4.

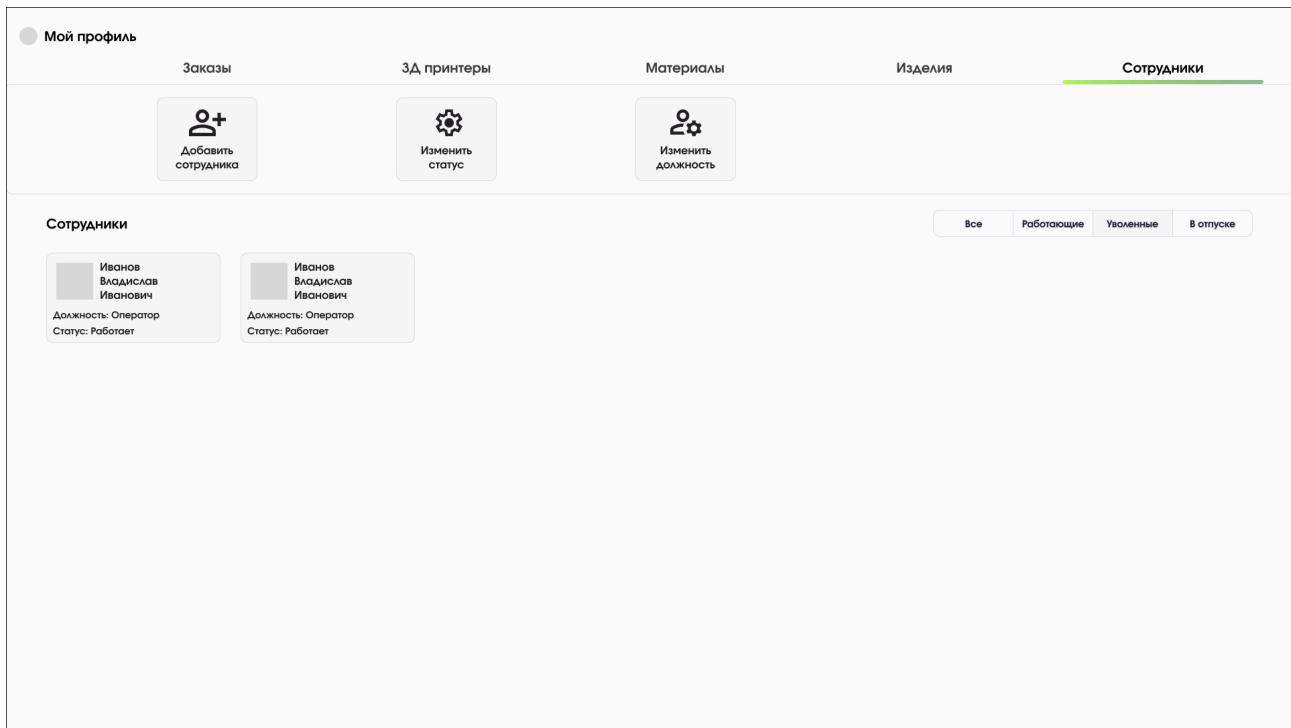


Рисунок 4 – Макет страницы сотрудников

Написание кода начнем с реализации общих элементов навигации и карточек действий на странице. Для удобства дальнейшей реализации будем сразу применять стили к элементам. Результат представлен на рисунке 5.

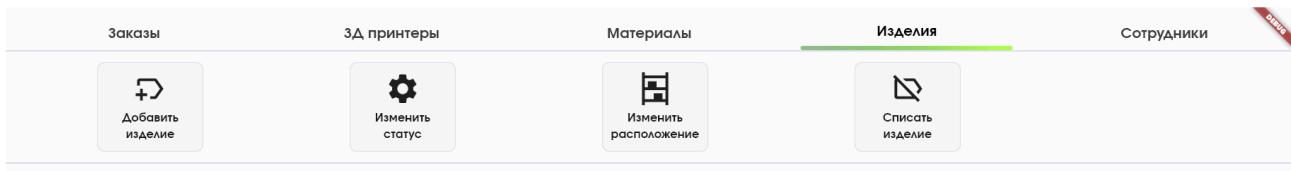


Рисунок 5 – Меню навигации и действий на странице изделий

Изменим список действий и добавим их на страницу сотрудников. Результат представлен на рисунке 6.

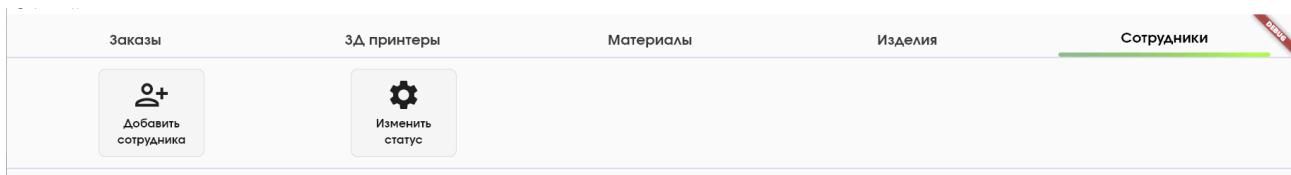


Рисунок 6 – Меню навигации и действий на странице сотрудников

Добавим блок “Изделия ожидающие размещения” и блок “Изделия” с кнопками фильтрации на страницу изделий. Результат представлен на рисунке 7.

Рисунок 7 – Страница изделий

Добавим блок “Сотрудники” с кнопками фильтрации. Результат представлен на рисунке 8.

Рисунок 8 – Страница сотрудников

#### 4 Постановка задачи ВКР

Каждый год технологии 3Д печати развиваются. Принтеры становятся более распространенными в обществе. При помощи 3Д печати у людей появляется возможности создать детали, которые невозможно купить или же цена их создания в разы меньше и при этом быстрее по срокам. Также детали из пластика в разы дешевле изделий из металла.

Количество организаций занимающихся 3Д печатью на заказ также стремительно растет. Однако стоит отметить, что большинство компаний имеют на производственных площадках не один принтер, а большое количество, чтобы успевать выполнять поступающие заказы. Как и у любого оборудования у 3Д принтеров есть свой ресурс. При беспрерывной печати на одном принтере он выйдет из строя быстрее, чем если бы заказы распределялись между несколькими имеющимися принтерами. Поэтому задача распределения заказов по принтерам на предприятиях важна.

Изучая существующие приложения были найдены только CRM системы от MyFabric, CRM Руководитель. Все названные ранее системы обладают учетом материалов, заказов, клиентов, но в них отсутствует помощь в определение оптимального оборудования для 3Д печати. Также в MyFabric готовится к выходу функция удаленного запуска печати на принтерах, но пока только в тестовом режиме, которые поддерживают не весь список производителей 3Д принтеров.

Также изучив принцип работы 3Д принтеров и различных технологий печати в качестве устройств печати были выбраны FDM принтеры, которые используют различные пластики, намотанные на катушки.

Для выпускной квалификационной работы была выбрана тема разработки информационной системы выбора оптимального оборудования для выполнения заказов студии 3Д-печати.

## 5. Описание информации, полученной на практике для выполнения ВКР

В ходе прохождения практики в отделе разработки мной были получены практические навыки разработки приложений с использованием фреймворка Flutter и языка программирования Dart. Данное техническое решение обладает встроенной поддержкой кроссплатформенности, что позволит при разработки информационной системы писать единую кодовую базу для разных платформ с дальнейшей возможностью сборки приложения под все существующие платформы.

Также в ходе практики был получен опыт практического взаимодействия с дизайн-системой от Google, которая содержит в себе множество различных иконок для приложений, использование которых не требует от разработчиков загрузки изображений иконок в отдельную папку проекта. Также используя такой подход интерфейс информационной системы будет иметь интуитивно понятный интерфейс за счет знакомых изображений иконок.

## Заключение

По итогам прохождения производственной практики в отделе разработки были получены навыки разработки Flutter приложений под Windows. Была поставлена задача ВКР

## Библиографический список

1. Методические указания к производственной практике и выпускной квалификационной работе бакалавра [Текст] – Липецк: Издательство Липецкого государственного технического университета, 2024. – 32 с.
2. Документация к Flutter [Электронный ресурс]. – Электрон. текстовые дан. – Режим доступа: <https://docs.flutter.dev/>, свободный.
3. Документация к Dart [Электронный ресурс]. – Электрон. текстовые дан. – Режим доступа: <https://dart.dev/guides>, свободный.
4. Документация по MyFabric - CRM для производства [Электронный ресурс] – Электрон. текстовые дан. – Режим доступа: [https://t.me/s/myfabric\\_crm](https://t.me/s/myfabric_crm), свободный.
5. Документация по CRM Руководитель – Электрон. текстовые дан. – Режим доступа: [https://www.rukovoditel.net.ru/database\\_templates.php?id=41](https://www.rukovoditel.net.ru/database_templates.php?id=41), свободный.
6. Habr [Электронный ресурс]. – Электрон. текстовые дан. – Режим доступа: <https://habr.com/ru>, свободный

## ПРИЛОЖЕНИЕ А

(Код приложения)

```
import 'package:flutter/material.dart';
import '../widgets/employee_info_card.dart';
import '../widgets/function_button.dart';
```

```
class EmployeesScreen extends StatefulWidget {
```

```
    @override
```

```
    _EmployeesScreenState createState() => _EmployeesScreenState();
```

```
}
```

```
class _EmployeesScreenState extends State<EmployeesScreen> {
```

```
    int selectedSegment = 0; // Выбранный сегмент: 0 - Все, 1 - Работающие, 2
```

- Уволенные, 3 - В отпуске

```
    final List<EmployeeInfoCard> allEmployees = [
```

```
        EmployeeInfoCard(
```

```
            lastName: "Иванов",
```

```
            firstName: "Иван",
```

```
            patronymic: "Иванович",
```

```
            position: "Инженер",
```

```
            status: "Работающий",
```

```
        ),
```

```
        EmployeeInfoCard(
```

```
            lastName: "Петров",
```

```
            firstName: "Петр",
```

```
            patronymic: "Петрович",
```

```
            position: "Техник",
```

```
            status: "Уволенный",
```

```
        ),  
        EmployeeInfoCard(  
            lastName: "Сидоров",  
            firstName: "Сидор",  
            patronymic: "Сидорович",  
            position: "Менеджер",  
            status: "В отпуске",  
        ),  
        EmployeeInfoCard(  
            lastName: "Кузнецова",  
            firstName: "Анна",  
            patronymic: "Ивановна",  
            position: "Секретарь",  
            status: "Работающий",  
        ),  
    ];  
  
    @override  
    Widget build(BuildContext context) {  
        // Фильтруем сотрудников в зависимости от выбранного фильтра  
        List<EmployeeInfoCard> displayedEmployees;  
        if (selectedSegment == 0) {  
            displayedEmployees = allEmployees; // Все сотрудники  
        } else if (selectedSegment == 1) {  
            displayedEmployees = allEmployees.where((employee) => employee.status  
            == "Работающий").toList();  
        } else if (selectedSegment == 2) {  
            displayedEmployees = allEmployees.where((employee) => employee.status  
            == "Уволенный").toList();  
    }
```

```
    } else {
        displayedEmployees = allEmployees.where((employee) => employee.status
== "В отпуске").toList();
    }

    return SingleChildScrollView(
        child: Column(
            mainAxisAlignment: MainAxisAlignment.start,
            children: [
                Column(
                    children: [
                        Container(
                            width: MediaQuery.of(context).size.width,
                            height: 2,
                            color: Color(0xFFDFDCED),
                        ),
                        Padding(
                            padding: const EdgeInsets.only(left: 130),
                            child: Row(
                                mainAxisAlignment: MainAxisAlignment.start,
                                children: [
                                    FunctionButton(
                                        title: 'Добавить сотрудника',
                                        isIcon: true,
                                        nameIcon: Icon(Icons.person_add_alt, size: 50,),
                                        onPressed: () { print('Добавить сотрудника'); },
                                    ),
                                    SizedBox(width: 207),
                                    FunctionButton(

```

```
        isIcon: true,  
        nameIcon: Icon(Icons.settings, size: 50,),  
        title: 'Изменить статус',  
        onPressed: () { print('Изменить статус'); },  
    ),  
],  
(  
),  
Container(  
    width: MediaQuery.of(context).size.width,  
    height: 2,  
    color: Color(0xFFDFDCED),  
),  
],  
(  
),  
Padding(  
    padding: const EdgeInsets.only(top: 30, left: 60, right: 60),  
    child: Row(  
        mainAxisAlignment: MainAxisAlignment.spaceBetween,  
        children: [  
            Text(  
                'Сотрудники',  
                style: TextStyle(  
                    fontSize: 20,  
                    fontFamily: "Evolventa",  
                    fontWeight: FontWeight.bold,  
                ),  
            ),  
            ToggleButtons(  

```

```
color: Color(0xFF3D3D3D),  
selectedColor: Colors.black,  
fillColor: Color(0xFFFFF6F5),  
borderColor: Color(0xFFDFDCED),  
selectedBorderColor: Color(0xFFDFDCED),  
borderWidth: 1,  
borderRadius: BorderRadius.circular(5),  
children: [  
    Container(  
        width: 120,  
        alignment: Alignment.center,  
        child: Text(  
            'Bce',  
            style: TextStyle(  
                fontSize: 14,  
                fontFamily: "Evolventa",  
                fontWeight: FontWeight.bold,  
            ),  
        ),  
    ),  
    Container(  
        width: 120,  
        alignment: Alignment.center,  
        child: Text(  
            'Работающие',  
            style: TextStyle( fontSize: 14, fontFamily: "Evolventa",  
                fontWeight: FontWeight.bold, ),  
        ),  
    ),
```

```
Container(  
    width: 120,  
    alignment: Alignment.center,  
    child: Text(  
        'Уволенные',  
        style: TextStyle(  
            fontSize: 14, fontFamily: "Evolventa",  
            fontWeight: FontWeight.bold,  
        ),  
    ),  
,  
    Container(  
        width: 120,  
        alignment: Alignment.center,  
        child: Text(  
            'В отпуске',  
            style: TextStyle(  
                fontSize: 14, fontFamily: "Evolventa",  
                fontWeight: FontWeight.bold,  
            ),  
        ),  
    ),  
,  
],  
isSelected: [  
    selectedSegment == 0,  
    selectedSegment == 1,  
    selectedSegment == 2,  
    selectedSegment == 3,  
],
```

```

 onPressed: (int newIndex) {
    setState(() {
        selectedSegment = newIndex;
    });
},
),
),
SizedBox(height: 5),
Padding(
    padding: const EdgeInsets.only(left: 60, right: 60),
    child: GridView.builder(
        shrinkWrap: true,
        physics: NeverScrollableScrollPhysics(),
        gridDelegate: SliverGridDelegateWithMaxCrossAxisExtent(
            maxCrossAxisExtent: 305,
            mainAxisExtent: 200,
            crossAxisSpacing: 20,
            mainAxisSpacing: 20,
            childAspectRatio: 300 / 190,
        ),
        itemCount: displayedEmployees.length,
        itemBuilder: (context, index) {
            return displayedEmployees[index];
        },
),
),
],
),
);
}
}

```

## ПРИЛОЖЕНИЕ Б

### (Код приложения)

```
import 'package:flutter/material.dart';
import '../widgets/function_button.dart';
import '../widgets/material_info_card.dart';

class MaterialsScreen extends StatefulWidget {
    @override
    _MaterialsScreenState createState() => _MaterialsScreenState();
}

class _MaterialsScreenState extends State<MaterialsScreen> {
    int selectedSegment = 0; // Выбранный сегмент (0 - все, 1 - установленные,
    // 2 - на складе, 3 - списанные)

    final List<MaterialInfoCard> allMaterials = [
        MaterialInfoCard(
            plasticName: 'PLA пластик 1',
            weight: '500',
            locationType: 'Принтер',
            locationValue1: '1',
            isPrinter: true,
            isWrittenOff: false,
        ),
        MaterialInfoCard(
            plasticName: 'ABS пластик 2',
            weight: '750',
            locationType: 'Принтер',
            locationValue1: '2',
        ),
    ];
}
```

```
    isPrinter: true,  
    isWrittenOff: false,  
,  
MaterialInfoCard(  
    plasticName: 'PETG пластик 3',  
    weight: '600',  
    locationType: 'Стеллаж',  
    locationValue1: '1',  
    locationValue2: 'A',  
    isPrinter: false,  
    isWrittenOff: false,  
,  
MaterialInfoCard(  
    plasticName: 'PLA пластик 42',  
    weight: '500',  
    locationType: 'Принтер',  
    locationValue1: '3',  
    isPrinter: false,  
    isWrittenOff: true, // Списанный материал  
,  
MaterialInfoCard(  
    plasticName: 'ABS пластик 5',  
    weight: '700',  
    locationType: 'Стеллаж',  
    locationValue1: '2',  
    locationValue2: 'B',  
    isPrinter: false,  
    isWrittenOff: true, // Списанный материал  
,
```

```
MaterialInfoCard(
    plasticName: 'PLA пластик 4',
    weight: '500',
    locationType: 'Принтер',
    locationValue1: '3',
    isPrinter: true,
    isWrittenOff: false, // Списанный материал
),
MaterialInfoCard(
    plasticName: 'ABS пластик 5',
    weight: '700',
    locationType: 'Стеллаж',
    locationValue1: '2',
    locationValue2: 'B',
    isPrinter: false,
    isWrittenOff: true, // Списанный материал
),
];
}

@Override
Widget build(BuildContext context) {
    // Определяем, какие карточки показывать в зависимости от выбранного
    сегмента в разделе "Материалы"
    List<MaterialInfoCard> displayedMaterials;
    if (selectedSegment == 0) {
        displayedMaterials = allMaterials; // Все материалы
    } else if (selectedSegment == 1) {
        displayedMaterials = allMaterials.where((material) =>
material.isPrinter).toList(); // Установленные на принтерах
```

```
        } else if (selectedSegment == 2) {
            displayedMaterials = allMaterials.where((material) => !material.isPrinter
&& !material.isWrittenOff).toList(); // На складе, не списанные
        } else {
            displayedMaterials = allMaterials.where((material) =>
material.isWrittenOff).toList(); // Списанные материалы
        }

        return SingleChildScrollView(
            child: Column(
                mainAxisAlignment: MainAxisAlignment.start,
                children: [
                    Column(
                        children: [
                            Container(
                                width: MediaQuery.of(context).size.width,
                                height: 2,
                                color: Color(0xFFDFDCED),
                            ),
                            Padding(
                                padding: const EdgeInsets.only(left: 130),
                                child: Row(
                                    mainAxisAlignment: MainAxisAlignment.start,
                                    children: [
                                        FunctionButton(
                                            title: 'Добавить материал',
                                            isIcon: true,
                                            nameIcon: Icon(Icons.new_label_outlined, size: 50,),
                                            onPressed: () { print('Материал'); },
                                        )
                                    ],
                                ),
                            ),
                        ],
                    ),
                ],
            ),
        );
    }
}
```

```
        ),  
        SizedBox(width: 207),  
        FunctionButton(  
            isIcon: true,  
            nameIcon: Icon(Icons.build_outlined, size: 50,),  
            title: 'Заменить у 3Д принтера',  
            onPressed: () { print('Материал'); },  
        ),  
        SizedBox(width: 207),  
        FunctionButton(  
            isIcon: true,  
            nameIcon: Icon(Icons.shelves, size: 50,),  
            title: 'Изменить расположение',  
            onPressed: () { print('Материал'); },  
        ),  
        SizedBox(width: 207),  
        FunctionButton(  
            title: 'Списать материал',  
            isIcon: true,  
            nameIcon: Icon(Icons.label_off_outlined, size: 50,),  
            onPressed: () { print('Материал'); },  
        ),  
    ],  
,  
),  
Container(  
    width: MediaQuery.of(context).size.width,  
    height: 2,  
    color: Color(0xFFDFDCED),
```

```
        ),  
    ],  
),  
Padding(  
    padding: const EdgeInsets.only(top: 30, bottom: 10, left: 60, right: 60),  
    child: Text(  
        'Материалы установленные на 3Д принтерах',  
        style: TextStyle(  
            fontSize: 20,  
            fontFamily: "Evolventa",  
            fontWeight: FontWeight.bold,  
        ),  
        textAlign: TextAlign.start,  
    ),  
,  
Padding(  
    padding: const EdgeInsets.only(left: 60, right: 60),  
    child: GridView.builder(  
        shrinkWrap: true,  
        physics: NeverScrollableScrollPhysics(),  
        gridDelegate: SliverGridDelegateWithMaxCrossAxisExtent(  
            maxCrossAxisExtent: 205, // Максимальная ширина карточки  
            mainAxisExtent: 170,  
            crossAxisSpacing: 20, // Расстояние между колонками  
            mainAxisSpacing: 20, // Расстояние между строками  
            childAspectRatio: 205 / 165, // Соотношение сторон карточки  
        ),  
        itemCount: allMaterials.where((material) => material.isPrinter).length,  
        itemBuilder: (context, index) {
```

```
// Отображаем только материалы, установленные на принтерах
        return allMaterials.where((material) =>
material.isPrinter).toList()[index];
    },
),
),
Padding(
    padding: const EdgeInsets.only(top: 30, left: 60, right: 60),
    child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
            Text(
                'Материалы',
                style: TextStyle(
                    fontSize: 20,
                    fontFamily: "Evolventa",
                    fontWeight: FontWeight.bold,
                ),
            ),
            ToggleButtons(
                color: Color(0xFF3D3D3D), // Цвет текста неактивных кнопок
                selectedColor: Colors.black, // Цвет текста активной кнопки
                fillColor: Color(0xFFF5F6F5), // Цвет фона кнопок
                borderColor: Color(0xFFDFDCED), // Цвет границы кнопок
                selectedBorderColor: Color(0xFFDFDCED), // Цвет границы
активной кнопки
                borderWidth: 1, // Ширина границы
                borderRadius: BorderRadius.circular(5), // Закругление углов
                children: [

```

```
Container(  
    width: 120, // Устанавливаем одинаковую ширину для всех  
    КНОПОК  
        alignment: Alignment.center,  
        child: Text('Bce', style: TextStyle(  
            fontSize: 14,  
            fontFamily: "Evolventa",  
            fontWeight: FontWeight.bold,  
        )),  
    ),  
    Container(  
        width: 120,  
        alignment: Alignment.center,  
        child: Text('Установленные', style: TextStyle(  
            fontSize: 14,  
            fontFamily: "Evolventa",  
            fontWeight: FontWeight.bold,  
        )),  
    ),  
    Container(  
        width: 120,  
        alignment: Alignment.center,  
        child: Text('На складе', style: TextStyle(  
            fontSize: 14,  
            fontFamily: "Evolventa",  
            fontWeight: FontWeight.bold,  
        )),  
    ),  
    Container(  
        width: 120,
```

```
        width: 120,  
        alignment: Alignment.center,  
        child: Text('Списанные', style: TextStyle(  
            fontSize: 14,  
            fontFamily: "Evolventa",  
            fontWeight: FontWeight.bold,  
        )),  
    ),  
],  
isSelected: [  
    selectedSegment == 0,  
    selectedSegment == 1,  
    selectedSegment == 2,  
    selectedSegment == 3,  
],  
onPressed: (int newIndex) {  
    setState(() {  
        selectedSegment = newIndex;  
    });  
},  
),  
],  
),  
),  
SizedBox(height: 5),  
Padding(  
    padding: const EdgeInsets.only(left: 60, right: 60),  
    child: GridView.builder(  
        shrinkWrap: true,
```

```
physics: NeverScrollableScrollPhysics(),  
gridDelegate: SliverGridDelegateWithMaxCrossAxisExtent(  
    maxCrossAxisExtent: 205, // Максимальная ширина карточки  
    mainAxisExtent: 197,  
    crossAxisSpacing: 20, // Расстояние между колонками  
    mainAxisSpacing: 20, // Расстояние между строками  
    childAspectRatio: 205 / 165, // Соотношение сторон карточки  
,  
itemCount: displayedMaterials.length,  
itemBuilder: (context, index) {  
    return displayedMaterials[index];  
,  
,  
),  
],  
,  
);  
}  
}
```