|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**Лабораторная работа №8**

по курсу «Разработка мобильных приложений».

«Разработка smtp мейлера»

Выполнил:

студент группы ИУ9-71Б

Митрошкин Алексей

Проверил:

Посевин Д.П.

Москва, 2024

1. **Цель**

Разработать приложение, отправляющее электронные письма на указанный адрес по протоколу smtp.

1. **Задание**

* Создание smtp аккаунта Google.
* Разработка приложения.

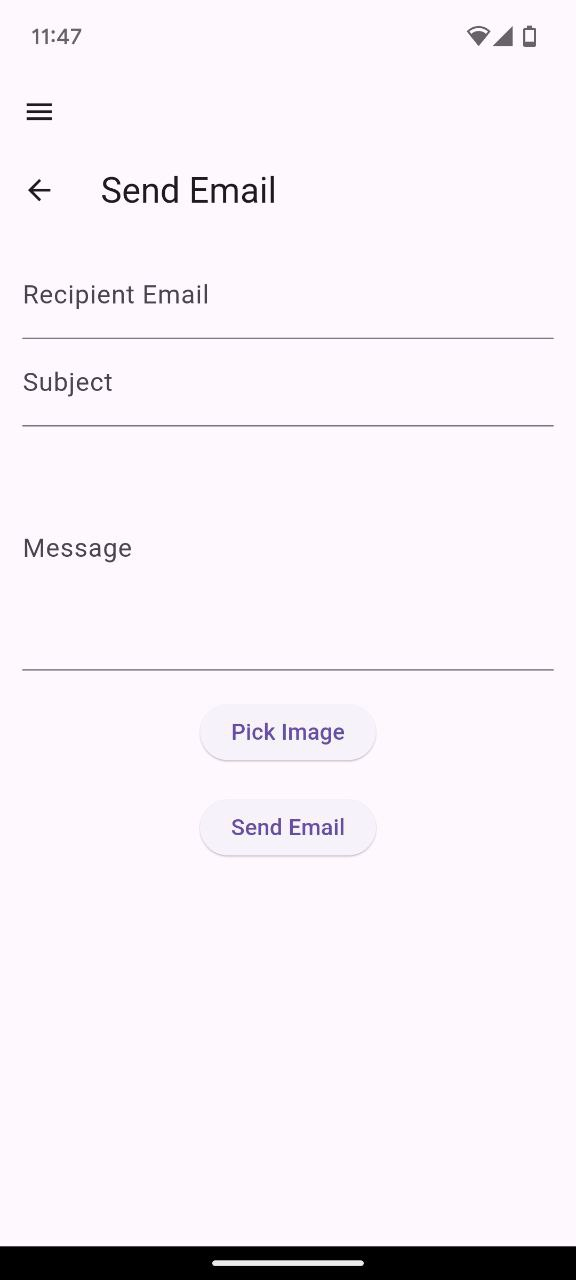
1. **Реализация**

Исходный код приведён в Листинге 1.

Листинг 1

|  |
| --- |
| import 'dart:io';  import 'dart:convert';  import 'package:flutter/material.dart';  import 'package:image\_picker/image\_picker.dart';  import 'package:mailer/mailer.dart';  import 'package:mailer/smtp\_server.dart';  class SmptScreen extends StatefulWidget {    const SmptScreen({super.key});    @override    State<SmptScreen> createState() => SmptScreenState();  }  class SmptScreenState extends State<SmptScreen> {    final TextEditingController \_recipientController = TextEditingController();    final TextEditingController \_subjectController = TextEditingController();    final TextEditingController \_bodyController = TextEditingController();    File? \_image;    Future<void> \_pickImage() async {      final picker = ImagePicker();      final pickedFile = await picker.pickImage(source: ImageSource.gallery);      if (pickedFile != null) {        setState(() {          \_image = File(pickedFile.path);        });      }    }    Future<void> \_sendEmail() async {      final smtpServer = gmail('alexey.mitroshkin@gmail.com', 'wtsf ggus zxry icwj');        final emailBody = '''        <html>          <body>            <p>${\_bodyController.text}</p>            <img src="cid:image\_cid" alt="Image" >          </body>        </html>      ''';      final message = Message()        ..from = Address('alexey.mitroshkin@gmail.com', 'Alexey Mitroshkin')        ..recipients.add(\_recipientController.text)        ..subject = \_subjectController.text        ..html = emailBody;      if (\_image != null) {        final imageBytes = await \_image!.readAsBytes();        final attachment = FileAttachment(\_image!)          ..location = Location.inline          ..cid = 'image\_cid';        message.attachments.add(attachment);      }      try {        final sendReport = await send(message, smtpServer);        ScaffoldMessenger.of(context).showSnackBar(          SnackBar(content: Text('Email sent successfully: ${sendReport.toString()}')),        );      } catch (e) {        ScaffoldMessenger.of(context).showSnackBar(          SnackBar(content: Text('Failed to send email: $e')),        );      }    }    @override    Widget build(BuildContext context) {      return Scaffold(        appBar: AppBar(          title: const Text('Send Email'),        ),        body: Padding(          padding: const EdgeInsets.all(16.0),          child: Column(            children: [              TextField(                controller: \_recipientController,                decoration: const InputDecoration(labelText: 'Recipient Email'),              ),              TextField(                controller: \_subjectController,                decoration: const InputDecoration(labelText: 'Subject'),              ),              TextField(                controller: \_bodyController,                decoration: const InputDecoration(labelText: 'Message'),                maxLines: 5,              ),              const SizedBox(height: 20),              ElevatedButton(                onPressed: \_pickImage,                child: const Text('Pick Image'),              ),              const SizedBox(height: 20),              ElevatedButton(                onPressed: \_sendEmail,                child: const Text('Send Email'),              ),              if (\_image != null)                Padding(                  padding: const EdgeInsets.only(top: 20),                  child: Image.file(\_image!, height: 100, width: 100),                ),            ],          ),        ),      );    }    @override    void dispose() {      \_recipientController.dispose();      \_subjectController.dispose();      \_bodyController.dispose();      super.dispose();    }  } |

На рисунках 1-2 приведёны результаты работы программы



*Рисунок 1*

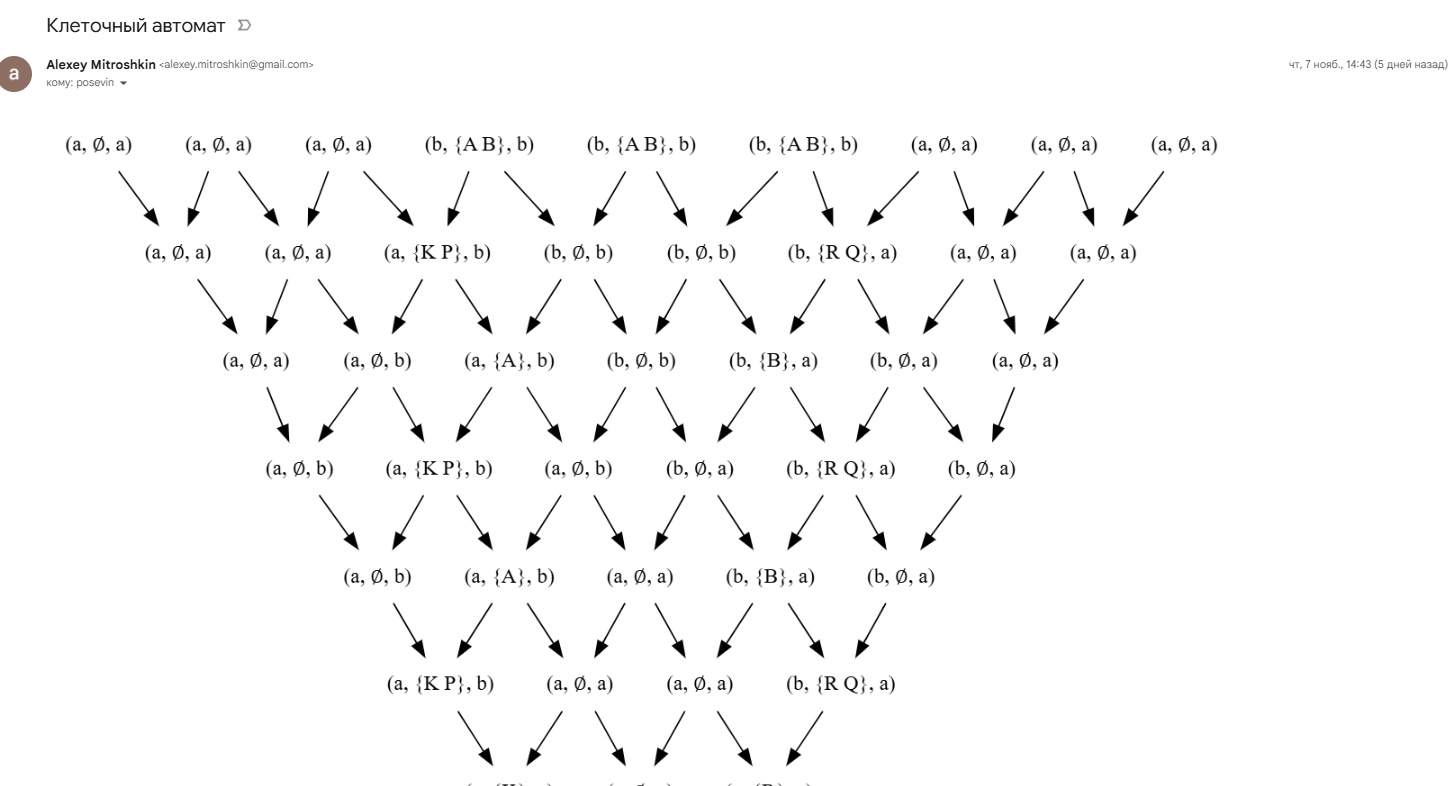


Рисунок 2