package br.com.fiap.applocaweb.screens

import androidx.compose.foundation.BorderStroke

import androidx.compose.foundation.background

import androidx.compose.foundation.layout.Arrangement

import androidx.compose.foundation.layout.Column

import androidx.compose.foundation.layout.Row

import androidx.compose.foundation.layout.Spacer

import androidx.compose.foundation.layout.fillMaxHeight

import androidx.compose.foundation.layout.fillMaxWidth

import androidx.compose.foundation.layout.height

import androidx.compose.foundation.layout.offset

import androidx.compose.foundation.layout.padding

import androidx.compose.foundation.shape.RoundedCornerShape

import androidx.compose.material.icons.Icons

import androidx.compose.material.icons.filled.MoreVert

import androidx.compose.material3.Button

import androidx.compose.material3.ButtonDefaults

import androidx.compose.material3.Card

import androidx.compose.material3.ExperimentalMaterial3Api

import androidx.compose.material3.Icon

import androidx.compose.material3.OutlinedButton

import androidx.compose.material3.Text

import androidx.compose.material3.TextField

import androidx.compose.runtime.Composable

import androidx.compose.ui.Modifier

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.navigation.NavController

import br.com.fiap.applocaweb.R

@Composable

fun SendEmailScreen(navController: NavController) {

Column(modifier = Modifier

.fillMaxHeight()

) {

Card(

shape = RoundedCornerShape(topEnd = 15.dp, topStart = 15.dp),

modifier = Modifier

.fillMaxWidth()

.background(Color(0xFFF5EDF0))

//.height(700.dp)

.offset(y = 30.dp))

{

Column(modifier = Modifier

.fillMaxWidth()

.fillMaxHeight()

) {

Row (horizontalArrangement = Arrangement.Center, modifier = Modifier

.fillMaxWidth()

){

}

Row(

horizontalArrangement = Arrangement.SpaceBetween,

modifier = Modifier.fillMaxWidth()

) {

OutlinedButton(

border = BorderStroke(0.dp, Color.Transparent),

onClick = { /\*TODO\*/ }

) {

Text(

text = "Salvar Rascunho")

}

OutlinedButton(

border = BorderStroke(0.dp, Color.Transparent),

onClick = { navController.navigate("caixaentrada") }

) {

Text(

text = "Cancelar")

}

}

Spacer(modifier = Modifier.padding(40.dp))

TextField(

modifier = Modifier

.fillMaxWidth()

.padding(10.dp)

.offset(y = 30.dp),

value = "Para", onValueChange = {})

//Spacer(modifier = Modifier.height(10.dp))

TextField(

modifier = Modifier

.fillMaxWidth()

.padding(10.dp)

.offset(y = 30.dp),

value = "Assunto", onValueChange = {})

TextField(

modifier = Modifier

.fillMaxWidth()

.padding(10.dp)

.offset(y = 50.dp)

.height(230.dp)

,

value = "Mensagem", onValueChange = {})

Spacer(modifier = Modifier.padding(40.dp))

Row(modifier = Modifier.fillMaxWidth(), horizontalArrangement = Arrangement.SpaceEvenly) {

Button(onClick = { /\*TODO\*/ },

colors = ButtonDefaults.buttonColors(containerColor = Color(0xFFFD5151))

) {

Icon(

painter = painterResource(id = R.drawable.baseline\_format\_color\_text\_24),

contentDescription = "formatação de texto"

)

}

Button(onClick = { /\*TODO\*/ },

colors = ButtonDefaults.buttonColors(containerColor = Color(0xFFFD5151))

) {

Icon(

painter = painterResource(id = R.drawable.baseline\_emoji\_emotions\_24),

contentDescription = "Emoji"

)

}

Button(onClick = { /\*TODO\*/ },

colors = ButtonDefaults.buttonColors(containerColor = Color(0xFFFD5151))) {

Icon(

painter = painterResource(id = R.drawable.baseline\_attach\_file\_24),

contentDescription = "anexar arquivo"

)

}

Button(onClick = { /\*TODO\*/ },

colors = ButtonDefaults.buttonColors(containerColor = Color(0xFFFD5151))

) {

Icon(

painter = painterResource(id = R.drawable.baseline\_add\_location\_24),

contentDescription = "adicionar localização"

)

}

Button(onClick = { /\*TODO\*/ },

colors = ButtonDefaults.buttonColors(containerColor = Color(0xFFFD5151))) {

Icon(imageVector = Icons.Default.MoreVert, contentDescription = "Fechar Email")

}

}

Row (modifier = Modifier

.fillMaxWidth()

.fillMaxHeight()

){

Button(modifier = Modifier

.fillMaxWidth()

.padding(5.dp), onClick = { /\*TODO\*/ },

colors = ButtonDefaults.buttonColors(containerColor = Color(0xFF5C77AC))) {

Text(text = "Enviar Email")

}

}

}

}

}

}

//@Preview(showSystemUi = true, showBackground = true)

//@Composable

//fun SendEmailScreenPreview() {

// SendEmailScreen()

//}