package br.com.fiap.applocaweb.screens

import androidx.compose.foundation.BorderStroke

import androidx.compose.foundation.background

import androidx.compose.foundation.border

import androidx.compose.foundation.clickable

import androidx.compose.foundation.layout.\*

import androidx.compose.foundation.rememberScrollState

import androidx.compose.foundation.shape.RoundedCornerShape

import androidx.compose.foundation.verticalScroll

import androidx.compose.material.icons.Icons

import androidx.compose.material.icons.filled.Delete

import androidx.compose.material.icons.filled.Search

import androidx.compose.material3.Button

import androidx.compose.material3.ButtonDefaults

import androidx.compose.material3.Card

import androidx.compose.material3.CardDefaults

import androidx.compose.material3.ExperimentalMaterial3Api

import androidx.compose.material3.Icon

import androidx.compose.material3.Text

import androidx.compose.material3.TextField

import androidx.compose.runtime.Composable

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.text.font.FontWeight

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

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import androidx.navigation.NavController

import br.com.fiap.applocaweb.R

@OptIn(ExperimentalMaterial3Api::class)

@Composable

fun GmailRowItem2(navController: NavController) {

Column(modifier = Modifier.fillMaxSize()) {

// Column{

Card(

modifier = Modifier

.height(200.dp)

.fillMaxWidth(),

colors = CardDefaults.cardColors(containerColor = Color(0xFF5C77AC)),

shape = RoundedCornerShape(bottomEnd = 19.dp, bottomStart = 19.dp),

elevation = CardDefaults.cardElevation(1.dp)

) {

Spacer(modifier = Modifier.padding(10.dp))

TextField(maxLines = 1,

modifier = Modifier

.fillMaxWidth()

.padding(22.dp), shape = RoundedCornerShape(5.dp),

value = "Pesquisar", onValueChange = {},

trailingIcon = {

Icon(imageVector = Icons.Default.Search, contentDescription = "Buscar")

}

)

Row(

horizontalArrangement = Arrangement.SpaceEvenly,

verticalAlignment = Alignment.Bottom,

modifier = Modifier

.fillMaxWidth()

.fillMaxHeight()

.background(Color(0xFF5C77AC))

) {

Button(

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

colors = ButtonDefaults.buttonColors(

containerColor = Color(0xFF5C77AC)

)

) {

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

Icon(

painter = painterResource(id = R.drawable.baseline\_all\_inbox\_24),

contentDescription = "Caixa de entrada"

)

}

Button(

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

colors = ButtonDefaults.buttonColors(

containerColor = Color(0xFF5C77AC)

)

) {

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

Icon(

painter = painterResource(id = R.drawable.baseline\_star\_border\_24),

contentDescription = "Favoritos"

)

}

Button(

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

colors = ButtonDefaults.buttonColors(

containerColor = Color(0xFF5C77AC)

)

) {

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

Icon(

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

contentDescription = "Enviados"

)

}

Button(

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

colors = ButtonDefaults.buttonColors(

containerColor = Color(0xFF5C77AC)

)

) {

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

Icon(

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

contentDescription = "Caixa de entrada"

)

}

Button(

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

colors = ButtonDefaults.buttonColors(

containerColor = Color(0xFF5C77AC)

)

) {

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

Icon(

imageVector = Icons.Default.Delete,

contentDescription = "Caixa de entrada"

)

}

}

}

Column(modifier = Modifier.height(500.dp)) {

Row(

modifier = Modifier

.fillMaxWidth()

.clickable { }

.verticalScroll(rememberScrollState())

) {

for (i in 0..10){

EmailCard()

Spacer(modifier = Modifier.height(4.dp))

}

}

}

Card(

modifier = Modifier.padding(vertical = 10.dp),

colors = CardDefaults.cardColors(containerColor = Color(0xFF5C77AC)),

shape = RoundedCornerShape(topEnd = 19.dp, topStart = 19.dp)) {

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

Button(onClick = { navController.navigate("login") },

colors = ButtonDefaults.buttonColors(

containerColor = Color(0xFF5C77AC))) {

Icon(

painter = painterResource(id = R.drawable.baseline\_keyboard\_backspace\_24),

contentDescription = "Voltar ao Login"

)

}

Button(onClick = { navController.navigate("telaperfil") },

colors = ButtonDefaults.buttonColors(

containerColor = Color(0xFF5C77AC))) {

Icon(

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

contentDescription = "perfil"

)

}

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

colors = ButtonDefaults.buttonColors(

containerColor = Color(0xFF5C77AC))) {

Icon(

painter = painterResource(id = R.drawable.baseline\_calendar\_month\_24),

contentDescription = "Calendário"

)

}

Button(

onClick = { navController.navigate("enviaremail")},

colors = ButtonDefaults.buttonColors(

containerColor = Color(0xFF5C77AC))) {

Icon(

painter = painterResource(id = R.drawable.baseline\_outgoing\_mail\_24),

contentDescription = "Enviar email"

)

}

}

}

}

}

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

//@Composable

//fun GmailRowItemPreview2() {

// GmailRowItem2()

//}

@Composable

fun EmailCard() {

Card(modifier = Modifier

.fillMaxWidth()

.padding(vertical = 13.dp)

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

) {

Row(verticalAlignment = Alignment.CenterVertically, modifier = Modifier.background(Color.White

).clickable { true }){

Column(

modifier = Modifier

.padding(12.dp)

.weight(2f)

) {

Text(text ="Remetente do Email", fontSize = 19.sp, fontWeight = FontWeight.Bold)

Spacer(modifier = Modifier.padding(1.dp))

Text(text = "titulo do email", fontSize = 16.sp)

Spacer(modifier = Modifier.padding(1.dp))

Text(text = "descrição do email", fontSize = 13.sp)

}

Icon(

modifier = Modifier.padding(horizontal = 15.dp),

painter = painterResource(id = R.drawable.baseline\_star\_border\_24),

contentDescription = "estrela"

)

Icon(

modifier = Modifier.padding(horizontal = 15.dp),

imageVector = Icons.Default.Delete,

contentDescription = "estrela"

)

}

}

}