МГТУ им. Н.Э. Баумана

Отчёт по лабораторной работе №6
по курсу «ПиКЯП»

Студент группы ИУ5-31Б
Крюков В. А.

2024 г.

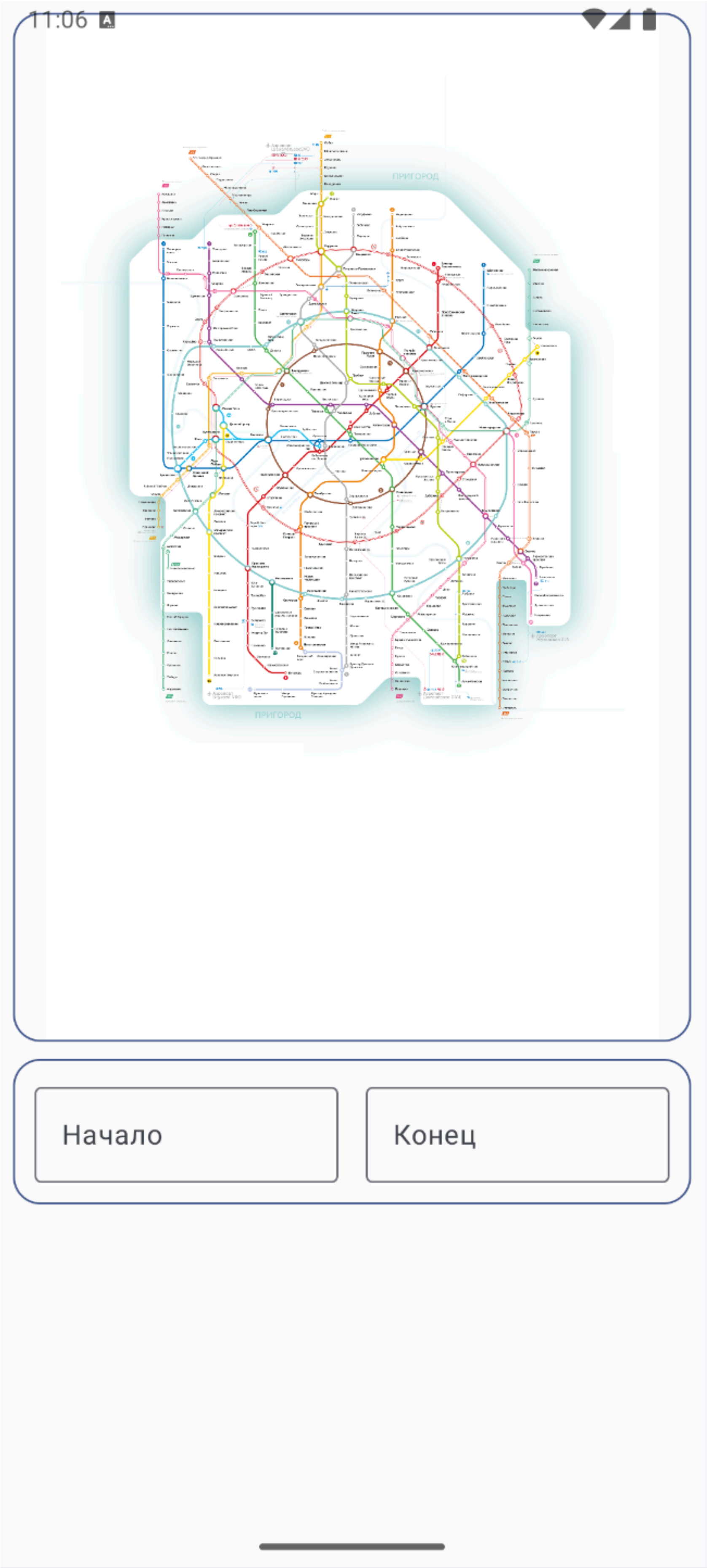
Тема: работа с элементами UI, работа с текстовыми полями. Попытка воплощения моей идеи приложения. Выполнив идею до определенного этапа, я понял, что это не то, как это приложение должно выглядеть.

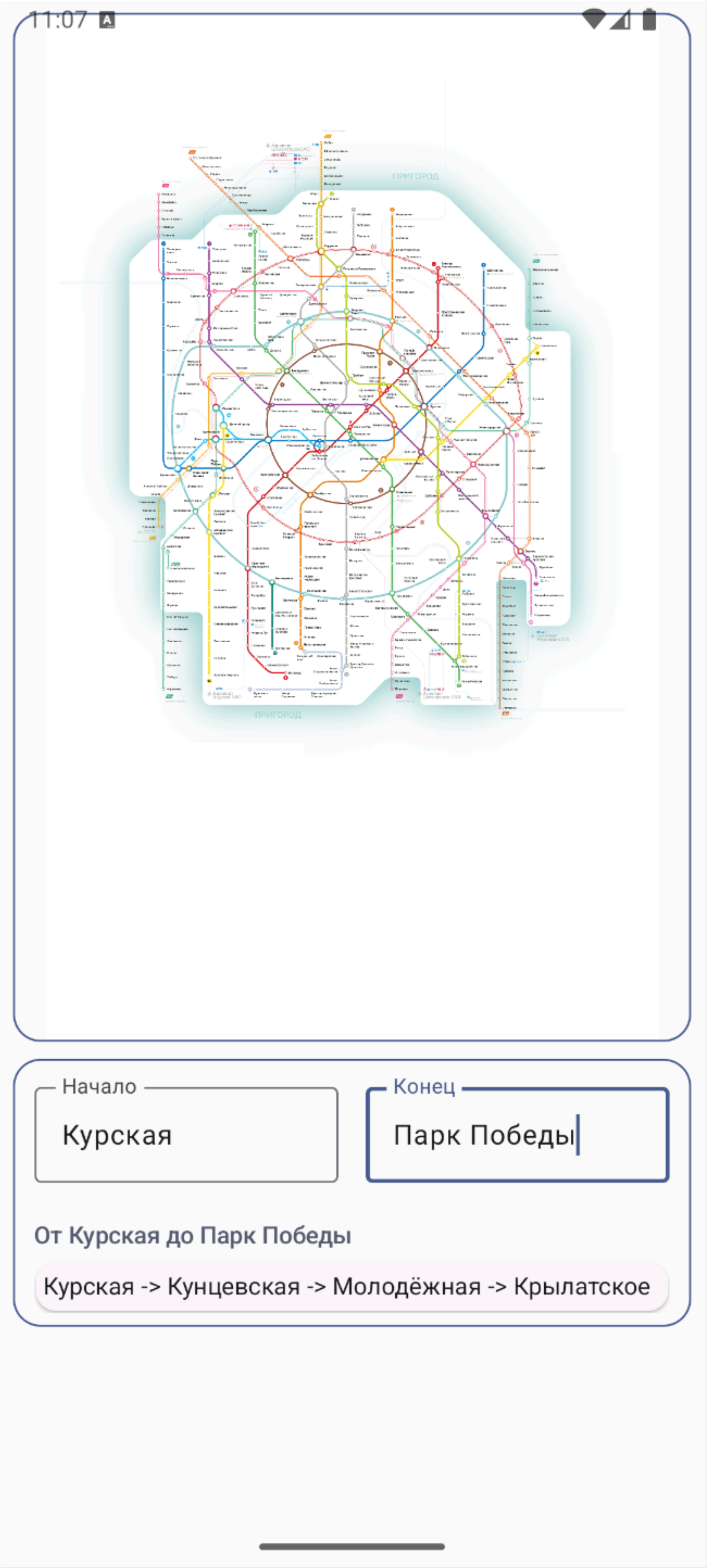
Имитация создания маршрута метро по вводным значениям.

Текст программы:
MainActivity.kt

```
package com.example.metrofymscreen

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.animation.animateColorAsState
import androidx.compose.animation.animateContentSize
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Text
import androidx.compose.material3.TextField
import androidx.compose.foundation.border
import androidx.compose.foundation.clickable
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.saveable.rememberSaveable
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.runtime.remember
import androidx.compose.material3.MaterialTheme
import com.example.metrofymscreen.ui.theme.MetrofyMScreenTheme
import androidx.compose.ui.Alignment
import androidx.compose.ui.draw.clip
import androidx.compose.ui.graphics.RectangleShape
import androidx.compose.ui.res.painterResource
import androidx.compose.foundation.gestures.detectTransformGestures
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.width
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.runtime.*
import androidx.compose.ui.geometry.Offset
import androidx.compose.ui.graphics.Path
import androidx.compose.ui.input.pointer.pointerInput
import androidx.compose.ui.layout.layout
import androidx.compose.ui.unit.IntOffset
import kotlin.math.roundToInt
import androidx.compose.ui.graphics.graphicsLayer // For
Modifier.graphicsLayer


val stations: Map<String, Int> = mapOf(
    "Щёлковская" to 3,
    "Первомайская" to 3,
    "Измайловская" to 3,
    "Партизанская" to 3,
    "Семёновская" to 3,
    "Электрозаводская" to 3,
    "Бауманская" to 3,
    "Курская" to 3,
    "Площадь Революции" to 3,
    "Арбатская" to 3,
    "Смоленская" to 3,
    "Киевская" to 3,
    "Парк Победы" to 3,
    "Славянский бульвар" to 3,
    "Кунцевская" to 3,
    "Молодёжная" to 3,
    "Крылатское" to 3,
    "Строгино" to 3,
    "Мякинино" to 3,
    "Волоколамская" to 3,
    "Митино" to 3,
    "Пятницкое шоссе" to 3
)


class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            MetrofyMScreenTheme {
                BottomSheet()
            }
        }
    }
}

@Composable
fun BottomSheet() {
    var pathFinderResult by remember { mutableStateOf(PathItem("",
listOf(""))) }

    fun pathFinder(start: String, end: String): PathItem {
        return PathItem("От $start до $end",
listOf(start,"Кунцевская","Молодёжная","Крылатское",end))
    }

    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(8.dp),
        //horizontalAlignment = Alignment.CenterHorizontally
    ) {
        // Metro map section
        Image(
            painter = painterResource(R.drawable.metro_map),
            contentDescription = null,
            modifier = Modifier
                .size(600.dp)
                .border(
                    1.dp,
                    MaterialTheme.colorScheme.primary,
                    RoundedCornerShape(16.dp)
                )
        )

        // Search part and path result
        Column(
            modifier = Modifier
                .fillMaxWidth()
                .padding(top = 10.dp)
                .border(
                    1.dp,
                    MaterialTheme.colorScheme.primary,
                    RoundedCornerShape(16.dp)
                )
            //horizontalAlignment = Alignment.CenterHorizontally,

        ) {
            SearchPart(
                onSubmit = { start, end ->
                    pathFinderResult = pathFinder(start, end)
                }
            )

            if (pathFinderResult != PathItem("", listOf(""))) {
                ShowPaths(pathFinderResult)
            }
        }
    }
}

@Composable
fun SearchPart(modifier: Modifier = Modifier, onSubmit: (String, String) ->
Unit) {
    var firstText by rememberSaveable { mutableStateOf("") }
    var secondText by rememberSaveable { mutableStateOf("") }

    Row(
        modifier = modifier
            .padding(4.dp)
            .padding(4.dp)
    ) {
        Column(
            modifier = modifier
                .weight(1f)
                .padding(bottom = 4.dp, end = 8.dp,start = 4.dp),
            horizontalAlignment = Alignment.Start

        ) {
            // First TextField
            OutlinedTextField(
                value = firstText,
                onValueChange = { newText -> firstText = newText },
                label = { Text("Начало") },
                modifier = Modifier.fillMaxWidth(),
                singleLine = true
            )
        }

        Column(
            modifier = modifier
                .weight(1f)
                .padding(bottom = 4.dp,start = 8.dp, end = 4.dp),
            horizontalAlignment = Alignment.End
        ) {
            // Second TextField
            OutlinedTextField(
                value = secondText,
                onValueChange = { newText -> secondText = newText },
                label = { Text("Конец") },
                modifier = Modifier.fillMaxWidth(),
                singleLine = true
            )
        }

        // Trigger onSubmit if valid inputs are provided
        if (firstText in stations && secondText in stations) {
            onSubmit(firstText, secondText)
        }
    }
}

//TODO()
//after i need to handle the firstText and secondText
//pass them to function when the user finishes the text in both of them ->
how do i do it?
//if the firstText and the secondText exists in the listOf stations -> the
input is done
//but it is the task for later

//later i should change the List<String> to List<whatever class i have for
my stations>
data class PathItem(val basicInfo: String, val pathList: List<String>)

@Composable
fun ShowPaths(path: PathItem){
    Row (modifier = Modifier.padding(vertical = 8.dp, horizontal = 12.dp)){
        var isExpanded by remember { mutableStateOf(false) }
        //surfaceCo;or will be updated gradually from one color to another
        val surfaceColor by animateColorAsState(
            if (isExpanded) MaterialTheme.colorScheme.primary else
MaterialTheme.colorScheme.surface,
        )

        //toggle the isExpanded when we click on this Column
        //all of the text handling -> msg.author, msg.body and the
expansion of the message
        //expansion changes -> color and size of the text
        Column (modifier = Modifier.clickable {isExpanded =!isExpanded}){
            Text(
                text = path.basicInfo,
                color = MaterialTheme.colorScheme.secondary,
                style = MaterialTheme.typography.titleSmall
            )
            //add a vertical space between the author and message text
            Spacer(modifier = Modifier.height(4.dp))

            Surface (
                shape = MaterialTheme.shapes.medium,
                shadowElevation = 1.dp,
                color = surfaceColor,
                modifier = Modifier.animateContentSize().padding(1.dp)
            ){
                Text(
                    text = path.pathList.joinToString (separator = " -> ")
,
                    modifier = Modifier.padding(all=4.dp),
                    maxLines = if (isExpanded) Int.MAX_VALUE else 1,
                    style = MaterialTheme.typography.bodyMedium
                )
            }
        }
    }
}
@Preview
@Composable
fun PreviewBottomSheet(){
    BottomSheet()
}
```

Результаты работы:


11:06
ПРИГОРОД
ПРИГОРОД
Начало
Конец



Начало

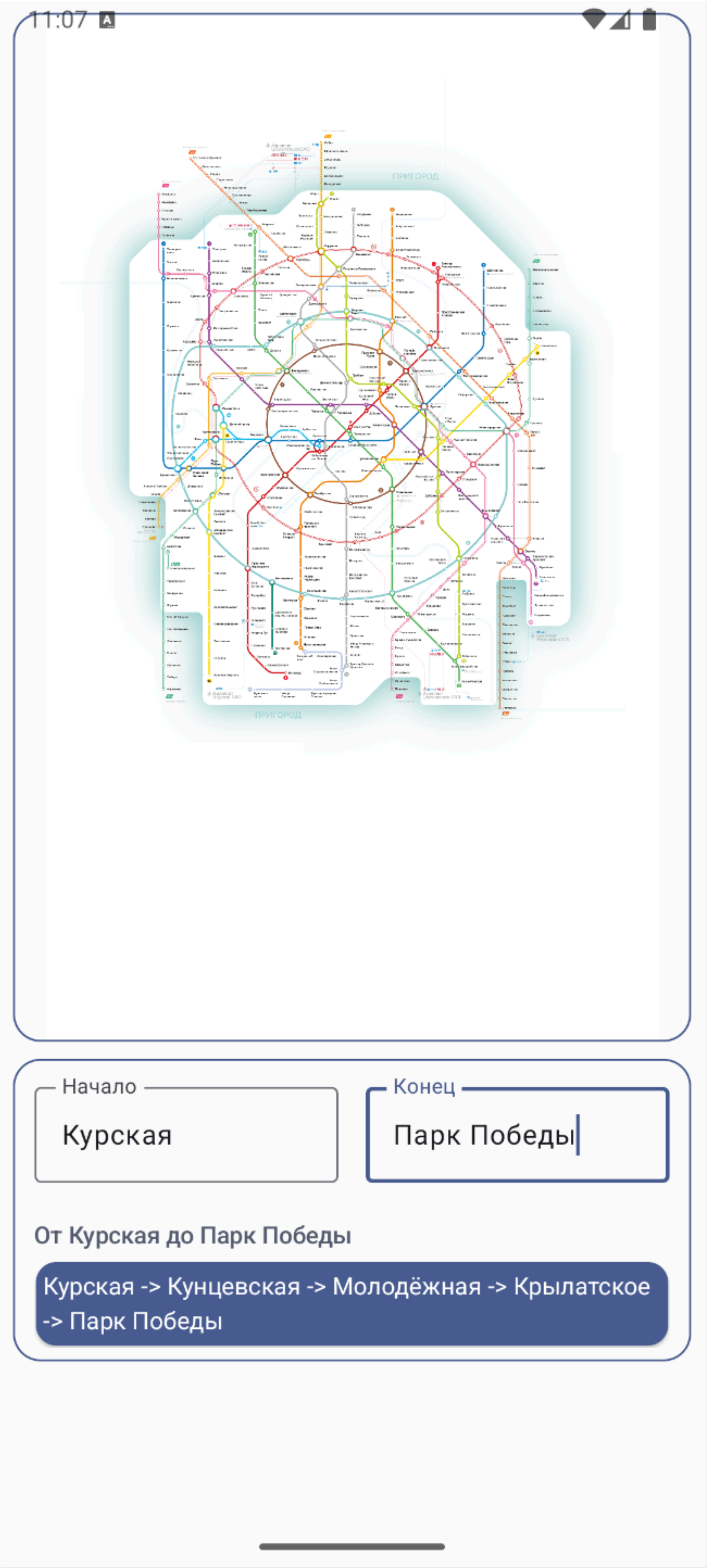
Курская

Конец

Парк Победы

От Курская до Парк Победы

Курская -> Кунцевская -> Молодёжная -> Крылатское



Начало

Курская

Конец

Парк Победы

От Курская до Парк Победы

Курская -> Кунцевская -> Молодёжная -> Крылатское -> Парк Победы