Transportation Problem Solver APP

Introduction:

The goal of this application is to solve the transportation problem using the North-West algorithm by using KOTLIN and XML for development

Developed by:

-BENGUELLA MOHAMMED ELAMINE TAHAR

-GHOURBAL AYOUB

North-West algorithm:

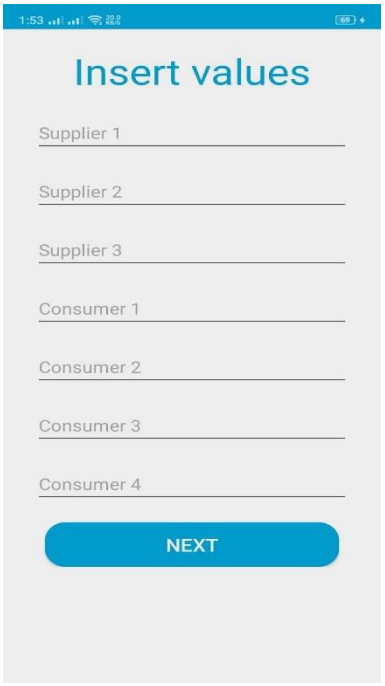
```
private fun solveTransportationProblem(supply: IntArray, demand: IntArray): Array<IntArray> {
        val m = supply.size
        val n = demand.size
        val allocations = Array(m) { IntArray(n) }

        var i = 0
        var j = 0

        while (i < m && j < n) {
            val quantity = minOf(supply[i], demand[j])
            allocations[i][j] = quantity
            supply[i] -= quantity
            demand[j] -= quantity

            if (supply[i] == 0) {
                i++
            }

            if (demand[j] == 0) {
                j++
            }
        }

        return allocations
    }
```

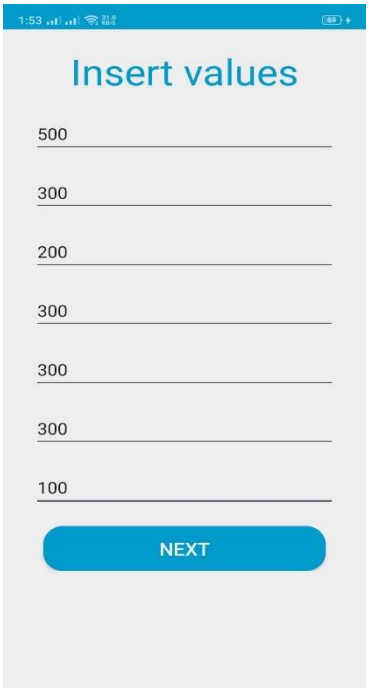
Calculate total cost:

```
private fun calculateTotalCost(allocations: Array<IntArray>, costs: IntArray): Int {
        var totalCost = 0
        var k = 0

        for (i in allocations.indices) {
            for (j in 0 until allocations[i].size) {
                totalCost += allocations[i][j] * costs[k]
                k++
            }
        }

        return totalCost
    }
```

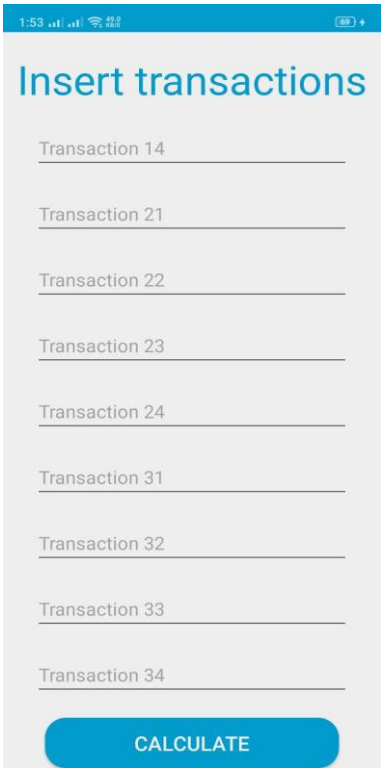
UI and the functionality of the app:

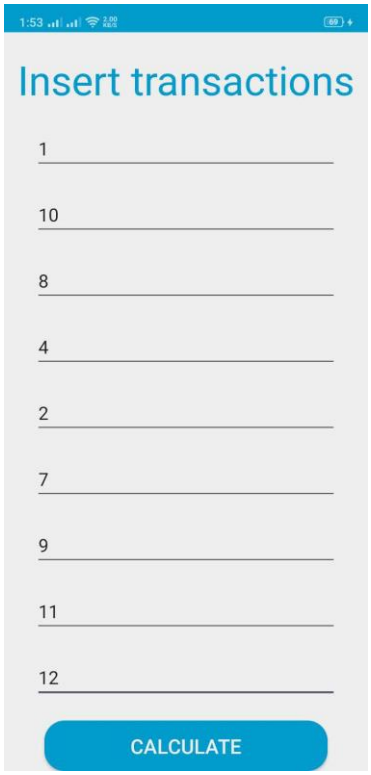
Here the user insert the number of suppliers and consumers (in this example it's suppliers=3 and consumers=4), then after clicking NEXT it shows:





Here the app displays another input to insert the values of each supplier and consumer (in this example it's 500 300 200 for suppliers and 300 300 300 100 for consumers), then after clicking NEXT it shows:





Here the app displays another input to insert the transactions of each supplier and consumer (in this example it's 6 5 3 1 10 8 4 2 7 9 11 12), then after clicking CALCULATE it shows the result:

How back-end work (by using KOTLIN):

The application creates dynamically all inputs depending on the number of suppliers and consumers entered firstly.

The application transfers the entered data via SHAREDPREFRENCES (named by CASHE), that's why if you click the back button of the phone you can still modify the entered values and get the new result.