Question: https://leetcode.com/problems/invalid-transactions/

So first we create a user defined class to store Transaction Entries.

And store entries of a user in a list as values and the user names as key, in a hashmap.

Now iterate through the hashmap and check all transaction of users and check for their invalid transactions.

Code:  
class Solution {

private static final String DELIMITER = ",";

private static final Integer MAX\_AMOUNT = 1000;

private static final Integer MAX\_DIFF = 60;

private static class TransactionEntry {

int id;

String name;

int time;

int amount;

String city;

public TransactionEntry(int id, String name, int time, int amount, String city) {

this.id = id;

this.name = name;

this.time = time;

this.amount = amount;

this.city = city;

}

}

public List<String> invalidTransactions(String[] transactions) {

if (transactions == null) {

return Collections.emptyList();

}

Map<String, List<TransactionEntry>> nameToTransactions = new HashMap<>();

for (int i = 0; i < transactions.len actions[i].split(DELIMITER);

TransactionEntry entry = new TransactionEntry(i, splittedTransaction[0], Integer.parseInt(splittedTransaction[1]), Integer.parseInt(splittedTransaction[2]), splittedTransaction[3]);

List<TransactionEntry> listTransactions = nameToTransactions.getOrDefault(entry.name, new ArrayList<>());

listTransactions.add(entry);

nameToTransactions.put(entry.name, listTransactions);

}

List<String> invalidTransactions = new ArrayList<>();

for(Map.Entry<String, List<TransactionEntry>> entry : nameToTransactions.entrySet()) {

checkList(entry.getValue(), invalidTransactions, transactions);

}

return invalidTransactions;

}

private void checkList(List<TransactionEntry> entries, List<String> invalidTransactions, String[] transactions) {

for (TransactionEntry entry1 : entries) {

if (entry1.amount > MAX\_AMOUNT) {

invalidTransactions.add(transactions[entry1.id]);

continue;

}

for (TransactionEntry entry2 : entries) {

if (entry1.id != entry2.id && Math.abs(entry1.time - entry2.time) <= MAX\_DIFF && !entry1.city.equals(entry2.city)) {

invalidTransactions.add(transactions[entry1.id]);

break;

}

}

}

}

}

Github Link :<https://lnkd.in/ecwtJeaz>