Technical Question

The problem

Payments.csv shows a list of payments based on device_id. We would like to use the payment 'history' to

calculate the time that a client's policy is valid for. This can be looked at as the total number of days until the client is suspended (Days from Suspension).

Once a payment has been made a client is covered for 30 days (this is the time during which any claims will be paid out). We know that clients will not always have the money on their exact payment date, and income is volatile, so after the 30 days of coverage period, the client enters into the grace period for 30 days. If a client misses their monthly payment, they will go into the grace period. They can then make a payment to catch up. Furthermore, after the end of the grace period a client will still have 30 days of their suspension period. A client will only be suspended after 91 days of their first payment.

Another way of looking at it, if a client is up to date on payments, a client is 90 days from suspension. We have numerous products with different premiums, but the same logic applies to all of them. We want to give our agents as much information possible to manage their clients.

We want to be able to give agents a "filtered" list that shows their most pressing clients first.

From an operational perspective, we want 3 reports:

1. Write a system/program that computes a *Days from Suspension Report*
2. Collection per agent, per day, and per payment type
3. Total collection per payment type

How to run the program

The program must accept two parameters: <input_file> <output_folder>.

The first parameter is a csv file with the name YYYY_MM_DD_payments.csv, and the second is a folder which must not initially exist.

YYYY_MM_DD_payments.csv file content

This CSV file contains the list of all payments logs created in our system until the date YYYY_MM_DD in the

file name. File format is:

```
id,payment_type,payment_amount,payment_signature_image,payment_photo,created,status,agent_user_id,device_id
```

Where:

id: Payment transaction id

payment_type: Payment type used by the client or operational team i.e. CASH, CLIENT_REFERRAL,

etc.

payment_amount: Payment amount.

payment_signature_image: Client's signature image url.

payment_photo: Proof of Payment image url.

created: Payment creation time

status: Payment status.

notes: Payment notes, added by operational team.

agent_user_id: agent identifier, agent responsible for the payment in question

device_id: Client/Device identifier, payer.

Output folder content

This folder must contain the result files of the execution of the program:

days_from_suspension_report.csv, containing the list of clients and their days from

suspension

agent_collection_report.csv, containing collections per agent, per day, and per payment type

payment_type_report.csv, containing total collections per payment type

days_from_suspension_report.csv file

A CSV file with that contains the list of clients and their days from suspension using the following format:

```
device_id,days_from_suspension
```

Where:

device_id: Client/device identifier.

days_from_suspension: Calculated day from suspension.

The file should be ordered by days_from_suspention in desc order.

For example:

18773,1232345,85321,30

agent_collection_report.csv

A CSV file containing the collections per agent, per day, and per payment type

The format for this file is:

agent_user_id,date,payment_type,total_amount

Where:

agent_user_id: agent identifier, agent responsible for the payment in question.

date: Payment date

payment_type: Payment type used by the client or operational team i.e. CASH, CLIENT_REFERRAL,

etc.

total_amount: Total amount paid on the day with the payment type in question.

File must be ordered by agent_user_id and date

For example agent_collection_report.csv:

1,2018-12-10,CASH,1800

1,2018-12-11,CLIENT_REFERRAL,60

2,2018-12-02,CARD,150

2,2018-12-11,BANK_DEPOSIT,360

payment_type_report.csv

A CSV file containing containing total collections per payment type

The format for this file is:

payment_type,total_amount

Where:

payment_type: Payment type used by the client or operational team i.e. CASH, CLIENTE_REFERRAL,

etc.

total_amount: Total amount paid with the payment type in question.

File must be ordered by payment_type in lexicographic order.

For example:

CARD,43200

```
CASH,20000

EASY_PAY,84000
```

Program requirements

Submission should have README.md file describing:

How to compile and run program

How to run tests

Some description of the algorithms and decisions taken during implementation