Transactions Cloud

OVERVIEW

Create a view in which it's possible to observe and highlight specific kinds of transactions within a given period for an account.

THE DATASET

These transactions, an example follows later, are grouped by a given set of rules whose format is as follows:

RULE SET

```
[{
   "ruleMatchValue": String, //Matched against transactionDescription
   "ruleMatchType": Enum String, //The type of the match e.g. exact, contains
   "ruleFlag": Enum String, //Color assigned to the category e.g Red, Yellow
   "ruleCategory": String //A category such as Withdrawal, Default etc
}]
```

Given this set of rules, the default forms of grouping are by **`ruleCategory`** and by **`ruleMatchValue`**, which are applied whenever a user interacts with the UI to select its preferred option.

`ruleMatchType` might be one of `exact`, `contains`, `startsWith`, `endsWith` and `regex`. These are nothing but 5 different ways of detecting if a string matches parts of another string. The string to be matched, **`transactionDescription`**, should be lower cased before invoking any of these.

The format for transactions that these rules are applied to is a follows

TRANSACTION SET

```
[{
   "transactionId": String,
   "transactionDate": String,
   "transactionType": String,
   "transactionDescription": String
}]
```

You'll find the data for this challenge here. Please don't copy/paste content. Download the files in order to avoid any encoding issues.

MATCH EXAMPLE

The following rule

RULE

```
{
   "ruleMatchValue": "loan",
   "ruleMatchType": "contains",
   "ruleFlag": "Yellow",
   "ruleCategory": "Loans"
 }
```

matches

TRANSACTION

```
{
   "transactionId": "3822888e-5f7e-4ae2-a723-06b5238cbf3a",
   "transactionDate": "Fri Dec 09 2016 11:30:26 GMT+0100 (CET)",
   "transactionType": "deposit",
   "transactionDescription": "XYZ  Loan  repayment"
 }
```

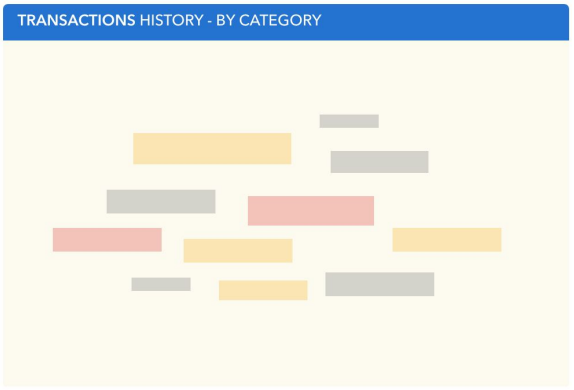
NOTE: The match is case insensitive.

THE USER INTERFACE

As the data volume may be too large for direct human evaluation, the interface is to provide a simplified view on the information, making it easier for anyone to verify the transactions history and, through that, the health of an account.

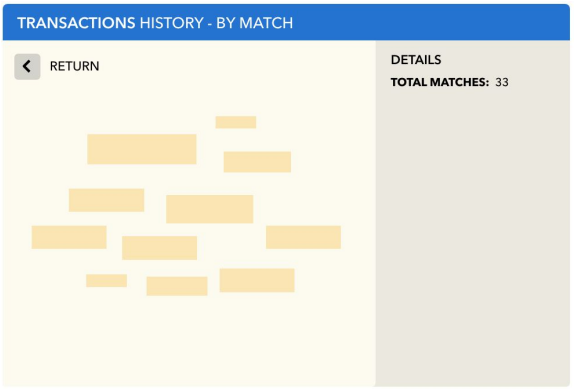
SCREEN STATE 1

On its first state, the application displays a cloud view on the transactions dataset grouped by **category,** colorized by their **flag**. e.g. “Withdrawal” with some visual reference to the color “Yellow”.



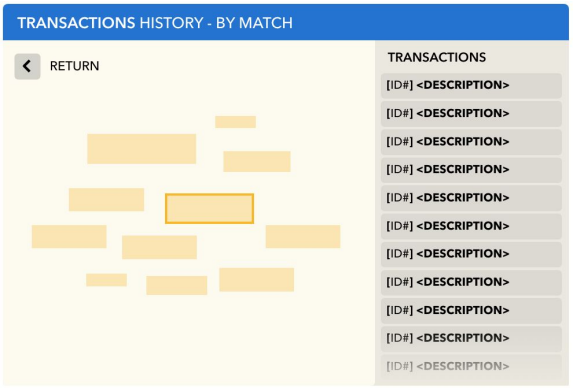
SCREEN STATE 2

When the user clicks on one of the categories, the view becomes grouped **by match**.
When hovering one of the items, the count of matches should be shown on the sidebar.



SCREEN STATE 3

When the user clicks on one of the match rules the application highlights the selection and displays on the side a list of the related transactions.



DEVELOPMENT AND TOOLING

ECMASCRIPT VERSION

You are free to use any version of Ecmascript. We recommend using the latest.

FRAMEWORKS & LIBRARIES

Implementing all the necessary functionalities (such as tag cloud generator etc) to execute this task is unrealistic. We encourage you to select any frameworks and libraries you consider appropriate.

You are also encouraged to use libraries such as Ramda, Ramda Fantasy, Sanctuary and/or Folktale ***if*** you feel comfortable working with them. This is a recommendation, not a requirement.

THE TOOLING

Webpack, Rollup, Browserify, whatever you feel comfortable with. Internally we use Webpack, but the relevant point is to evaluate the understanding of the concepts behind build tools and transpilation, so any of them will suffice.

TESTING

Unit and functional tests are a requirement for this task.

EXTRA POINTS

As extra points to the task, you may feel like:

- Adding an option directly in the UI to add a new rule that reevaluates the data and updates the cloud.
- Separate the data by quarters of the year to make analysis more detailed.
- Create charts to show the viewer the development of an account over time.

NOTE: These are not essential.

SUBMISSION

Please create a **private** Bitbucket Git repository and share with the following accounts:

ppurang-spotcap, **karthik-spotcap** and **tszekely-spotcap**