**M1 Finance – Take-home Technical Assessment Requirements**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| **1.0** | **03/13/2021** | **Initial version of document. Requirements build on those provided as instructions for the project.** |

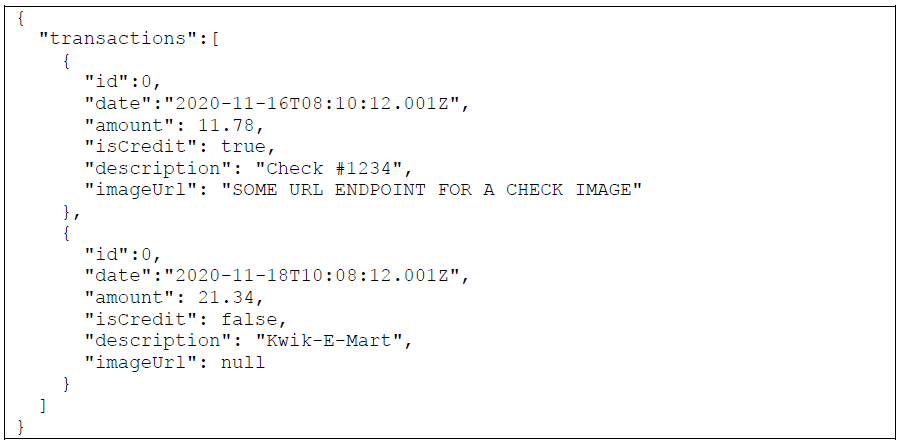
These are the requirements for the app fulfilling the goal of M1 Finance’s asynch coding challenge. These requirements build on the ones provided as instruction for the challenge. Additions to the requirements have been made to make behaviors more specific or to cater to a more streamlined user experience.

Data:

1. The primary data of the app shall include financial transactions for any given user pertaining to their financial account.
   1. Data sources.
      1. A provided endpoint.

Primary transaction data shall be accessed using an endpoint.

* + - 1. The JSON definition of the endpoint is as follows:



* + - 1. CRUD operations on primary transaction data shall be done on the provided endpoint.
      2. Supported financial transactions shall include debit purchases, refunds, and deposited checks.
         1. Each supported financial transaction shall include the following components:

The amount of the transaction.

The description of the transaction.

The date of the transaction.

* + - * 1. Deposited checks shall additionally include a component that is the image of the check.

The image shall not necessarily be available for each transaction of this type.

* + 1. Locally saved user data.
       1. The following shall be saved locally on the user’s device:
          1. All user-defined app settings.
          2. All secondary transaction data, including:

Any notes that a user makes on individual transactions.

Each note saved shall pertain to only one transaction.

User interface:

1. List of financial transactions.

The user’s financial transactions shall be rendered as a list.

* 1. Each row item in the list shall represent a single financial transaction.
     + 1. The row item shall show all the primary transaction data retrieved from the endpoint.
     1. The row item shall have a visually distinguishable color scheme to notify the user of that transaction’s effect on their account balance.
        1. If the row item represents a debit transaction, it shall have a red color scheme.
        2. If the row item represents a credit transaction, it shall have a green color scheme.
     2. Clicking on the row item shall open a detailed transaction view (see below) for that transaction.
  2. The list shall initially be rendered in the same order as provided by the endpoint.
  3. The user shall have the ability to sort the list of transactions.
     1. The control to sort transactions shall be located on the top right corner of the screen, above the list of transactions.
     2. The sorting options available to the user shall include:
        1. Sorting by date.
           1. Descending.

This sort shall reorder the list of transactions such that the transactions shall appear from most recent at the top to least recent at the bottom.

* + - * 1. Ascending.

This sort shall reorder the list of transactions such that the transactions shall appear from most recent at the top to least recent at the bottom.

* + - * 1. The default sort type for sorting by date shall be descending.
      1. Sorting by amount.
         1. Descending.

This sort shall reorder the list of transactions such that the transaction with the largest amount shows at the top and the transaction with the smallest amount shows at the bottom.

* + - * 1. Ascending.

This sort shall reorder the list of transactions such that the transaction with the smallest amount shows at the top and the transaction with the largest amount shows at the bottom.

* + - * 1. The default sort type for sorting by amount shall be descending.
    1. Only one sorting option shall be selected at any given time.
       1. The default sorting option shall be a descending sort by date.
    2. A button shall be shown for each sorting method.
       1. The button shall show the name of the method and its type.
          1. The type shall be represented by an upwards arrow or downwards arrow for ascending and descending, respectively.
    3. The user shall choose the sort to use by clicking on its icon.
       1. Clicking on a sorting method that is not selected shall enable the default sort type for that method.
       2. Clicking on a sorting method that is selected will toggle the sort type. For instance, if sorting by date is selected and the current type is ascending, clicking the icon shall change the sort type to descending.
       3. Any other sorting method other than the currently selected method shall be deselected.
  1. Refreshing of the transaction list.

The transaction list shall refresh when any of the following events occur:

* + 1. The app being started.
    2. The app being re-opened.
    3. The user clicking on a refresh button positioned on the top left corner of the screen, above the transaction list.

1. Detailed transaction view.

The detailed transaction view shall give the user an expanded view of a particular transaction, depending on the transaction type.

* 1. The user shall be able to open a transaction’s detailed transaction view by clicking on the transaction’s row item.
  2. The detailed transaction view shall be an overlay on top of the transaction list.
     1. If the user clicks anywhere outside the overlay:
        1. The detailed transaction view will go away.
        2. No interaction with the part of the screen that was tapped will occur. For instance, if another transaction is clicked to dismiss the overlay, that transaction’s detailed view shall not open unless the user clicks on it again.
  3. Clicking on a debit transaction shall show a detailed transaction view including the following information.
     1. The amount of the transaction.
     2. The description of the transaction.
     3. The date of the transaction.
     4. The user’s notes on transaction.
        1. If the user has not written any note for the transaction, the field will be blank.
  4. Clicking on a credit transaction.
     1. If an image of the associated check is available, a detailed transaction view shall be shown to the user containing only that image.
     2. If an image of the associated check is not available, clicking on the transaction shall not have any resultant effect.

Definitions:

1. CRUD: These are the four basic types of transactions one can carry out on persistent data, namely, create, read, update, and delete.
2. Debit transaction: a transaction that has a net result of decreasing a user’s account balance.
3. Credit transaction: a transaction that has a net result of increasing a user’s account balance.