1. **Which pages do you envision on this app? (rough description or mockups)**
   1. The first page will be a login page
   2. The second page will be a dashboard, where user can choose his account and see his balance, last 10 transactions, his account health status and any kind of important information we may consider.
   3. The Main component will have a menu with
      1. Dashboard
      2. Account Transactions: user will be able to choose his account (may have more than one) containing all transactions information, in a table, containing date, transaction type, amount, approved or canceled, source account and destination account. (maybe some automatically charges, if you received some money from another account). It may be to long to show in a screen, so depending on its resolution we will hide/show columns. Also a details button will be provided in every line, to see detailed information, such transaction history.
      3. New Transaction Button opens a modal.
   4. The third page will be a modal, providing user to create a new transaction. User will choose his account, the destination account and amount. Validations will occur and transaction should be finished or canceled lately, based on a automated process that runs asynchronous.
2. **Describe or illustrate the database schema of this application. Please include data types and relationships.**
   1. See attached sql file to create this database (in sql server)
   2. Database Model print

Diagram

Description automatically generated

1. **Describe or illustrate the HTTP routes of the app - or graphql schemas, if you prefer. What** parameters do they expect?
   1. GetAccounts(int userId).
   2. GetAccountTransactions(int accountId).
   3. NewTransaction(int sourceAccountId, int destinationAccountId, decimal amount).
   4. I’m not considering security in those cases, as described in test base conditions.
2. **Which components do you envision on your frontend? (Thinking in terms of angular|react|vue components)**
   1. We will need a component for main application.
   2. DashboardSectionComponent that will represent the dashboard page.
   3. TransactionsSectionComponent that will represent the transactions page.
   4. NewTransactionSectionComponent that will represent the new transaction modal.
   5. TransactionsComponent, will have all transactions details, will be used in dashboard and transactions section component.
   6. UserAccountsComponent, will have all user accounts, will be used in dashboard and new transation sections.
   7. A few components to enhance dashboard experience.
3. **How are you dealing with money? (the account balance)**
   1. The account balance is a result of all operations made into it.

Also, we may think in a few summarized values by month, that will increase the calculation speed.

We must provide a way to recalculate its totals. Just in case.

1. **How are you dealing with currencies?**
   1. We must have a service who will deal with those calculations.

It will analyze each account, must query another centralized system to provide current conversion rate and make the appropriate conversion. The account has a fixed and immutable currency type that will provide way to identify necessary conversion.

1. **How are you storing and displaying the transaction dates?**
   1. All transactions date must be stored as UTC date. All user accounts will have an attribute called CityId, which will provide the necessary date conversion information. All dates will be converted to account city location.
2. **Assume the app took off and you are now dealing with millions of daily users. What comes to our mind in terms of making sure the app performance is good? (front, back or even server-wise)**
   1. We must use tools to monitor those api transactions, like Api Dynamics.
   2. Use Containers for web application under a Kubernetes is mandatory.
   3. We have tools to monitor page crashes, page loading time and pages that are often accessed, generating a lot of useful information for us to improve it somehow.
3. **(Optional) Working in a small squad (5~ devs), how would you break down this app into tasks?**
   1. I have created a briefly job separation, in azure devops, that may be developed at same time, depending on team size, iteration length and developer skills. It should be defined in Planning Meeting, defining the tasks effort with all team members.

A picture containing table

Description automatically generated

1. **(Optional) Share a snippet of code that you are proud of — whether it was you who wrote it, or something you read online — and describe why you enjoyed that code.**
   1. It will not be possible to show the code itself here but I’ll describe it.

One of our projects we had to export dozen of excel sheets based on classes. So we created some viewmodels, warping desired classes and worked with class/property attributes to automate excel export routine.

It was a charm, just must configure that viewmodel and, via reflection and conditions, the excel sheet would be created. It’s simple, but was useful. Saved us a lot of time