Saaif Ahmed

saaifza2

CS:598 Foundations of Data Curation

Assignment 2

Note: Step 2 Visualization can be found at the end of this document. The screenshots are spread across multiple pages, but it is all-inclusive. Concatenating the pictures vertically reveals the entire visualization.

Step 3:

Elements are in bold, and *Attributes* will be in italics.

accountStatement: this is the overarching element of the xml document. This schema is for account statements that banks can use to send to members at the end of each statement period. This xml schema covers the main information present on a real account statement, that being the trend of the funds in the account through the statement period.

summaryHeader: This element encapsulates the information of the account summary present on the document.

accountSummary: The accountSummary is the title of the section breaking down the account over the course of the statement period. This has an attribute of *summary* which indicates the version of account summary in relation to the original document.

summary: This is the attribute applied to accountSummary. This attribute, as a string, is meant to indicate between the major types of bank accounts, those being Checking, Savings, Certificate of Deposit, Money Market, and Individual Retirement Accounts. That is why these values are restricted to this attribute, as this schema is meant to represent these kinds of bank accounts. Any other account requires a different schema to represent.

accountTypeDesc: This element encapsulates the information relating to the sub-title of the accountSummary.

desc: This string element is plain text description of the account's "sub-type". Most banks have additional indicators beyond the major types shown in *summary*. For example, a "College" or "Student" checking account is different than a "Standard" checking account. This element serves to describe that indicator.

accountType: Much like the *summary* attribute this is used to distinguish what specific "sub-type" of account the overall account is. This is left as an attribute for ease of searching as statements with multiple accounts can use this to search through the document or verify that an account is a given type.

balanceBreakdown: This element starts the first major part of the document. Again, this encapsulates the breakdown of the balance from start to end of the statement period.

balanceBreakdownHeader: The header provided at the top of the breakdown. This string is used to provide a label for the following balances.

beginningBalance: This string element serves as the name identifier for its respective amount.

beginningBalanceAmount: This decimal represents the starting value of the account at the beginning of the statement period.

deposits: This string element is the name identifier for the deposit amount.

depositAmount: This element encapsulates the deposit value and restriction.

deposit: This decimal is the total amount deposited over the course of the statement period. It has a restriction where the minimum value must be 0, as one cannot deposit into their account a negative amount.

physicalWithdrawals: This string element is the name identifies for the amount physically withdrawn, either through an ATM or bank teller or other method that wasn't electronic or digital.

physicalWithdrawalsAmount: This element encapsulates a withdrawal element indicating the physical amount withdrawn.

electronicWithdrawals: This string element is the name identifies for the amount electronically withdrawn, either through online platforms, 3rd party tools, or other method that wasn't physical.

electronicWithdrawalsAmount: This element encapsulates a withdrawal element indicated the electronic amount withdrawn.

withdrawal: Present within electronicWithdrawalsAmounts and physicalWithdrawalsAmount this decimal element is meant to represent the amount withdrawn in these fashions. It has a restriction as the max value must be 0. As a withdrawal is meant to be a negative number, a withdrawal cannot be positive at all.

endingBalance: This string element serves as the name identifier for its respective amount.

endingBalanceAmount: This decimal represents the final value of the account at the end of the statement period.

noteToMember: This element is the second major part of the transaction summary. This element encapsulates the note to the member that the bank can use to provide additional data on the account standing.

note: This string element is meant to map the message from the bank to the member. This is useful to provide any additional information or clarity the member may require. Furthermore, this describes the state of the account.

noteType: This attribute is used to distinguish between the types of notes the bank can provide to the user. These may include "waiver" notes, or "fee" notes among others.

transactions: This element represents the last major part of the document. This marks the transaction section of the document, and the transaction table associated with it.

headersList: The headersList is meant to store the headers of the transaction table. These are the descriptors of the columns of the transaction table.

tableHeader: This string element is a single header in the header list. These can be mapped to whatever the document has them listed as.

transBeginningBalance: The transaction table should start by showing the starting value of the account at the beginning of the statement period. Therefore, this element serves that purpose as the descriptor for that amount.

transBeginningBalanceAmount: This string element is the amount associated with the beginning of the transaction table.

transactionTable: This element encapsulates the table of transactions and/or the history of the transactions through the statement period.

row: This element is a representation of an individual row in the transaction table. In each row, among all documents, should be a date, description of the transaction, the amount in the transaction, and the state of the balance afterwards. This represented by the elements that this one encapsulates.
date: This is the date of the transaction that appears in the transaction table.

description: This is the description of the transaction that appears in the transaction table.

amount: This is the amount of the transaction that appears in the transaction table.

balance: This is the balance of the account after the transaction in the transaction table.

transEndingBalance: At the end of the transaction table the final amount present in the account should be present. This represents the descriptor of the final value of the account at the end of the statement period.

transEndingBalanceAmount: This string element is the final value of the account at the end of the statement period present at the end of the transaction table.

currency: This attribute indicates the specific currency associated with the account. This is required because the numbers present in the document lose information without the currency applied to them.

language: This attribute indicates the language of the document. The default is EN to represent English.

Step 5:

The document that I chose is a typical bank statement letter that one receives at the end of a statement period. Knowing this, the decision to separate the document into 3 major components was clear. In any bank statement they have the beginning summary of the account showing the overarching changes made to the account, followed by a note explaining if any fees or waivers were applied to the account, lastly a table breaking down all the transactions involving the account is included. These were the 3 major elements of complex type that included the in the xml schema. As for the choice of elements I used my experience from HTML design and thought process on how to break down the design document. For the 3 major elements, 3 major encapsulating elements would be required. Within them, breaking them down by vertical alignment seemed to be the best. For this genre of document, all the information is typically stated in tabular fashion vertically. Hence why financial decisions are usually based around the “bottom line”. Therefore, creating the schema to break it down to that bottom line was the clear decision. The attributes were chosen to provide additional, yet optional information

surrounding the statement. Some attributes had to be required. Such as the currency and the type of account summary the statement referred to. These provided critical information relating to the context of the document and provided additional data that is useful for curatorial activities, namely archival purposes. Language as an attribute is commonly used in xml schemas as they let the parsing, or reader, or other data analyzer know what language the information in the document is conveyed in, which is clearly useful for comprehension of the information. The noteType attribute is again useful to indicate the note from the bank or similar organization to the customer or member. These are useful not at a given moment, but rather over a long course of time. One can search through these documents and gather the values for this attribute to see if a member had many fees, or if they've qualified for many waivers in the past. This behavior deduced from the implementation of this attribute is useful to a, for example, bank when assessing an application for a loan. The accountType attribute is yet another helpful attribute to provide additional context to the specific subtype of account the member has. This can provide information on how easy it was for the members to get a fee waiver or not, and it helps with other analytical activities.

The hardest decisions made during the process was determining the initial design of the schema. This document category, while structured can vary greatly depending on the bank, the account, the amounts of transactions, and much more. It was a challenge to determine what each element was trying to convey in a general sense, and then applying that information to the schema. I overcame this challenge by recalling the structure of financial documents, and the goal of any financial reports I have read in the past few years. The importance of the "bottom line", when I realized it, allowed me to cement my schema design process. I was then able to break the document down in the manner shown in the image at the bottom of this narrative. A lesser yet still prominent challenge was conveying the semantics of information present within the document category. I have a college checking account from which I retrieved a statement from to base the xml schema off. That semantic identifier of "college" checking account was critical, but I did not know how to reflect that in the document. However, finding attributes provided me with a solution as I knew it would allow the xml schema to further the goals of data curation.

Data independence is the idea that the logical schemas, and representations of the data should not be affected by changes in the data itself, and vice versa. An ER diagram or schema accomplishes this by taking pure data, applying pair to pair relationships onto it, and can adjust that diagram as needed. An xml schema performs the same task, but for unstructured data, text, and other information present on documents. By creating an xml schema, we can represent the information present on documents completely independently of the data itself. The schema is a representation of the structure of the information on the document but can be manipulated independent of the data. Furthermore, the data itself can change but the structure/schema can be applied as well. As an ER diagram abstracts away from the data in the context of bytes of data, an xml schema extracts away from the data present on documents.

The curatorial objectives supported or not supported by this xml schema are as follows:

Collection: The xml schema is designed as a representation of the information on the documents within the document category. Without this xml schema, extracting information from the documents would be incredibly challenging. With this xml schema the data can be mapped to a logical framework for use in curation.

Organization: This xml schema is a mapping for collection, but it is organized as well. This schema, and the xml documents that can be formed from it are well-formed and valid. Organization deals with the ability to retain attributes of the schema and changes to the schema. If this schema is stored in some repository a version history can be easily created and managed, and attributes are naturally a part of xml schemas.

Discoverability: As this is an xml schema, used to create xml documents, there are many ways to add metadata to the schema file for discoverability. While not done in this document, in an xml schema it can be added as additional elements, and then expanded upon in the marked-up document.

Access: Access is one of the curatorial activities not supported in its entirety by the xml schema. This schema is meant to apply for the general document genre, but by doing so the access is unrestricted and uncontrollable. This, while not a large negative, is undoubtedly not in full accordance with access as a curatorial objective. However, this xml schema by definition and implementation is a tool that supports the efficient and reliable retrieval and distribution of the data.

Identification: An xml schema supports identification by use of the markup language's tags. By applying tags, specifically element tags and names, to the xml schema we can break down the information on the document to a general mapping for use in analysis and comprehension.

Reformatting: An xml schema, including this one, is subject to an xml standard. While it is still valid to use the standard set in 2001, should it need to be changed it can be done so easily. Therefore, the reformatting curatorial goal is supported.

Sharing: This schema can be shared quite easily. As it is just a file that anyone can use, sending this by email is enough to get another data team going. There are many tools that are easy to use to determine if an xml file is valid against a given schema. And this narrative mockup can have the prose relating to elements and attributes extracted and packaged alongside the shared product.

Security: This is a curatorial objective not necessarily supported by this xml schema. This file isn't locked behind anything, and if a bank wanted to make this private to their organization it would not be available outside the box.

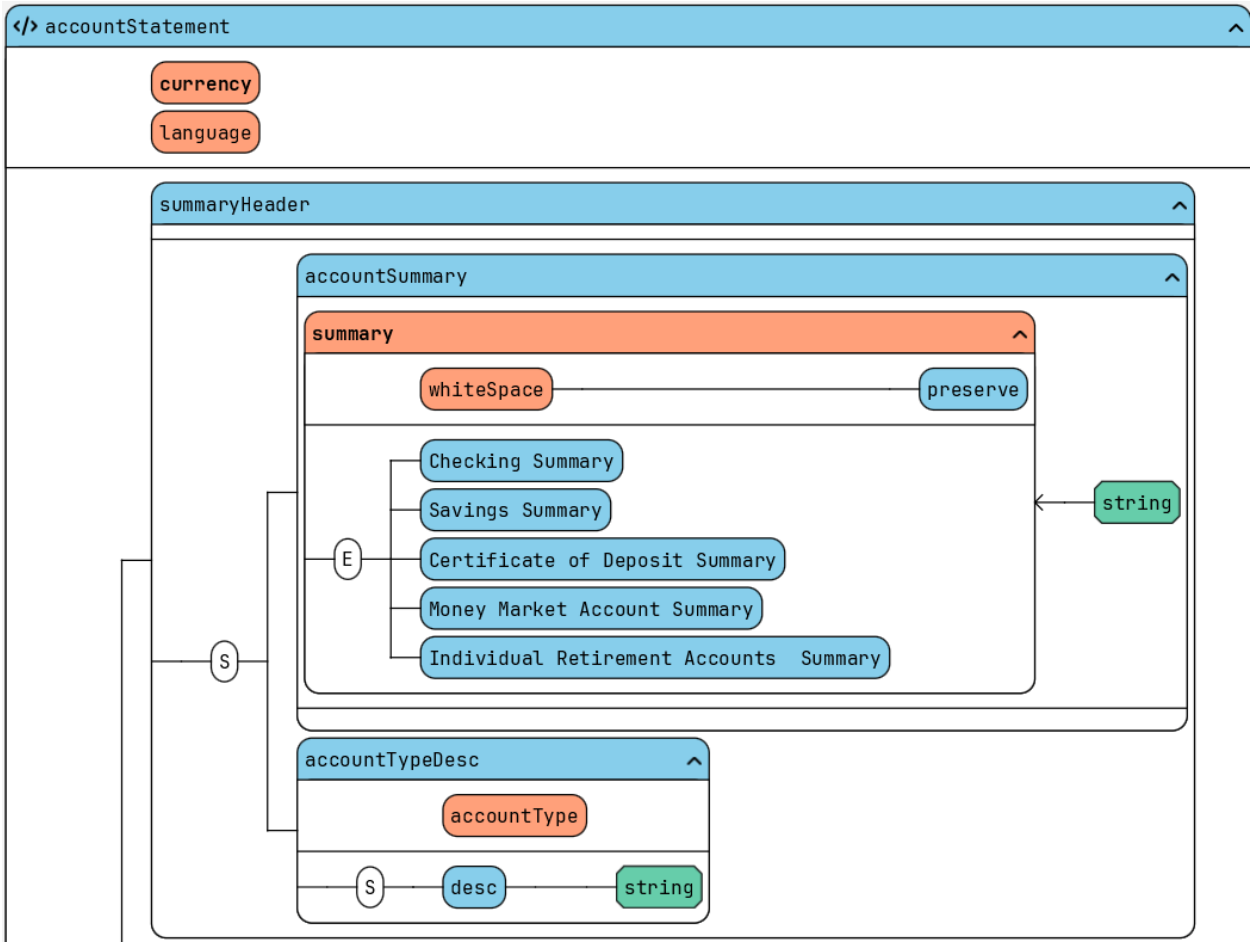
Modification: The xml schema is used to create an xml document that manifests the information within the document genre. If that data on the document were to change then the corresponding xml document can be modified to support that. This is because the xml schema abstracts away from the data itself and instead focuses on the data's representation and thus supporting this curatorial goal.

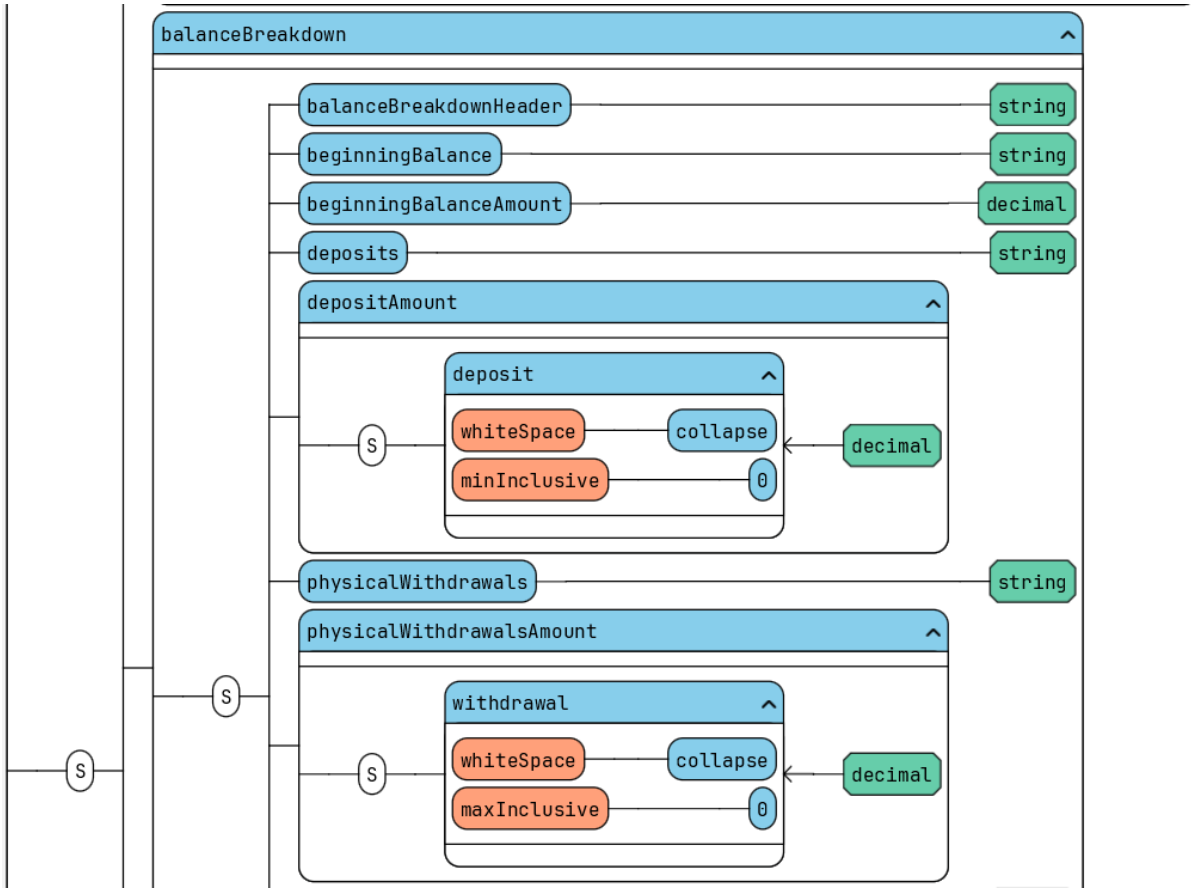
Workflow: The elements of the xml schema all have prose written about them that can assist in the workflow of the curation of the data. The naming conventions within the document are very general yet distinct from each other to avoid confusion. This means that the documentation could be generated automatically, in tandem with the workflow curation goal.

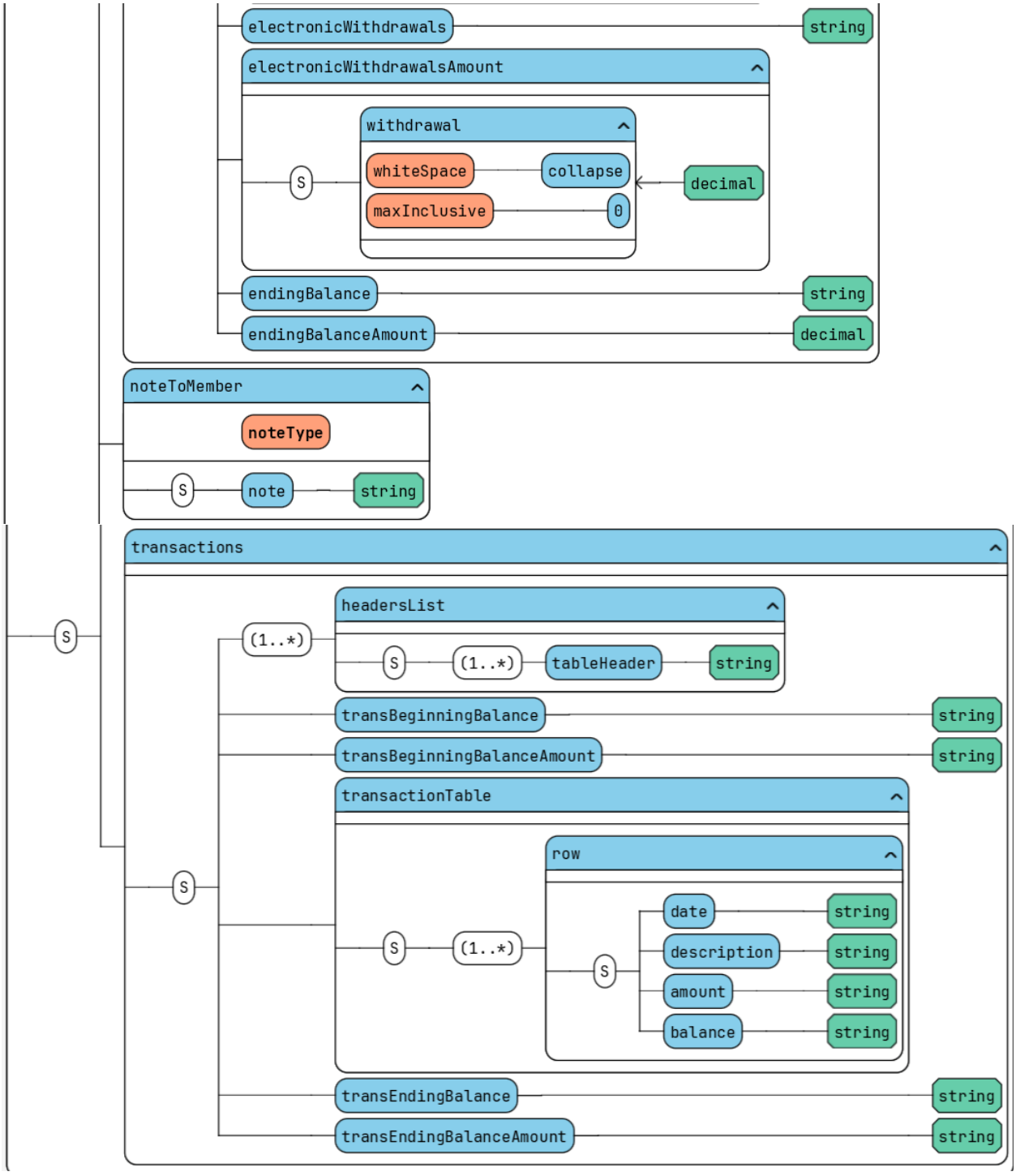
This xml schema is meant to be general to all bank statements or other financial institutions account summary statements. This does not mean there are pros and cons to the xml schema itself. One large benefit is that this document is exhaustive in the sections in which the schema is created for. Any institution, at the end of a statement period, wants to communicate the start and end values of the account and document how the changes occurred. An executive summary is placed at the start for ease

of understanding, followed by the detailed breakdown of transactions, purchases, and other expenses that caused the account balance to change through the statement period. This xml schema covers all of that, with the bonus of customization so that it is not exclusive to one financial institution. The largest con however is the fact that the xml schema is not exhaustive outside of what should be expected in a message to member. The xml schema as it stands is sufficient for the bank for simple calculations, but not exhaustive for further means of analysis that can be inferred from the data. For example, there is no address element that can be used to store the member's residence for organization. There is no phone number element for use of the member or the financial institution. Fortunately, these can be added to the schema quite easily, as xml schemas support the modification and formatting curatorial objectives, but as it stands these are the critiques of the schema.

Additional Note: When validating the schema online using Liquid Oxygen, I had to remove the 1st line from the .xsd and/or .xml files. When doing this, the validation worked, and without the validation failed.


accountStatement
currency
language
summaryHeader
accountSummary
summary
whiteSpace
preserve
Checking Summary
Savings Summary
Certificate of Deposit Summary
Money Market Account Summary
Individual Retirement Accounts Summary
E
string
S
accountTypeDesc
accountType
S
desc
string


balanceBreakdown
balanceBreakdownHeader
string
beginningBalance
string
beginningBalanceAmount
decimal
deposits
string
depositAmount
deposit
whiteSpace
collapse
decimal
minInclusive
0
S
physicalWithdrawals
string
physicalWithdrawalsAmount
withdrawal
whiteSpace
collapse
decimal
maxInclusive
0
S
S
S


electronicWithdrawals
string
electronicWithdrawalsAmount
withdrawal
S
whiteSpace
collapse
decimal
maxInclusive
0
endingBalance
string
endingBalanceAmount
decimal
noteToMember
noteType
S
note
string
transactions
S
(1..*)
headersList
S
(1..*)
tableHeader
string
transBeginningBalance
string
transBeginningBalanceAmount
string
transactionTable
row
S
(1..*)
S
date
string
description
string
amount
string
balance
string
transEndingBalance
string
transEndingBalanceAmount
string