1.
- <u>One-page concept statement</u>:

The financial planning application provides financial management utilities for customers via a mobile application interface. Customers should be able connect existing financial tools to the application which generates analysis and graphical interaction. The application will collaborate with Banking, Credit, and Investment institutions that hold accounts of our target customers. Banks and other institutions will process transactions and maintain accounts on their own; the application will access this information through externally. The customer can use the application for personal or institutional purposes with the same result. The application is for customers who have financial assets and liabilities in many places and need a single place to manage them.

The application will enable users to view bank accounts, credit accounts, and investment portfolios in a single point of entry which expedites and organizes the process of evaluating a financial position. Along with creating a single access point for all of a customer's financial utilities, the application provides functionality to programmatically create a budget based on evaluation of bank and income statements. After analyzing a bank statement required by the Budget component, the application will be able to create a spending tracker and cash flow statement. Customers will be able to set up weekly, monthly and yearly bill reminders. These will be utilized by a Calendar component that reminds customers of upcoming and overdue bills.

The application will be used by individuals and organizations alike. The user interface is provided for mobile phones and tablets as an application downloaded from a phones app store. The app can be used at home, the office, or on the go without sacrificing functionality.

The application provides a solution for the need to consolidate financial assets and liabilities into a single point of access. The ability to create and manage budgets is necessary for those living on a fixed income or anyone looking for a better way to manage their money. The application is intended for day-to-day use and can be utilized at any point in time. Customers' financial well-being changes every day, this application serves to keep users up to date with those changes.

The application will work by providing a user interface through phones and tablets. The application system will collaborate with Bank, Credit Union, and Investment Firm systems to compile financial information needed for the UX. Each collaborating system (Banks, Credit Unions, and Investment Firms) will hold and maintain individual accounts on the customers behalf but will allow access to the confidential data after login credentials have been supplied.

- Conceptual Domain Model:
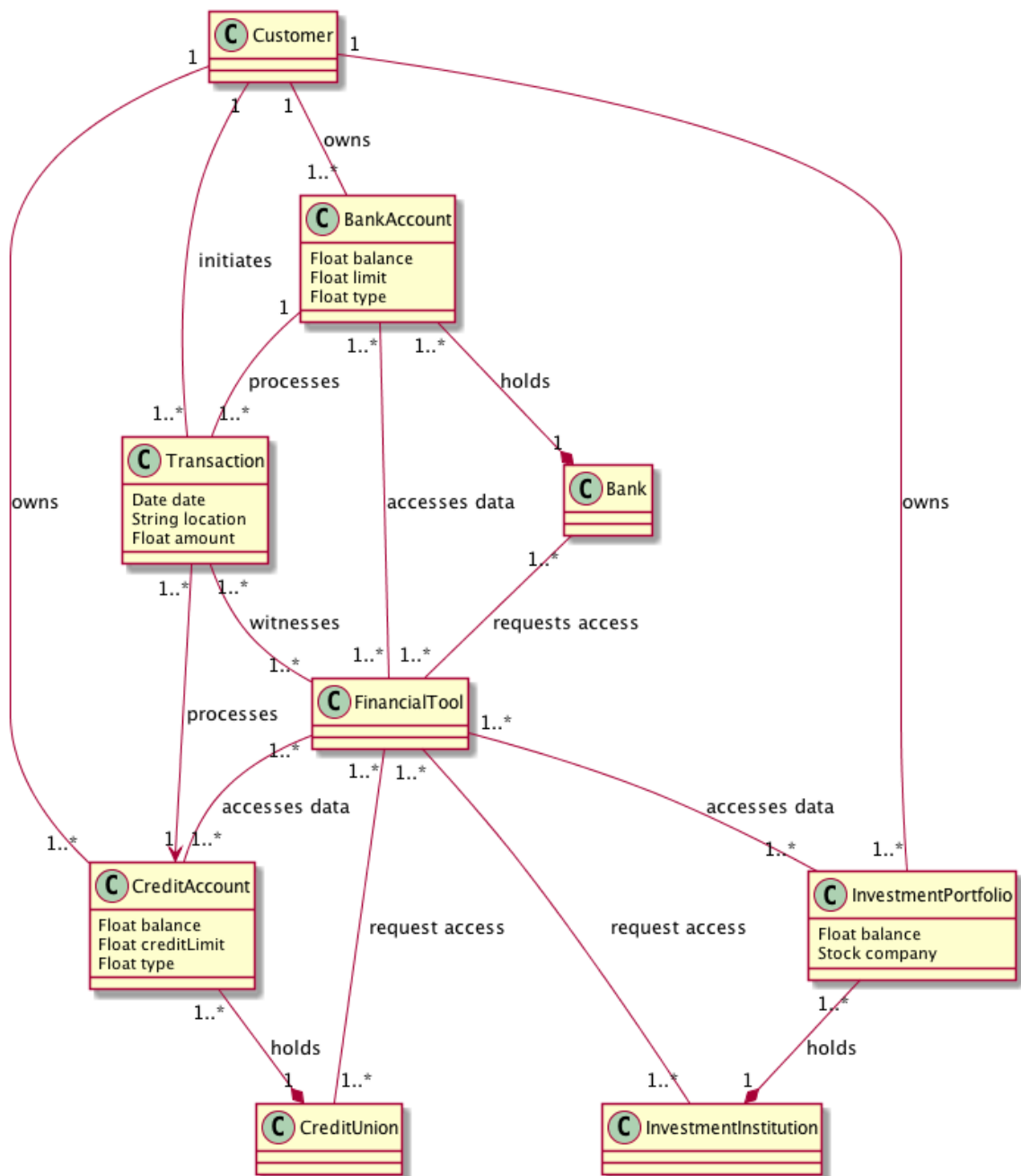
Several other class components were debated after linguistic analysis was performed, they are as follows:

Financial Asset: Vague and redundant

Firm system: Redundant

Confidential data: Redundant

Login Credentials: Can be an attribute
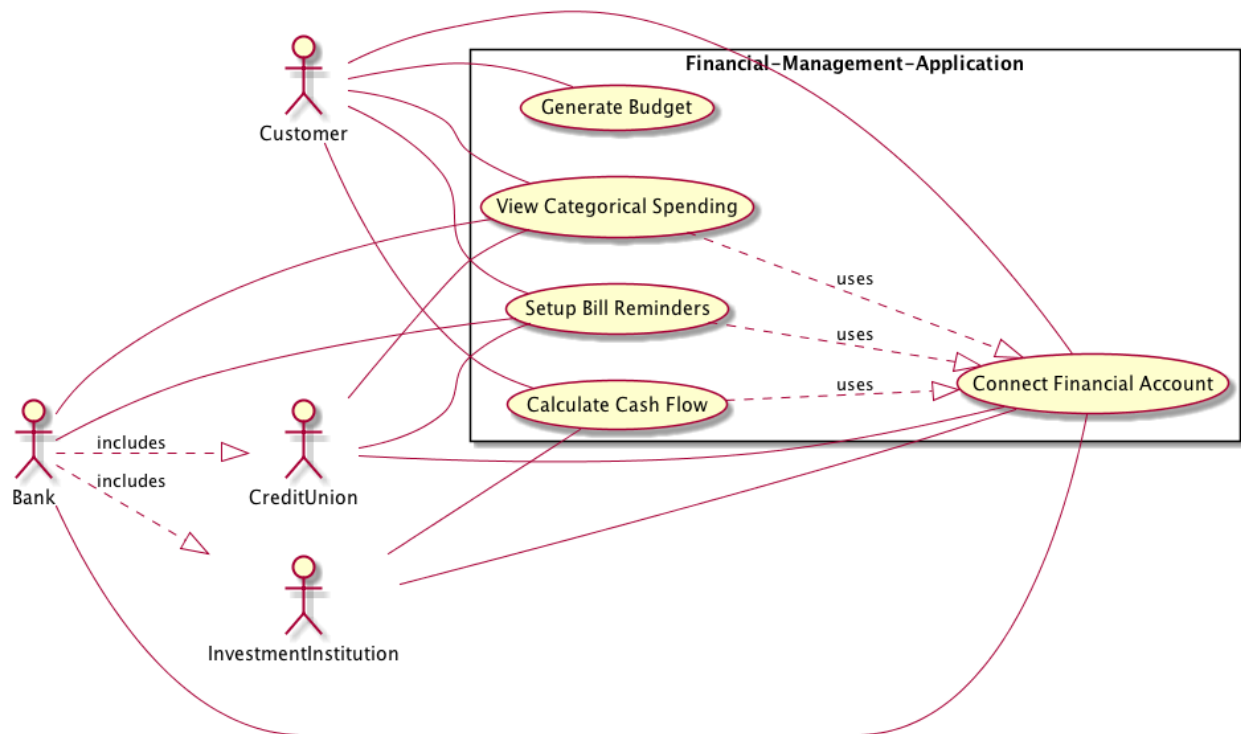
App Store: Irrelevant at the moment

- OCL Constraints/Invariants:
  Context FinancialTool **inv**:
  self.transactionRecords.length > 0

  Context FinancialTool **inv**:
  self.accounts() > 0

  Context FinancialTool **inv**:
  self.accessPermissions > 0

2.

3. Detailed Text Description of a single use case

Pre-condition: User has set up app account
Actors: Customer, Bank, Credit Union,
Investment Institution

1 Customer initiates account connection
2 Customer selects type of account
3 Customer enters account credentials

4 Bank/Credit Union/ Investment Institution
verifies account credentials
- If credentials are wrong, return to 3.
5 Bank/Credit Union/ Investment Institution
queries account records
6 Bank/Credit Union/ Investment Institution
consolidates records into standard format
7 Bank/Credit Union/ Investment Institution
sends records to application system

8 Application system stores account
credentials in secure database
9 Application system stores account records
in secure database
10 Application system updates UI to reflect
new account

4. Scenarios & High-Level System Sequence Diagrams

Scenario 1:
Customer selects to connect a new financial account
Customer determines account type
Customer enters login information
Bank or Institution verifies account credentials
Bank notifies application that credentials were correct
Application stores account credentials
Bank accesses account records
Bank consolidates records into transportable format
Bank sends account records to system
Application stores account records
Application updates UI

Scenario 2:
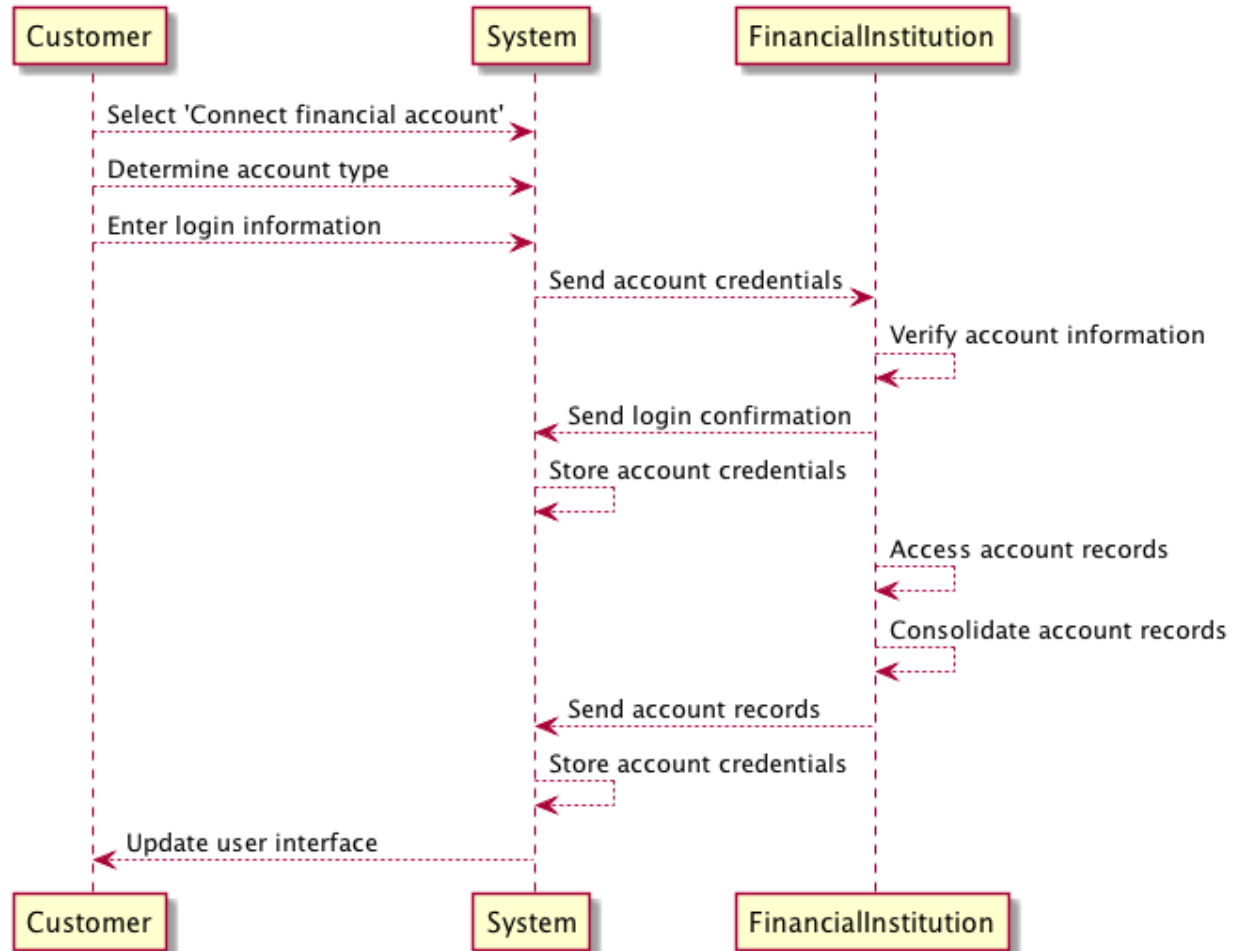Customer selects to connect a new financial account
Customer determines account type
Customer enters login information
Bank or Institution verifies account credentials

Bank determines account credentials incorrect
Bank notifies application system of login failure
Application prompts user to enter credentials again
Repeat

High-Level SSD (for Scenario 1)

High-Level SSD (for Scenario 2)

Customer	System	FinancialInstitution

Customer → System: Select 'Connect financial account'

Customer → System: Determine account type

Customer → System: Enter login information

System → FinancialInstitution: Send account credentials

FinancialInstitution → FinancialInstitution: Verify account information

FinancialInstitution → System: Send login failure

System → System: Update login attempts left

System → Customer: Display login failure

System → Customer: Return to home

Customer	System	FinancialInstitution