Money Hub

Modest Software Engineering Project

Software Requirement Specification

Created by:
Sam Dressler & John Neis
CSCI 463- Software Engineering
4/1/2020

Table of Contents

1. Revision Summary

Revision	Name	Description of Change	Date
1.0	Sam Dressler	Initial report design and layout Introduction sections Report Description sections Project Requirement Specification	3/29/2020
	John Neis	Requirements Models -Use case, state chart, relationship diagram	3/29/2020
1.1	Sam Dressler	Added additional requirements and description for the product perspective model.	3/31/2020
	John Neis	Descriptions for models Server requirements Final touch ups to format	3/31/2020

2. Introduction

2.1 DOCUMENT PURPOSE

Money Hub, henceforth known as the *project*, is the product whose requirements will be specified in this document. The requirement specification shall cover the current functional and non-functional requirements as well as any design or process constraints.

The product is split into two main partitions, the Money Hub Client, henceforth known as the *client*, and the Money Hub Server, henceforth known as the *server*. The four specifications noted in the above paragraph will be specified for each of the partitions.

In addition to the specifications, this document will contain detailed models that will model the data and behavior of the system. These models will include a state chart and use case diagram. Additionally, an entity relationship diagram will be included to show a rudimentary connection between the user and the data in the system.

2.2 PRODUCT SCOPE

Money Hub was launched with the goal in mind of simplifying the user's financial situation. The goal of creating Money Hub was to give users access to a product that allows them to improve their financial competency.

Money Hub will provide the user with access to a variety of their accounts in one place. Having a mirror of their checking, and savings accounts, allow the user the benefit of being able to track their spending and saving.

Where Money Hub goes beyond your typical banking website, is its access to showing the user their investment portfolios being used in online investment firms. On top of that, one of our goals in creating the system is too be able to see what debts the user has in car, student, and other various loans.

Creating a comprehensive summary of this information in one place that the user can have easy access too will achieve the objective of increasing the financial intelligence of the population of the United States.

2.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW

The rest of this document contains a further description of the project and its environment as well as limiting factors such as constraints and dependencies. Additionally, this document will provide requirement specifications and certain models detailing the usage of the system.

The main purpose of this document is to present the customer (Dr. Hassan Reza Ph.D.), with the contents summarized in the previous paragraph. This document shall be considered a living document throughout the development phase as the requirements shall be influenced and modified based on feedback from the customer.

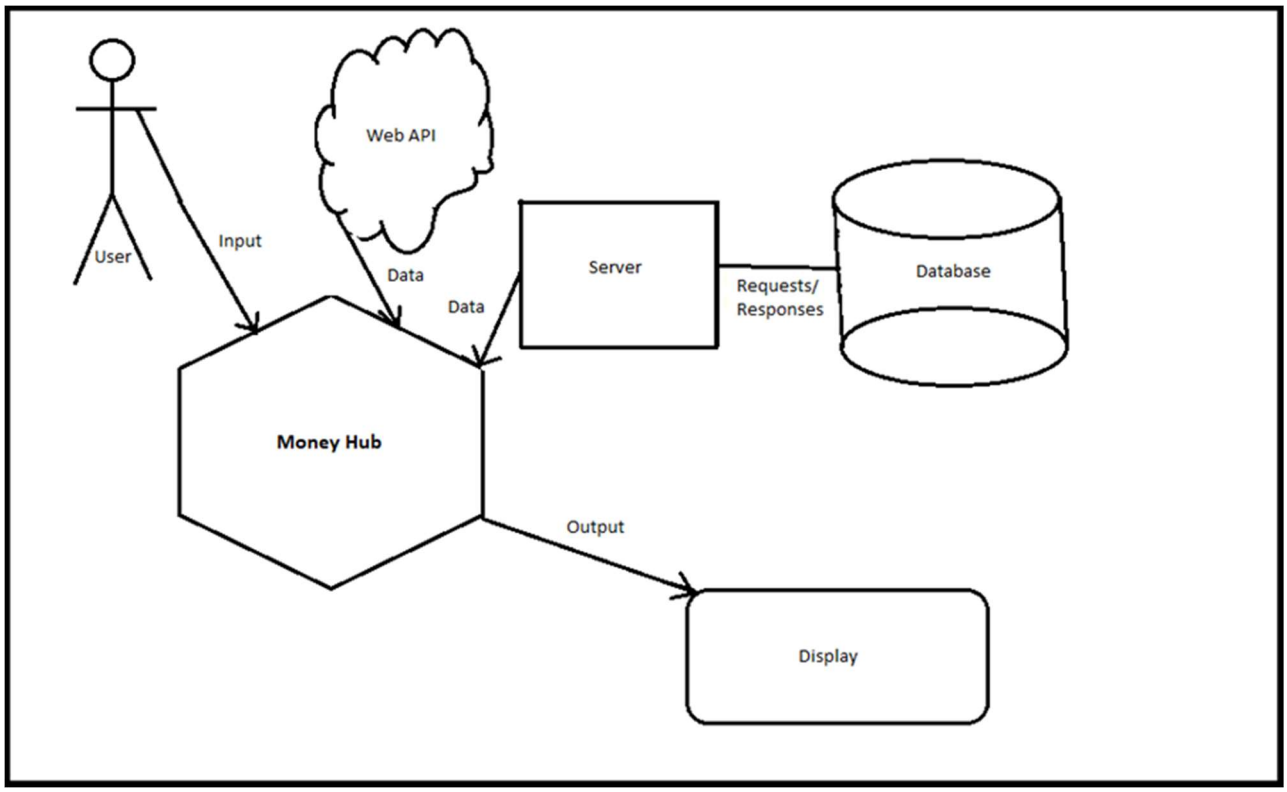
While the customer is the main motivator for writing this document, the contents can also be used by various other parties. The next important of these will be the people who are charged with the actual development of the system such as the software developers and testers. For those parties, the critical sections to read are found in section three. Readers in those categories can jump to that section to begin learning the specific specifications of the system as well as to begin studying the information and data flow in the models.

For readers in the categories of customer, project managers, marketing staff, or documentation writers, continue reading into section two. This section will give a broader description of the project that will help in further work that will help in marketing, planning risk, or writing insightful documentation. Lastly, the information from section two will make understanding the final section easier to comprehend.

3. Project Description

3.1 PRODUCT PERSPECTIVE

Money Hub is a product that was envisioned as an entirely new system. In the information age, nothing is more important than having the facts. This is especially true in our own lives. Opening an app and seeing all the important financial information is something that our team believes is crucial to survive in today's fast-moving world.



This model shows how the user will interact with the Money Hub system which is tied to external components. The web API and Server both pull information from sources that hold information for the user. Once the data is entered into the system, the processing will be executed and output to the user through the user interface on the display.

3.2 PRODUCT FUNCTIONALLITY

This system runs concurrently by connecting to a server through client applications. Additionally, there will be multiple web-based API's that allow our third-party software to extract account information from banks and lenders and feed it into the client.

First time users will create an account that is then stored on the server before being logged onto the Money Hub "Home" page. Here the user will see what the web APIs return to the client. This information will include the amounts for the accounts that the user has connected. At the same time, the information loaded from the web is then stored into that user's account information on the server.

For users to have the balance of an account show up in the client, they will need to log into the respective company's website through the client. Connecting to the accounts this way ensures that Money Hub never directly sees the user's password for an account but still has access to the amounts they are trying to see inside of client application.

Returning users will log into the client and have their credentials validated through the server. Once into the client, the application will begin updating the balances shown in the application.

Users will be able to sync their Money Hub account with accounts from other firms by logging into the respected accounts through the client.

3.3 USERS AND CHARACTERISTICS

The usability of the system will be heavily considered when designing the user interface. The reason for this is so that whether the user is a recent college grad just gaining a real income for the first time, or a Millennial looking to get their finances in check, the experience will be one of ease.

Our team wanted the characteristics of the program to be one that is attractive and clean. Many times, applications are cluttered or difficult to follow from page to page. In designing a page like this, it will make sure that you don't need to be a computer expert to use the system.

Reading the financial information will also be a breeze because all the amounts will be visible on a single page. Additionally, analytics performed on the account information will make it clear as to what the user's financial situation is so that they can decide what their next correct step is.

3.4 OPERATING ENVIRONMENT

We live in a world where we are constantly on the move. In order to keep up with the pace of our lives, we need to develop a product that can be accessed on the go. To achieve this goal, The Money Hub application will eventually be developed to be used on the web, by browsers such as Google Chrome, Microsoft Edge, or Mozilla Firefox.

Currently the system will be a simple desktop application due to the short-handed development phase length and the lack of knowledge in our team in the area of web-based development. The limitations will be elaborated further on in clause 2.7.2. The operating system that the prototype will be able to operate on will be Windows 10.

3.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

During the development process of a prototype for Money Hub, there are a few challenge areas that would need to be overcame in the development of a full fledge system.

1) **External Data Availability**
   Access to external accounts takes time and permission. For the development of a prototype like this we have neither, so the data being fed into the system is non-existent. To develop the project fully, time and energy would need to be invested into getting firms such as banks and lenders to allow us access to their data through an API. Another constraint that comes from volatile data availability is the time it takes to access the data from the external firm.
   While this process may seem daunting, there are already proofs of the concept out there. Many personal financial management tools are out there that implement the connections to firms that would be needed in this product.
2) **Data Security**
   There are a few different levels of security that will be necessary. For example, the client will need to log in but where will the data used in validating the account be stored? In order to provide security for the user's information, that validating information will need to be stored in a database on a server rather than on the client.
3) **Portability**
   Currently, Money Hub is a desktop application. Since we have a limited time frame to develop the prototype, we had to choose an operating system to develop on. For us we chose Windows 10, which means we won't have time to make sure that all the functionality works on another host running Linux or MacOS.
4) **Internet Access**
   Because the client sync's new financial data from the web and validates its logins via a server, if a user is trying to log into the client without internet access, then they will not be able to access their account.

3.6 USER DOCUMENTATION

Utilization of the application and the service held within were designed so that the user would have minimal issues when learning to use the system. The user interface will be designed such that it is self-explanatory. However, one area where problems may arise is syncing an external account. To account for this, help will be available to the user via an internal help button near the option to add an account.

3.7 ASSUMPTIONS AND DEPENDENCIES

3.7.1 assumptions

During the planning for the development of this system, certain assumptions had to be made. These assumptions are those came during the preliminary brainstorming of the system and more may be added further into development.

1) The account information for all the user's that are added during the prototyping phase will be the same. These amounts will need to be hard coded into the prototype so that they can be loaded into the client once they log in.
2) The instance "web API" means all the API's that would be required to retrieve account information from banks, investment firms, and loan venders.
3) In the prototype, the web API will return the hard-coded values for a few accounts that will be decided during the design phase.
4) The application will be build using Java, C#, and an SQL database with the interface being designed using Visual Studio.

3.7.2 Constraints

The development process for the project will also face constraints on its development.

1) Time and effort to complete the prototype is limited. The entirety of the development process will take place over just a couple of months so complexity of the system will consider such constraints.
2) The system prototype will not be able to interface via an API with online firms. While this concept is possible, there must be a period of communication and development between organizations in order to ensure the security, legality, and feasibility of such ventures.
3) The money being displayed in the system for a given user will have not actually exist in the real world.

3.7.3 Dependencies

The development and implementation of the full Money Hub system will depend on the cooperation of partner firms in order to give the user access to the information that they desire. Additionally, once the system is fully operational, it will depend on these firms giving access to account information on a regular basis with low downtimes.

4. Project Specification Requirements

Requirement Specification Key

Token	Description
FR	Represents a functional requirement.
NFR	Represents a non-functional requirement.
<#ID>	three-digit number indicating the ID of the requirement.
<#ID>C	Indicates the requirement involves the client.
<#ID>S	Indicates the requirement involves the server component.
<#ID>SE	Indicates the requirement involves the security of the system.

4.1 CLIENT USER INTERFACE REQUIREMENTS

The user interface is one of if not the most crucial components of the system. This interface will contain multiple pages that allow for the user to access the remainder of the system and its functionality.

ID	Component	Description
FR001C	User Interface	The client shall validate credentials and log a user into the system in less than five seconds.
FR002C	User Interface	The client shall display the options to exit and minimize the program on every page, including the login page.
FR003C	User Interface	The client shall display the option to log out after a user has been logged in successfully.

FR004C	User Interface	The client shall correctly navigate to a new page when a button is clicked.
FR005C	Client-Server Communication	The client shall send a user account login ID and a password, ideally encrypted, to the server.
FR006C	Client-Server Communication	The client shall receive and display account information from the server.
NFR007C	User Interface	The client shall be able to connect to the internet and provide timely connection to the partner firms that the user has accounts with. Loading of this information and storing to the database shall not exceed 10 seconds.

4.2 SERVER REQUIREMENTS

The Server will be the means by which the client will communicate to access the database. The server will be implemented using the Java programming language.

ID	Component	Description
FR001S	Database	The database shall record user account login IDs, password, a list of accounts and balances, and ages of the respective accounts.
FR002S	Client-Server Communication	The server shall query the database using login information received from clients in order to generate a login token.
FR003S	Server	The server shall sanitize any data received from the client, in order to avoid SQL injections.

4.3 SECURITY REQUIREMENTS

The client and the server will both have aspects that would be disastrous if they were not secure. Client's will trust the platform to ensure the safety of their information for their external accounts.

ID	Component	Description
FR001SE	Security	The client will not track account credentials to external firms.
FR002SE	Security	User login credential validation will be performed on the server side of the application.
NFR003SE	Security	Users of the Money Hub shall authenticate themselves using a username and password that meet the requirements for passwords.
FR004SE	Security	User's password will consist of a minimum of eight characters. The password shall contain a capital letter, a number, and a special character.

5. System Specification

5.1 DATA MODELING

5.1.1 – Entity Relationship Diagram

Shown below is a rudimentary model, demonstrating the relationships between relevant data that will be used. A user will consist of account number, password, first name, and last name attributes, with account number and password being candidate keys. An account will consist of account number, balance, and type attributes, with the account number being the primary key. Here, a single account can only be held by a single user, however a single user may hold many accounts.

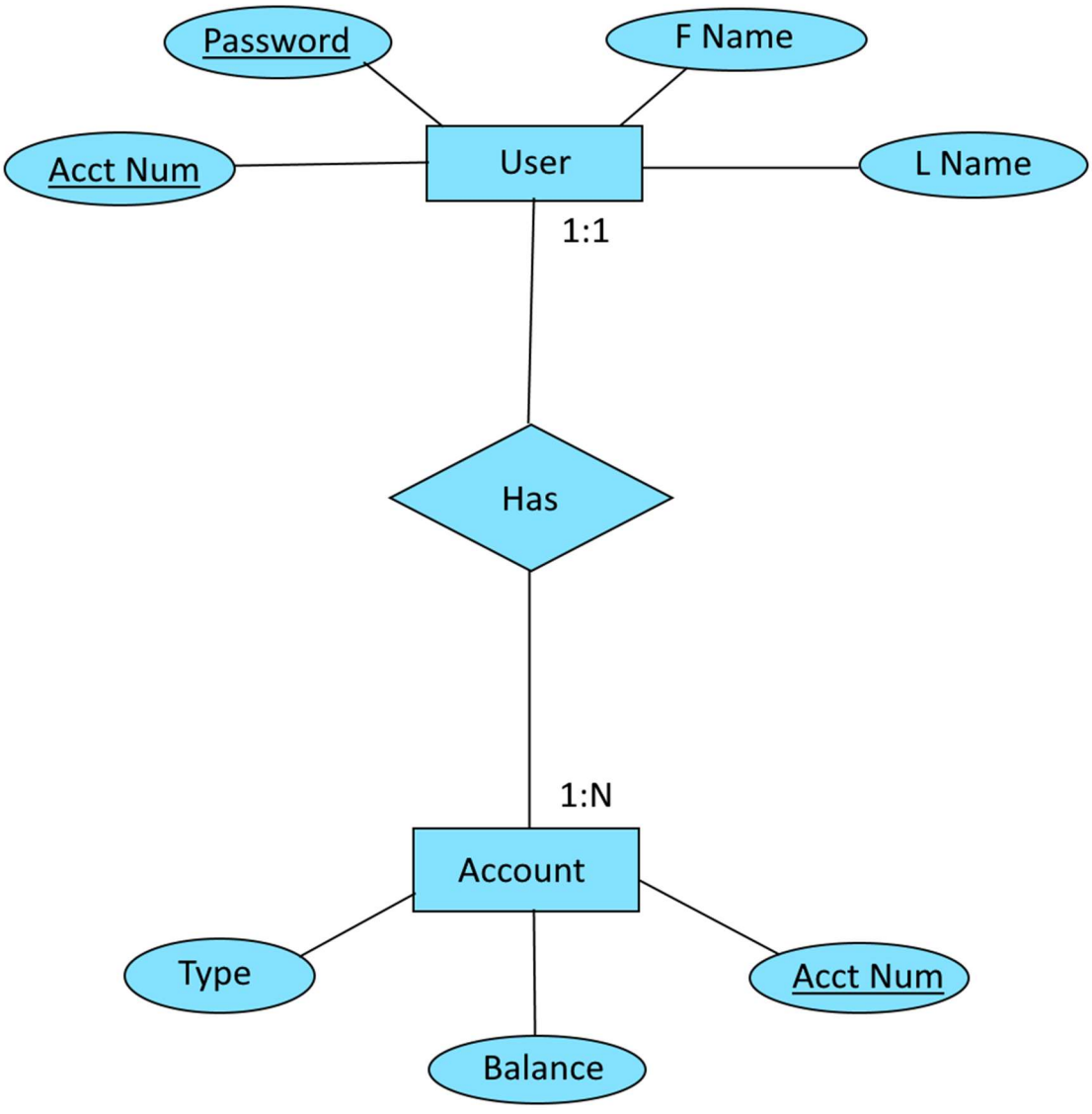


Fig. 4.1.1 – Shows how the data of relevant to the database is structured

5.2 BEHAVIORAL MODELING

5.2.1 State Chart

Shown is a state chart which is meant to describe the program flow. The system consists of three major components: the client, the server, and the database. Initially, the server will be in a standby mode as it listens for incoming requests from clients. The client will initially show a login screen. When the user logs in, the request must be verified by the server, which will query the database in order to determine whether or not the login information is good. At this point the server will send back a login token to the client. If the token is good, the client can display the information returned by the server. If the token is bad, and error message will be displayed, prompting the user to re-enter their login credentials.

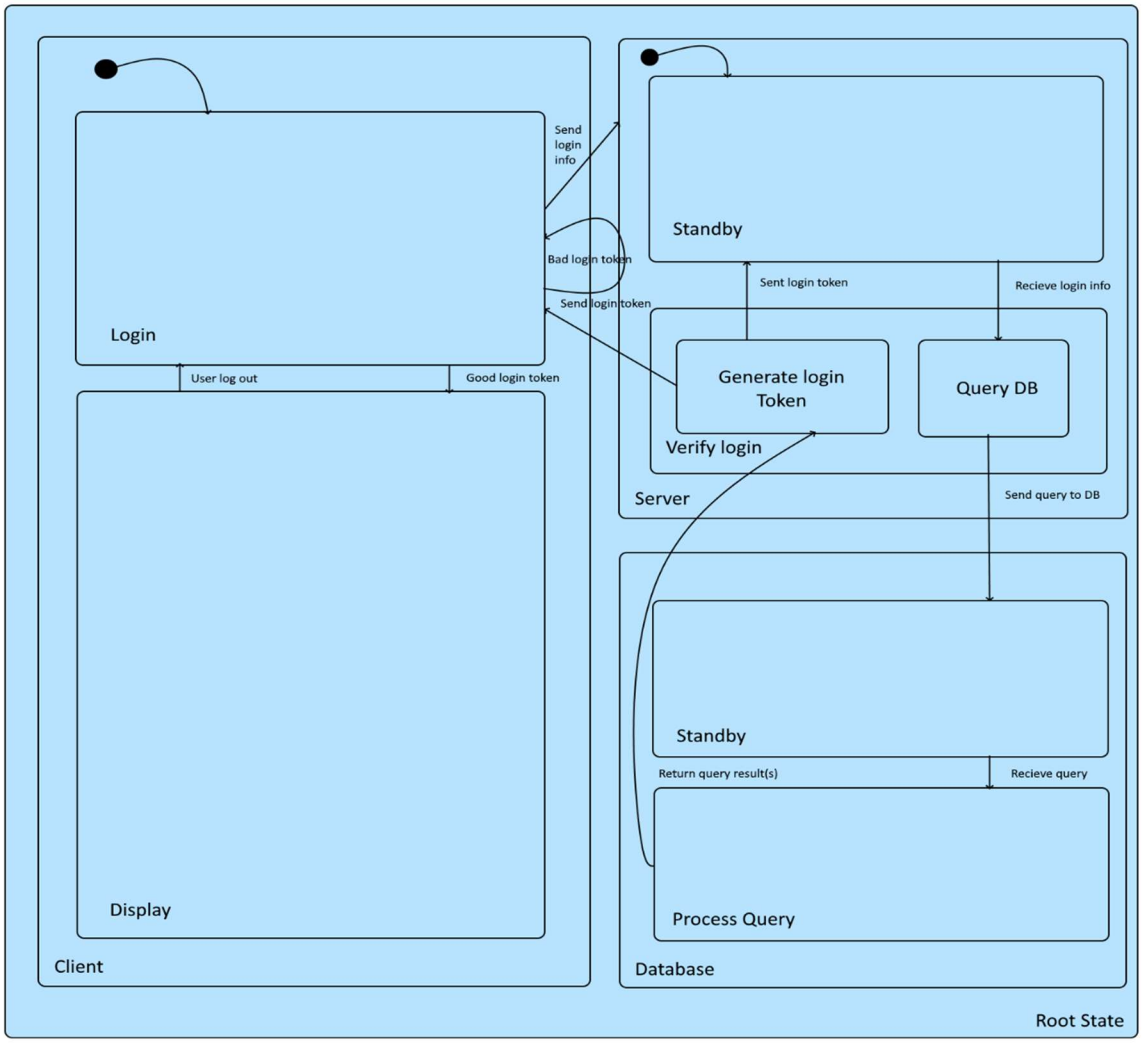


Fig 4.2.1 – The state chart diagram demonstrates the flow of control and data throughout the execution of the program. Here, the client and server are treated as separate states, as one is active while the other is inactive. This, however, can be extended to include multiple concurrent sessions between multiple clients and the server.

5.2.2 Use Case Diagram

Shown is a use case diagram, meant to illustrate the various cases and users that might utilize this system. Clients will be able to view their account information, freeze or unfreeze use of any payment cards tied to their accounts, wire money from one account to another, and monitor transactions they make on those accounts, for example.

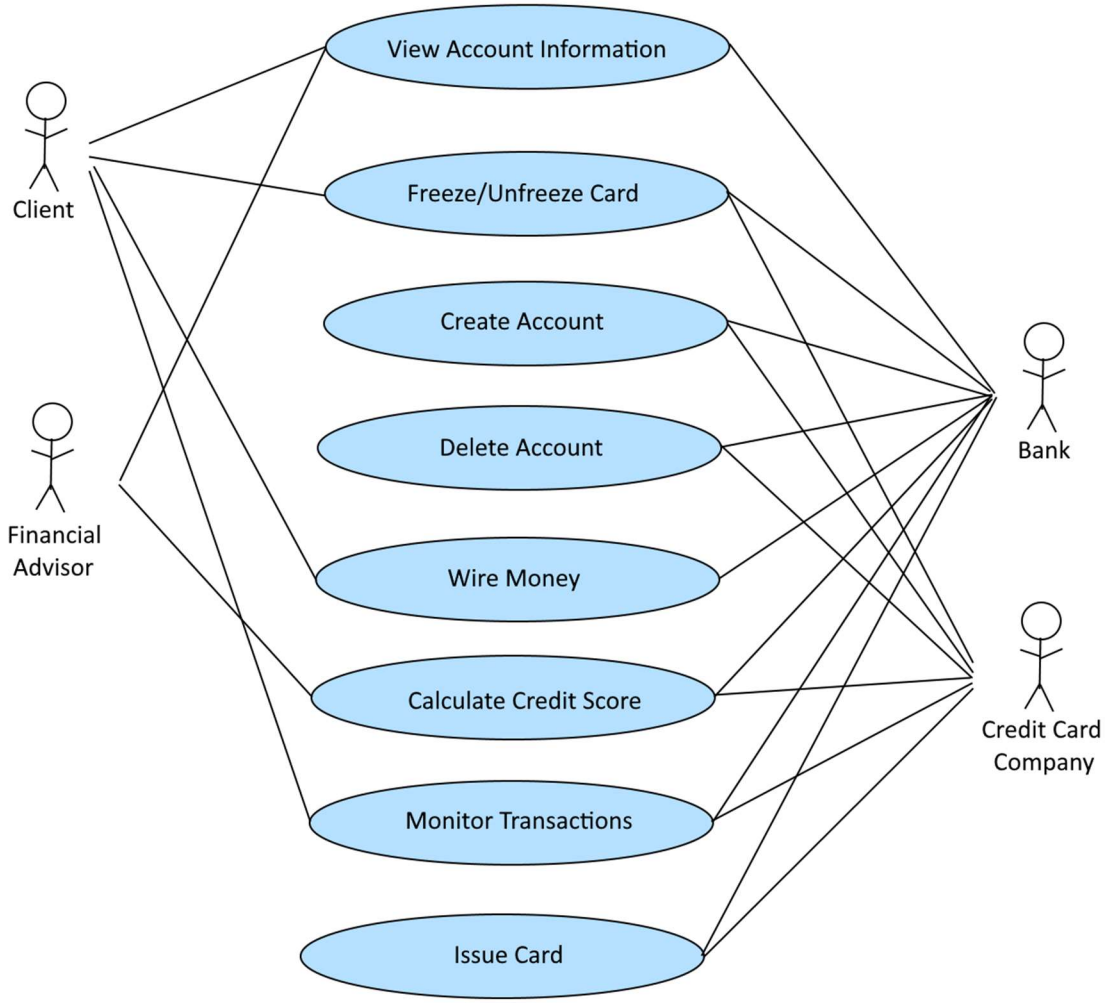


Fig 4.2.2 – The Use Case Diagram demonstrates the various uses the system will provide. Throughout the development process, more use cases may be discovered, so this may not be a totally comprehensive visual.