Cash Stash

Financial Management Application

Team Members

Cameron Kubik – cmk0037@auburn.edu – Team Leader

Sean Ramuchak – str0005@auburn.edu – Designer

Dylan McCardle – dcm0033@auburn.edu – Programmer

Daniell Yancey – dby0001@auburn.edu – Requirements Engineer

Zachary Hayes – zgh0001@auburn.edu – Architect

Professor Yilmaz

Software Modeling and Design

March 19, 2018

Table of Contents

1. Domain Analysis
   1. Concept Statement…………………………………………………….
   2. Conceptual Domain Model…………………………………………
   3. Domain State Model…………………………………………………..
2. Application Analysis
   1. Application Interaction Model……………………………………
   2. Application Class Model…………………………………………....
   3. Application State Model…………………………………………….
3. Consolidated Class Model………………………………………………….
4. Model Review…………………………………………………………………….

Concept Statement

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.

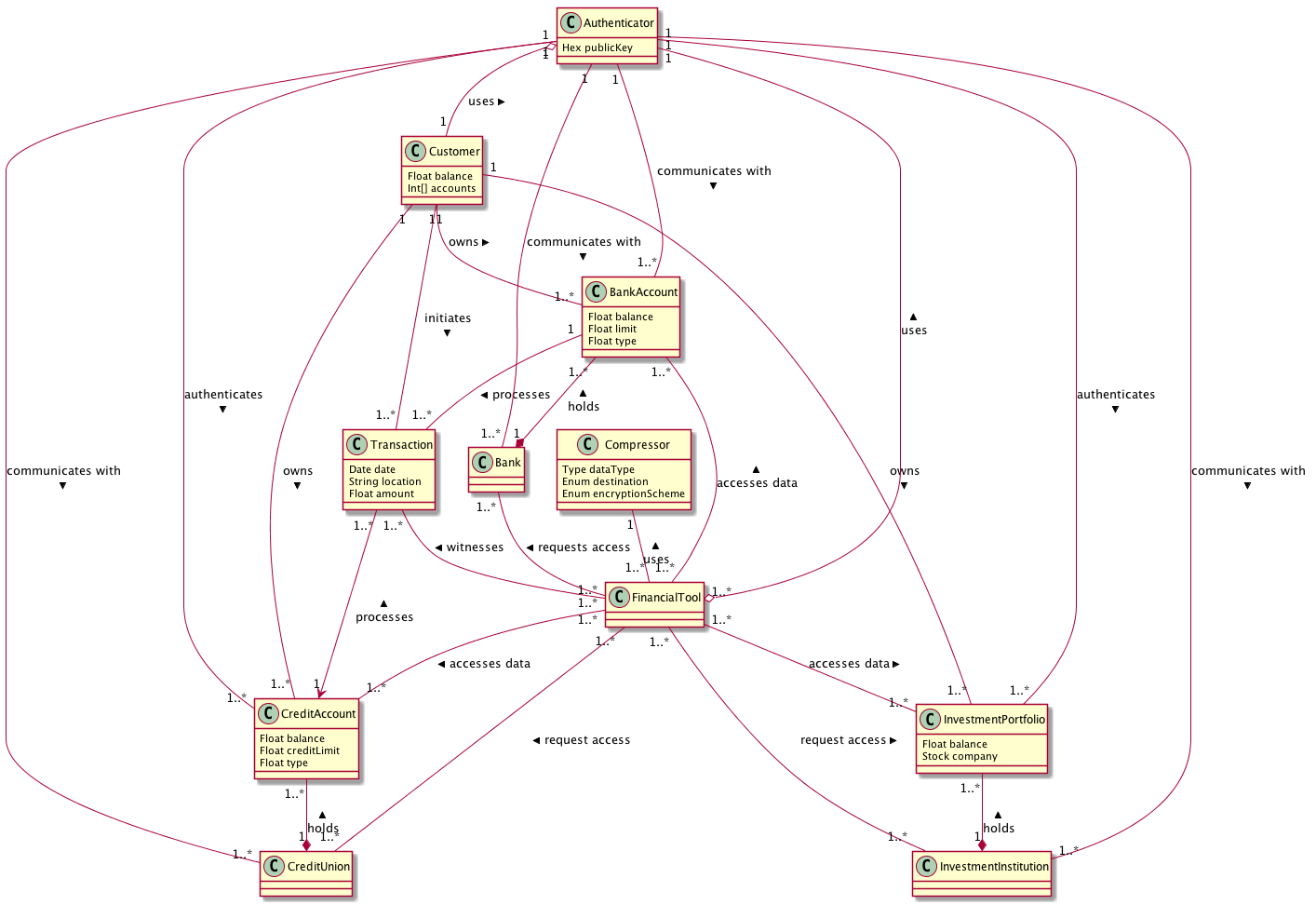
The application will be used by individuals and organizations alike. The user interface is provided for mobile phones and tablets as an application downloaded from a phones app store. The app can be used at home, the office, or on the go without sacrificing functionality.

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.

Concept Statement continues…

Conceptual Domain Model



Domain State Model

Application Interaction Model

Essential Use Cases

|  |  |
| --- | --- |
| **Create Account**  Pre-condition: The user is on the Cash Stash login screen. | |
| 1. User provides account information. | 1. System validates user information. 2. System creates account. |
| Exceptions:  3a. If the account information couldn’t be validated, then the user will receive an error message.  3b. If the password provided is less than 7 characters, then the user will receive an error message. | |
| Post-condition: The user account is created. | |

|  |  |
| --- | --- |
| **Login**  Pre-condition: The user is on the Cash Stash login screen and has already created an account. | |
| 1. User provides login information. | 1. System validates user information. 2. System logs the user in. |
| Exceptions:  2a. If the account information couldn’t be verified, then the user will receive an error message. | |
| Post-condition: The user is logged in. | |

|  |  |
| --- | --- |
| **Logout**  Pre-condition: The user is logged in to Cash Stash. | |
| 1. User chooses the logout option. | 1. System saves any open data. 2. System logs the user out and displays login screen. |
| Exceptions: | |
| Post-condition: The user is logged out. | |

|  |  |
| --- | --- |
| **Manage Profile**  Pre-condition: The user is logged in to Cash Stash. | |
| 1. User chooses the manage profile option. 2. User submits account settings. | 1. System loads the account settings screen. 2. System saves account information and loads the main screen. |
| Exceptions:  4a. If the account settings are missing information, an error message is displayed. | |
| Post-condition: The user’s account information is updated. | |

|  |  |
| --- | --- |
| **Delete Account**  Pre-condition: The user is on the “Account Settings” screen of the Cash Stash application. | |
| 1. User chooses the delete account option. 2. User submits account password. | 1. System prompts the user for their account password. 2. System erases all data related to the user’s account and displays the login screen. |
| Exceptions:  4a. If the user fails the account verification step, an error message will display. | |
| Post-condition: The user’s account and data no longer exists. | |

|  |  |
| --- | --- |
| **Link Financial Account**  Pre-condition: The user is on the “Account Settings” screen of the Cash Stash application. | |
| 1. User chooses the link financial account option. 2. User provides financial account details. | 1. System prompts the user for the account details. 2. System sends account details to financial account service for authorization. 3. Financial account service authorizes the account details. 4. System syncs financial account information with Cash Stash account. |
| Exceptions:  4a. If there is information missing from the account details, the system will prompt the user to finish filling in the details.  6a. If the account details were not authorized, then the system will display an error message to the user and ask them to try again. | |
| Post-condition: The user’s Cash Stash account is now synced with their financial account. | |

|  |  |
| --- | --- |
| **Create Transaction**  Pre-condition: The user is logged in to Cash Stash. | |
| 1. User chooses the create transaction option. 2. User submits transaction details. | 1. System displays new transaction screen. 2. System verifies transaction details. 3. System saves the transaction. |
| Exceptions:  5a. If the transaction verification fails, the system will display an error message instead of saving the transaction. | |
| Post-condition: The transaction is saved within the user’s account. | |

|  |  |
| --- | --- |
| **Schedule Transaction**  Pre-condition: The user is logged in to Cash Stash. | |
| 1. User chooses the schedule transaction option. 2. User submits transaction details. | 1. System displays new transaction screen. 2. System verifies transaction details. 3. System saves the recurring transaction. |
| Exceptions:  5a. If the transaction verification fails, the system will display an error message instead of saving the recurring transaction. | |
| Post-condition: The recurring transaction is saved within the user’s account. | |

|  |  |
| --- | --- |
| **Create Budget**  Pre-condition: The user is logged in to Cash Stash. | |
| 1. User chooses the create budget option. 2. User submits budget settings. | 1. System displays the new budget screen. 2. System verifies budget settings. 3. System displays the budget report. |
| Exceptions:  5a. If the budget settings are missing information, the system will display an error message instead of creating the budget.  5b. If the sum of percentages for categories is less than 0 or greater than 100, the system will display an error message. | |
| Post-condition: The budget report is displayed. | |

|  |  |
| --- | --- |
| **Calculate Cash Flow**  Pre-condition: The user is logged in to Cash Stash. | |
| 1. User chooses the calculate cash flow option. 2. User chooses a timeframe for the cash flow. | 1. System prompts the user for a timeframe. 2. System calculates income and expense transactions for given timeframe. 3. System displays cash flow report. |
| Exceptions:  4a. If the account doesn’t contain any transactions within the given timeframe, the system will display an error message. | |
| Post-condition: The cash flow report is displayed. | |

|  |  |
| --- | --- |
| **Calculate Slippage**  Pre-condition: The user is logged in to Cash Stash and has setup a budget. | |
| 1. User chooses the calculate slippage option. 2. User chooses the timeframe for which to calculate slippage. | 1. System prompts the user to choose a timeframe. 2. System loads budget and calculates cash flow. 3. System displays a slippage report which compares values from the budget to the actual cash flow. |
| Exceptions: | |
| Post-condition: The slippage report is displayed. | |

|  |  |
| --- | --- |
| **Check Credit Score**  Pre-condition: The user is logged in to Cash Stash and has completed their account identity information. | |
| 1. The user chooses the check credit score option. | 1. System will query the credit check service for the user’s credit score. 2. Credit check service verifies the identity information and gets the credit score. 3. System displays the user’s current credit score. |
| Exceptions:  4a. If the credit check service returns an error, the error message is displayed to the user instead. | |
| Post-condition: The user’s credit score is displayed. | |

|  |  |
| --- | --- |
| **Export Data**  Pre-condition: The user is logged in to Cash Stash. | |
| 1. User chooses the export data option. 2. User chooses a save location. | 1. System prompts the user to choose a location to save the file. 2. System will compile an excel file of the account transactions. 3. System sends a request to the operating system to save the excel file. |
| Exceptions:  4a. If the account doesn’t contain any transactions, an error message will display to the user. | |
| Post-condition: The operating system receives the create file request. | |

|  |  |
| --- | --- |
| **Print Data**  Pre-condition: The user is logged in to Cash Stash. | |
| 1. User chooses the print data option. | 1. System will send a request to the printer service to print the current screen. |
| Exceptions: | |
| Post-condition: The printer service receives the print request. | |

|  |  |
| --- | --- |
| **Refresh Data**  Pre-condition: The user is logged in to Cash Stash. | |
| 1. User chooses the refresh data option. | 1. System request a sync operation from the financial account service. 2. Financial account service will respond with the appropriate data. 3. System reloads the current screen. |
| Exceptions:  4a. If any of the sync requests returned with an error, these error messages will be displayed to the user before reloading the current screen. | |
| Post-condition: The user’s Cash Stash account now contains any data in linked financial accounts. | |