

---

# MyMoney App

---

COMP354

PB-PJ

## **0.1 Justification of Use-Cases and Rational**

### **0.1.1 Drop down menu - categories**

Data organization is a useful feature for financial applications. Many users have their own set of demands on what they need the product for. By using drop-down-categorization it allows the user to save space and time in their experience in using this software. Customization also allows users to feel as if they are part of the product and allow them to tailor the experience of using the software for their own personal needs. Limitations are in the form of what can be customized and what can not. For example, you can not set new data fields to see, you must stay in the designated fields

### **0.1.2 Statistics**

Users may need additional tools to conceptualize their financial situation. All banking systems include statistical data on your investments, this one does too. Some users may need averages to get a rough idea of how much they spend in a year while others just want to know how much they spent on average. Regardless of your financial competency, statistical information is a must have on all financial applications. While the data may not be complex it is a must have to include at least a limited set of statistical functionality.

### **0.1.3 Styling (CSS)**

User experience is enhanced by good design. Interactions must be intuitive to the user via color, patterns, fonts and effects all drawing attention to what the user should be clicking on. Good design must take this into account and make the features of software easily discoverable and enhance the feedback given upon sending the software a message from beyond the simple OS and native level feedback. Customizability of styling would be nice, however in some ways it is unnecessary if the one main template is designed well.

### **0.1.4 Deleting Transactions**

Users will inevitably need to make edits to the data contents. In order to do this, there must be a way to communicate with the database. In terms of giving the user control over his or her own experience then we must trust that the user knows when it's best to remove traces of his financial history from the logbooks. This use-case allows for that. While there are certain dangers inherent with letting the user delete his own

data, it's assumed that a person using financial software has the good judgement not to take excessive risks.

#### **0.1.5 Account Transactions**

This software wouldn't be much of a financial application if the users could not see their own data in the form of account transactions. This feature allows for users to navigate the view and click on systems to view their transaction history. Since the system already connects to the database to make insertion and delete requests, it is only natural that the user be given access to see his entire transaction table. It's assumed that the user wants to see all of the data so that freedom is given to him.

#### **0.1.6 Enter a Transaction**

Entering in information for transaction is essential to building a database. The user is given four fields of data to add to a transaction since it's assumed that these are all he needs. Name, type, amount and description make up all the data a user can submit for a transaction since it's assumed that things like, interest rate, credit limits or anything more complicated is best left for the user to calculate himself. Also it is inherent that every transaction is marked by a unique ID number that is not set by the user himself.

#### **0.1.7 Edit an Existing Transfer**

Users will inevitably make mistakes, when they do they will need to edit their data. It is assumed that a user can make errors on any field except for the ID therefore they are given the freedom to modify any of this data at any time from a list of all transactions they have made. Assuming they remember the name of their transaction, they can find it in a list and edit it.

#### **0.1.8 Add a Bank Account**

Setting up an account is the first thing a user must do when running this software and the most fundamental. If a user does not have an account then he can not use the software or access the database. Accounts are given four identification fields. Your name, your bank account number, the type of account used and the balance. Upon saving the data a table is produced in the database holding your account info. The assumption is that the user can enter any kind of bank account type he wants and edit it as he sees fit. There are more possibilities of what the user can do on the application if everything is left as a simple template he can fill

out and not limit the choices that can be done by giving out radio boxes or select bars.

#### **0.1.9 Set Saving Goals**

Since saving goals are a main reason why an individual might use a financial planning application this feature is included in the product. All people who save money are doing it for a reason and typically it is assumed that they want to save a certain amount of money by a certain date in time. This use-case allows for that by giving them a form to fill out specifying the account balance they want to achieve by a certain date in time. Since the user is in need of keeping track of his spending against the goal, this saving's goal is presented on the account page to allow the user to constantly be reminded by it. This does not take into account things like spending goals, or witstrain goals since that is considered practices that should not be encouraged ethically by a piece of financial designed to help with spending.

#### **0.1.10 Generate Monthly Report**

Users will want to know how they are doing every month. A simple report displaying informations on their spending, saving and frequency of use of bank accounts is the end goal of this product and is represented in the monthly reports. While it aims to offer a comprehensive guide of what the user has done with their money, it does not allow for multiple account views. It is the monthly report of a single account. While a limitation it is understandable since sometimes more than one account could use a computer's software and viewing their information is a breach of privacy. Finally it is important for a user to check how he is doing against his set saving goals. While no bar graphs, pie charts and histograms are generated, hopefully the user can be satisfied knowing he has met his goals and that he should continue using the software.