

Financial Management Application for UTM CSCI 352 Fall 2017

Lucian Freeze, Brett Whitson, and Adam Chisolm

Abstract

This project is a graphical application which allows a user to view and manage financial accounts and budgets. The target user for the project is primarily a college-age student; however, the application should be useful to anyone who wishes to use it. The intent is to provide said users the ability to manage finances in an easy-to-use, versatile environment.

1. Introduction

Students like (read: "need") to save money. Having a simple and intuitive application to manage ones finances would make this much more achievable. This application intends to serve that purpose by giving the user the tools needed to achieve this goal.

A section to keep track of accounts and balances is to be implemented, allowing a user to view this information and add/remove accounts with possible customizations. Users will also be able to enter and edit transactions, dynamically updating the initial balance.

Another section for budget creation and editing is also to be implemented, allowing users to track how their money is spent based on categories.

A reports section will be implemented to allow users to visually examine their transaction data within a certain time period.

1.1. Background

Inspiration for this project came from the stereotypical financial struggle of a college student. We wanted to create an application that might possibly help those in this situation.

1.2. Challenges

Possible challenges for this project include implementing graphical components and managing information's security within the application. The "Reports" section is to have elements such as dynamic graphs, which we are not yet certain of how to implement. Information security on a real-world scale is also a challenge we have yet to face, so this could prove to be difficult.

2. Scope

Our project aims to develop a simple, operable program in which users can securely and easily manage their finances. When we have fully implemented account and budget functionality along with some aesthetic aids and security measures, the project will be at completion.

Additional goals for this project include incorporating real-time investment profiles and a possible implementation of remote server-based account and credential storage/management.

2.1. Requirements

2.1.1. Functional.

- Users need user accounts – each user needs to only be able to access his/her personal data/settings and no one else's
- Users need financial accounts – checking and savings accounts will need to be implemented and able to be created/edited by the user
- Users need budgets – budgets will need to be implemented and able to be created/edited by the user
- Users need transaction functionality – user will need to be able to edit and add transactions per account; possible database will be needed to created
- Users need reports – opt-in; gives user a weekly/bi-weekly/monthly representation of spending in categories; bar/pie graph form

2.1.2. Non-Functional.

- Security – user credentials and account information must be encrypted, users should be able to reset their passwords if forgotten
- Data integrity – user and especially account data must be reliably stored and accessible for successive use

2.2. Use Cases

Use Case #: 1

Use Case Name: Create User Account

Description: A user would like to create an account within the application. They will click on a "Create User Account" button. This event will transition the user to a new page where they can enter their user information and confirm their account.

Flow:

- 1) User left-clicks on "Create User Account" button on the initial page.
- 2) User is presented with a new page to enter user account details such as username, password, etc.
- 3) If username is not taken, user account is created and the Dashboard page is shown. Otherwise, new username must be chosen.

At Termination: The user now has a user account and access to the rest of the application's utilities.

Use Case #: 2

Use Case Name: Create Financial Account

Description: A user would like to create a financial account within the application. They will click on a "New Account" button. This event will trigger a new page to appear, asking the user which type of account to create. The user clicks the appropriate button and is presented with another page to enter account details.

Flow:

- 1) User left-clicks on "New Account" button on the Dashboard's Accounts tab.
- 2) User is presented with a new page with "Checking" and "Savings" buttons. User left-clicks the appropriate choice.
- 3) A new page is presented which allows the user to edit details for the chosen account type (name, balance, etc.)

At Termination: The user now has a financial account of the chosen type.

Use Case #: 3

Use Case Name: View Budgets

Description: A user would like to view their budgets within the application. They will click on "Budgets" tab. This event will trigger the "Budgets" page to appear. The user can then view their budgets.

Flow:

- 1) User left-clicks on the "Budgets" tab.
- 2) User is presented with the "Budgets" tab/page.

At Termination: The user is able to view their budgets.

2.3. Interface Mockups



Figure 1. This image is a mockup of the initial sign in/up screen for our application.

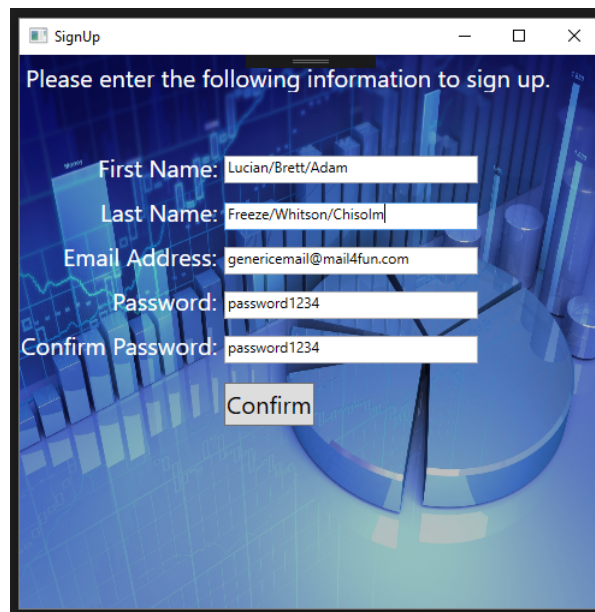


Figure 2. This image is a mockup of the sign up page for our application.

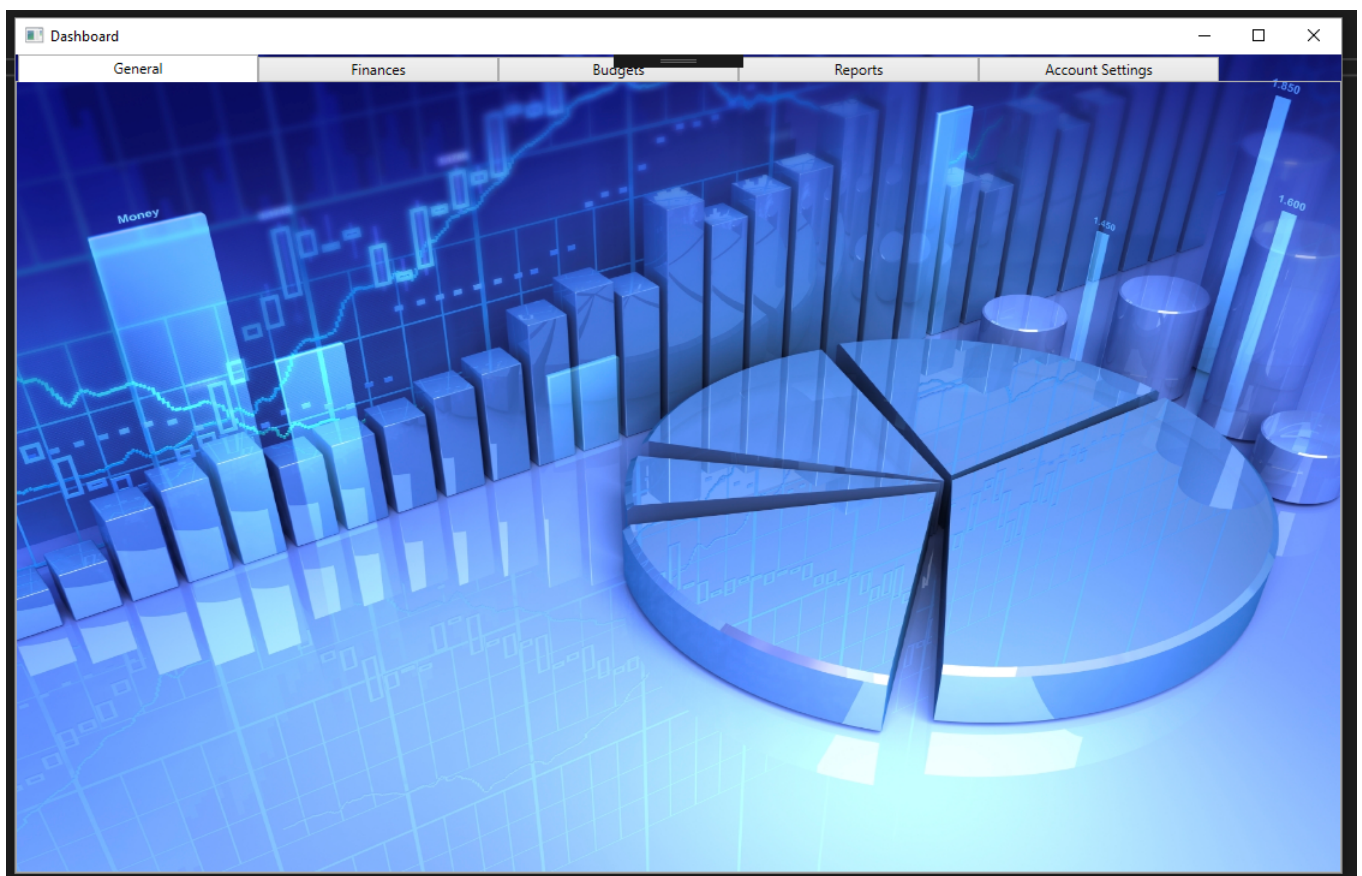


Figure 3. This image is a mockup of the Dashboard (the main user interface) for a logged-in user.