Financial Management Web Application

Raymond Lawson

CST-452 Capstone Project Requirements Document

Grand Canyon University

Instructor: Professor Bradley Mauger

Revision: 1.6

Date: 12/9/2023

**ABSTRACT**

The capstone project is a financial management web application. This web application will assist the average individual with managing their personal finances. Features of this application are expense tracking, income highlighting, goal setting, and overall budget assistance. Simplicity is very important for the design of this application. This application differs from other financial applications currently on the market because it does not focus on cramming as many features as possible and instead on its ease of use. This document is the Final Plan document. This document outlines the final technical requirements, contains a summary of the design planning, the design concepts, a detailed description of the system architecture, and the normal revision and sign off sheet.

## **Final Plan**

Financial Management Application

Prepared by Raymond Lawson

Contributors Raymond Lawson

|  |
| --- |
| Design Planning Summary |

This development project is a financial management web application. The system contains many features such as expense management, goal setting and monitoring, budget assistance, and income monitoring. Financial information has been lacking in the public education system in America for a long time. This results in adults that graduate from high school and go into the world without the proper knowledge of how to manage their finances. This application aims to provide an easy to use, simple interface that is not too overwhelming to the average person while still providing the essential features. With this application, users can manage their personal finances, never miss a expense, set goals and save money for those goals without breaking the bank, and set a budget that is appropriate for them.

|  |
| --- |
| Overview of Design Concepts |

Screenshots

Home Page

A screenshot of a computer

Description automatically generated

Register Page

A screenshot of a computer

Description automatically generated

Home Page

A screenshot of a computer

Description automatically generated

Expenses Page

A screenshot of a computer

Description automatically generated

Add Expense Page

A screenshot of a computer

Description automatically generated

Income Page

A screenshot of a computer

Description automatically generated

The rest of the pages have a similar format.

Code

The following screenshots are the classes and a snippet of their code.

Login Controller

A screen shot of a computer

Description automatically generated

Home Controller

A screenshot of a computer program

Description automatically generated

ExpenseDao

A screenshot of a computer program

Description automatically generated

UserDao

A screenshot of a computer

Description automatically generated

IncomeDao

A screenshot of a computer

Description automatically generated

Expense Date Update

A screenshot of a computer

Description automatically generated

Income Date Calculation

A screenshot of a computer program

Description automatically generated

Login Page

A computer screen shot of text

Description automatically generated

Registration Page

A screenshot of a computer program

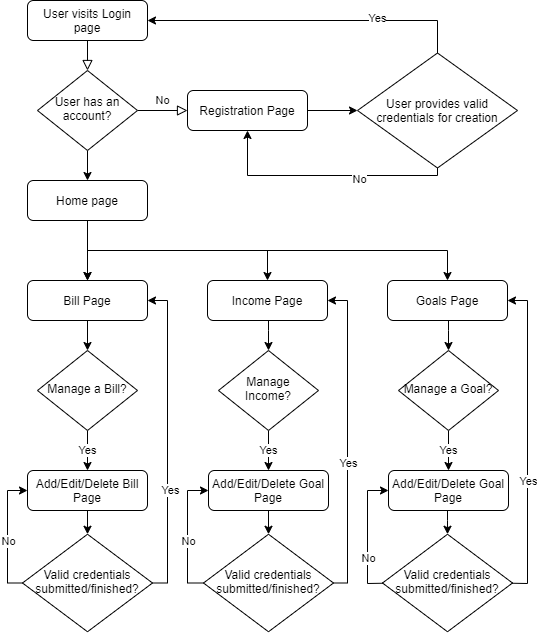
Description automatically generated

Home Page

A screenshot of a computer

Description automatically generated

Flowchart



Process Flow

When the user visits the application, they will land on the login page. If the user has an account they will log in, if not they will navigate to the registration page. If they have an account, they will log in and be directed to the home page. The home page will show their goals, income, and expenses. There are two expense tables, current paycheck and next paycheck where the users expenses are appropriately contained within each table. The home page also shows the users balance, the balance of expenses left to come out this pay period, and the total when that number is deducted from the current balance. The system will automatically remove the expense from the current paycheck list on the appropriate date, deduct the amount from the users balance, and increment the expense by the chosen interval and type. When visiting the expense page, the user will be able to add, edit, and delete any items on their list. There are add, edit, and delete pages for each feature as well. From any page, the user can navigate back to the home page where everything is displayed.

There are no stakeholders, the design concepts and flow are satisfactory for the application.

1. Use the template to list the project deliverables.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Deliverable Acceptance Log | | | | | |
| ID | Deliverable Description | Comments | Evaluator (internal or external as applicable) | Status | Date of Decision |
| 1 | Goals Feature | Goals page, add/edit/delete goal pages, goal credential validation, DB table creation. | Raymond | Scheduled development on 11/6/2023 | 11/4/2023 |
| 3 | Income Feature | Income page, add/edit/delete income pages, income credential validation, DB table creation. | Raymond | Scheduled development on 11/08/2023 | 11/4/2023 |
| 4 | Linking Feature | Link expenses to their specific income (paycheck) | Raymond | Scheduled development on 11/09/2023 | 11/4/2023 |
| 5 |  |  |  |  |  |

|  |
| --- |
| Detailed Solution Architecture |

The proposed design aligns well with the project structure, at this time there are no incompatibilities. The design displays all user expenses, divided into current paycheck and next paycheck. Everything is automatically updated and deduced by the system. Models in the project are User, Income, Expense, and Goal. These models have respective tables in the database. The user object is to save user information such as account information, expense list, income list, and balance. The Income object is used to save income information. The Expense object is used to save expense information. The Goal object is used to save goal information.

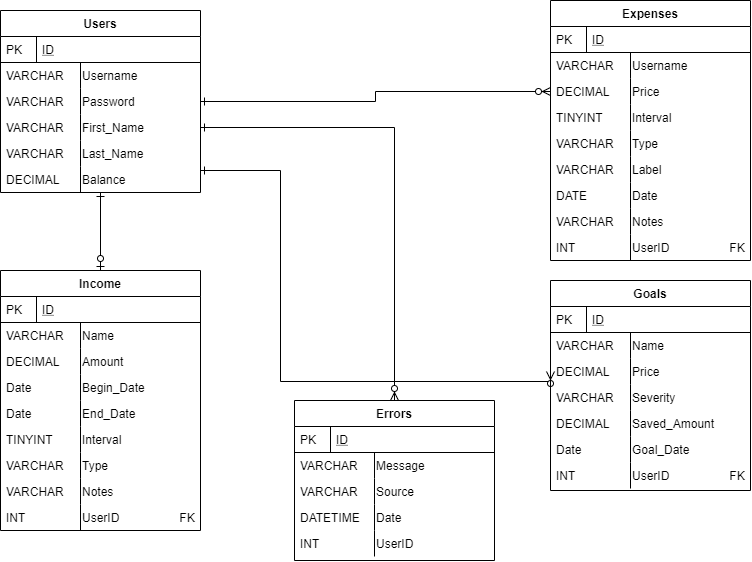
Data Dictionary Link: <https://docs.google.com/spreadsheets/d/1eVTZJ-ayBaNEEmcm1Att03EuOcuBrOja/edit?usp=sharing&ouid=110853295913392877928&rtpof=true&sd=true>

Business Logic

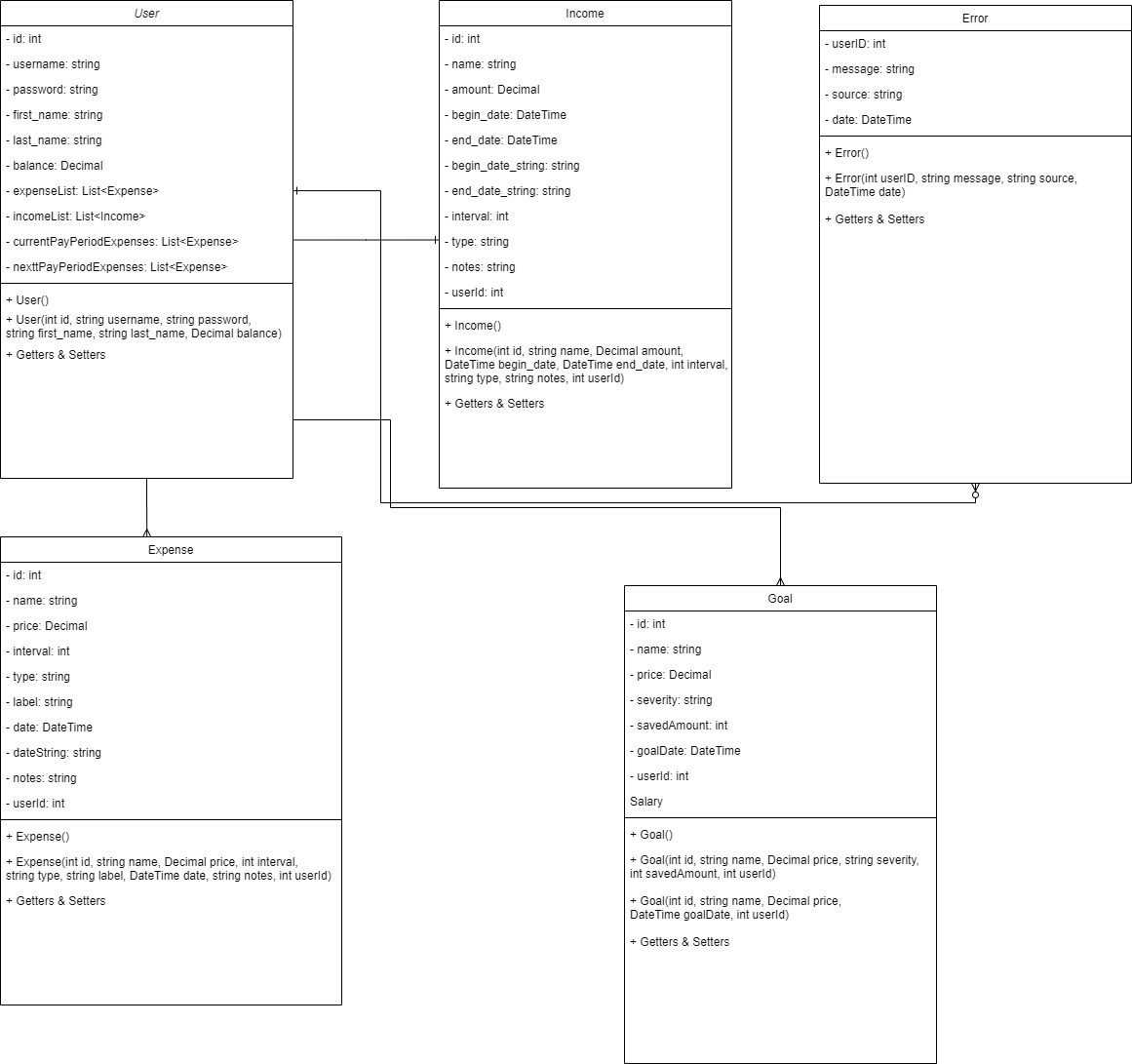
ExpenseDateUpdate – This class has three methods (Update, UpdateCurrentPayPeriod, and UpdateNextPayPeriod). The main method (Update) takes in a User object, parses through the users expense list and for each expense it compares the due date with the systems current date. If the due date is less than the current date it increments the expense by the appropriate interval. This repeats until the expense’s due date is in the future. When this happens it also deducts the price of the expense from the users balance. This allows the user to not have to worry about manually deducting expenses from balance and manually updating expense due dates. After this the method calls the other two methods (UpdateCurrentPayPeriod and UpdateNextPayPeriod). The UpdateCurrentPayPeriod method takes in a user object, grabs the income and looks at its begin and end date. Every expense in the users expense list is added to this currentPayPeriod list and is returned to set the users currentPayPeriod list. The same thing happens with next pay period, with some calculation to determine the dates of the next pay period.

IncomeDateCalculate – This class handles setting the incomes end date. The user only needs to provide the income start date and the interval at which they get paid and this method will use that information to determine the last day of the paycheck. TODO: Automatically update the income dates upon the next day after the last day of the income.

Database Design Diagram



Object UML



|  |
| --- |
| Hardware and Software Technologies |
| 1 – MySQL (Relational) Database – I am using a MySQL database which includes a schema, tables, and rows/columns on those tables. |
| 2 – MySQL Workbench is the IDE for accessing / managing the database. |
| 3 – ASP.NET and C# is the framework and language used to create the application. |
| 4 – HTML/CSS/Bootsrap/Razor Pages – These make up the foundation of the front-end of the application. |
| 5 – Microsoft Visual Studios 2019 – This is the IDE the application is developed with. |
| 6 – Google Chrome – This is the web browser used to test and analyze the web application. The browser developer tools are indispensable. |

|  |
| --- |
| Revision and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Editor** | **Revision Notes** |
| 11/4/2023 | Raymond Lawson | Final Plan draft. |
| 12/9/2023 | Raymond Lawson | Final Architecture review before final submission. |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |