Personal Budget Tracker

By:

Farzad Afshari

Mike Shoss

Ryan Jayakody

Sulaiman Quresh

Table on contents

**Project Overview 1**

**Project Design: Wireframe2**

**Project Design: Class Diagram3**

**Project Design: Sequence Diagram4**

**Project Design: Tiers5**

**Deployment Guidelines6**

Project overview

**Project Brainstorming**

* *The Team*

Ryan Jayakody : Developer, GUI Designer

Mike Shoss : Developer, Concept Design

Sulaiman Qureshi : Developer, Tester

Farzad Afshari: Developer, Integration Coordinator

* *Proposed Ideas*

- Personal Budget Tracking System

- Student Grading System

- IT Ticket Tracking

- Phone Book

- Password Keeper

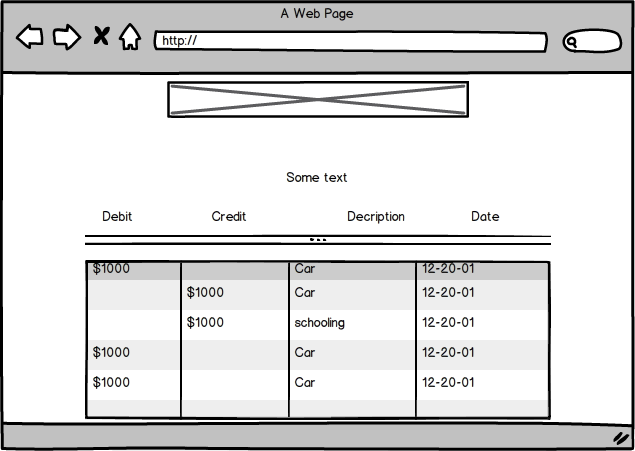
**Project Overview**

The Application our group decided to create is a “Personal Budget Tracking” Application. The purpose of this application is to aid any user with their budgeting needs. As a student it is extremely difficult to keep a track of our money and the idea behind this project is to create a simple to use effective budgeting tool that is also easy to use. Our project will utilize a database, glassfish web server.

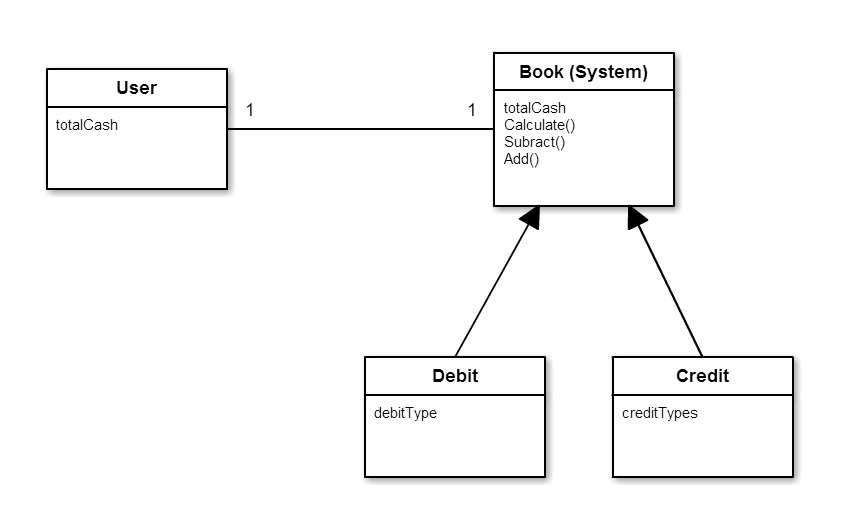
**Functional Requirements**

Our application will start by asking the user how much cash he/she has in their account. Below this field will be a t-chart setup, debits on the left and credits on the right. There will be fields for the user to add an expense or an income. A list of debits and credits will accumulate for each entry. This way, the user will be able to see all credits and debits that were done!

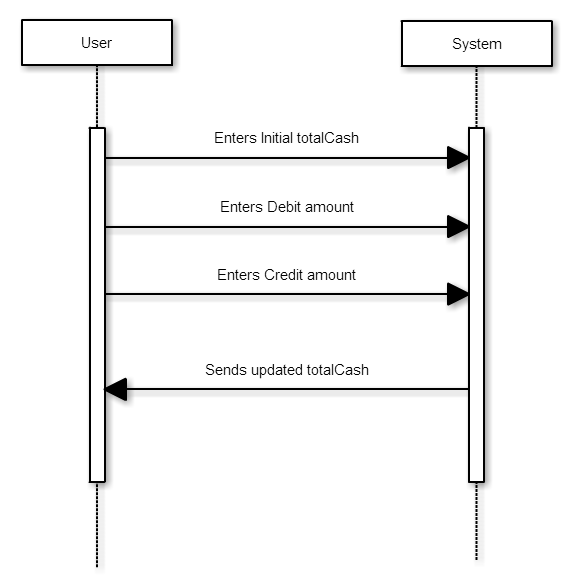
**Project Design:** **Wireframe**



**Project Design: Class Diagram**



**Project Design: Sequence Diagram**



**Project Design: Tiers**

* **User**
* **Category:**
  + Debits
  + Credits
* **System:**
  + Transaction
  + LoginServlet
  + AddCategoryServlet
  + AddTransactionServlet

Deployment Guidelines

Step 1 – Import Project

Step 2 – Open index.jsp

Step 3 – Run Glassfish, If not started

Step 4 – Click Run (Run Program)

Step 5 – Follow on screen instructions

a. Detailed instructions on how to run your project on another machine

b. What to load, where, what to configure, etc.

c. Note: I will be following the steps in this section to run your project and deploy on my

Eclipse installations. Please ensure they work by trying them out on a group member’s

(or any other) computer by starting with a fresh install of Eclipse.

Project Implementation

The project utilises the MVC model, with Model being POJO, View being JSP and the controllers being Servlets.

Model

Class Transaction.java –

Code: A POJO that contains all debits and credits for the user.

Class BudgetUser.java –

Code: A POJO that contains all user entered transactions.

View

HomePage.jsp –

Code: Home page where user enters main info.

SystemPage.jsp –

Code: Page where you enter new transactions.

TChart.jsp –

Code: Page where all the debits and credits are shown.

Controller

AccountSetupServelet.java –

Code: Creates instance of BudgetUser

Classes: AccountSetupServelet

SubmitTranionServelet.java –

Code: Submits Transaction

Classes: SubmitTranionServelet

Developing the front end and the POJOs where the easiest parts for this project, because POJOs where just basic java and we knew what had to go in each one. Also the front end was easy because we had good knowledge in HTML and designing.

Over all this project was simple but just took a long time to do it, it was straight forward. The only problem our team faced our team was debugging our program on many different occasions where information was not passing from controller to the view.

We have no incomplete or unfinished work, but if we had time we would continue to keep developing aspects we would work on using a login page and use a database to have more than one user. We would also add the ability to edit the credits and debits for mistakes.

Screen Shots