AccountAble:
Asset Management Software for the Modern Age

Developed By:
Data Over Sata

Table of Contents

System Documentation for: AccountAble

- INTRODUCTION

  Accountable is a software solution for the tracking and accounting of the Computer Science Department's cashflow at the University of Montana. Currently, there is a paper system in place that is cumbersome and hard to manage, and this software aims to alleviate the burden of managing these accounts.

Purpose

Accountable is intended to ease the process of tracking money moving in and out of the accounts for the Computer Science Department. To do so, AccountAble provides a system where workflow is optimized and user-friendly.

Development Team and Roles

David Rich

- *Scrum master, Chief developer*

Derek Hickman

- *Documentation Engineer, QA engineer*

Clayton Tallwhiteman

- *Developer, User-Experience Manager*

Cordell Appel

- *Developer, Resource Manager*

System Overview

Basic System Design

Below is a UML Diagram of the project. It shows the inheritance of the model and controller classes and their relationships.

MVC Architecture

This program uses the Model-View-Controller Architecture to simplify the design process. Our project has all the classes in individual folders for the corresponding types. Models are in the Model folder, Views in the View Folder and Controllers in the Controller folder. By using fxml files for the view, it forced us to use the MVC architecture, as the views themselves have no code in them. Rather, the xml files are imported, and another class controls all inputs and the model handles the output.

Functional Requirements

- Administrative login (csadmin, csci323)
- Logout

- Developed by at the bottom
- Accounts
    - Balance, Name, Description, Phone, Email
    - Create Acct
        - Must fill all fields
        - Shouldn’t be able to create 2 accounts with same name
    - Modify Acct
    - Delete Acct
        - Warning: You are about to delete acct
        - Can’t delete linked transactions
    - View Acct
- Transactions
    - Credit Card
        - Date, description, code, amount
        - 4% fee
        - 8% university fee
        - Must fill all fields
    - Check
        - Date, description, code, amount
        - 8% university fee
        - Must fill all fields
    - Expense
        - Date, description, code, amount
        - Must fill all fields
    - Delete Transaction
        - Warning: You are about to delete transactions
- Views
    - All Transactions
    - Transactions for specific accounts
- Every screen should have software name at top
- Every screen should have developed by at the bottom

Non-Functional Requirements

- Entire Account Balance
    - View
        - Overall balance
        - Balance for each acct
        - 8% held
        - 4% held
- Benefits calculator
- Users Guide
- Information is loaded on startup and saved during execution or shutdown
- Won’t accept invalid data in fields

dATABASe design

Database Architecture

We did not use a traditional database for this project. Rather, we opted to use a variety of text files which are written to and read from by the program. These text files are located at Accountable/SRC/data.

System Requirements

Hardware Requirements

- AccountAble requires Java 8, which will only run on a computer with the following hardware specifications:
    - RAM: 128 MB
    - Disk space: 124 MB for JRE; 2 MB for Java Update
    - Processor: Minimum Pentium 2 266 MHz processor

Software requirements

- AccountAble requires Java 8, which will only run on a computer with the following software specifications:
    - Running an operating system in the following groups:

Windows

- Windows 10 (8u51 and above)
- Windows 8.x (Desktop)
- Windows 7 SP1
- Windows Vista SP2
- Windows Server 2008 R2 SP1 (64-bit)
- Windows Server 2012 and 2012 R2 (64-bit)

Mac OS X

- Intel-based Mac running Mac OS X 10.8.3+, 10.9+

Linux

- Oracle Linux 5.5+
- Oracle Linux 6.x (32-bit), 6.x (64-bit)
- Oracle Linux 7.x (64-bit) (8u20 and above)
- Red Hat Enterprise Linux 5.5+, 6.x (32-bit), 6.x (64-bit)
- Red Hat Enterprise Linux 7.x (64-bit) (8u20 and above)
- Suse Linux Enterprise Server 10 SP2+, 11.x
- Suse Linux Enterprise Server 12.x (64-bit) (8u31 and above)
- Ubuntu Linux 12.04 LTS, 13.x
- Ubuntu Linux 14.x (8u25 and above)
- Ubuntu Linux 15.04 (8u45 and above)
- Ubuntu Linux 15.10 (8u65 and above)
    -

References

Git Repository

- The Git Repository for this project is located at https://github.com/davidrich27/DataOverSata
- It includes an executable JAR file to be run on all systems located in the final branch, in

the folder Accountable/SRC.

Product Backlog

- The Product Backlog of this project was managed with the online software Trello. It is located at https://trello.com/b/5QVMRjQt/product-backlog

External References

- JavaFX Documentation
  - https://docs.oracle.com/javase/8/javase-clienttechnologies.htm
- Gantt Project – Tool for managing Gannt Charts
  - http://www.ganttproject.biz/
- Scene Builder- Tool for creating FXML files
  - http://gluonhq.com/products/scene-builder/