98/100

Jason Eide, Asif Zaman, Ethan Morris, Jordan Smith, Jose Hernandez

CSCE 3513 Software Engineering Spring 2015

Abstract

This document contains the user requirements, functional requirements and the non-functional requirements of a Personal Financial System

Project Deliverable

Requirement Specification

Group 8:

**Part I: User Requirements in the form of User Stories**

|  |  |
| --- | --- |
| \*\*(Priority: lower value has higher priority)   1. Real Time Expense/Income Entry | |
| User Story | As a user, I want to be able to enter my expenses in real time so that I can monitor my budget. |
| Description | The user should be able to enter monthly expenses and daily spending’s in order to track their overall saving. |
| Priority value | 1 |
| Ideal Person-Hours | The user should be able to learn this feature in 5 minutes. |

|  |  |
| --- | --- |
| 1. Savings Calculator | |
| User Story | As a user, I should be able to enter a savings goal so that I make a particular purchase in the long term and evaluate the duration needed to meet that goal. |
| Description | If the user desires to save for a big ticket purchase, the user should be able to evaluate how long it will take in order to save up for the item. |
| Priority value | 2 |
| Ideal Person-Hours | The user should be able to learn this feature in 5 minutes. |

|  |  |
| --- | --- |
| 1. Budget Plan | |
| User Story | As a user, I should be able to develop a budget plan that is spanned throughout an indefinite amount of time, via, weekly, monthly, yearly, etc. so that I can plan for any personal goals. |
| Description | The user should be able to plan for the future given their current costs and income in order to evaluate their saving at any point of time. These are only rough estimates and changes given Index 1 and 2. |
| Priority value | 1 |
| Ideal Person-Hours | The user should be able to learn this feature in 15 minutes. |

|  |  |
| --- | --- |
| 1. Income Taxes | |
| User Story | As a user, I want to be able to store all desired documents, such as receipts and other desired income tax documents as well as use a tax emulator to estimate my return so that I can be well prepared for filling my taxes. |
| Description | The user should be able to store important documents that are necessary for filing their taxes. The user should be able to use an emulator that mimics tax filing and see what the estimated return is. This is an optional feature for the user. |
| Priority value | 3 |
| Ideal Person-Hours | The user should be able to learn this feature in 15 minutes. |

|  |  |
| --- | --- |
| 1. Secure Storage System | |
| User Story | As a user, I want to be able to use a secure sign-in so I can keep my documents and personal information safe. |
| Description | A login system must be implemented and connected to a database to securely store information. |
| Priority value | 1 |
| Ideal Person-Hours | The users should be able to make an account in 10 minutes. |

|  |  |
| --- | --- |
| 1. Remote Real Time Updates | |
| User Story | As a user, I should be able to logon from any location or devices so that I can manage my budget in real-time. |
| Description | Information will be stored in a database so that the user can update their information via a web-portal, specified desktop application and specified mobile apps. |
| Priority value | 2 |
| Ideal Person-Hours | The user should be able to learn this feature in 2 minutes. |

|  |  |
| --- | --- |
| 1. Email Alerts for Spending Budget | |
| User Story | As a user, I want to be able to get notification when I am reaching a desired spending amount so that I can be reminded when I am close to my spending threshold. |
| Description | The user should be able to justify when an email alerts are to be sent given a specific spending ceiling. If the spending goes above this ceiling, an email is to be sent. This is an optional feature for the user. |
| Priority value | 3 |
| Ideal Person-Hours | The use should be able to learn this feature in 2 minutes. |

|  |  |
| --- | --- |
| 1. Stock Simulator | |
| User Story | As a user, I want to use a stock market simulator so that I can see how I could have done in the market had I put money in the stock market. |
| Description | This only simulates the stock market and does not produce real results. It allows the user to see how the user could have done if he/she had invested in the stock market. This is an optional feature for the user. |
| Priority value | 4 |
| Ideal Person-Hours | The user should be able to learn this feature in 1 hour. |

40/40

|  |  |
| --- | --- |
| 1. Debt Payoff | |
| User Story | As a user, I should be able to see how long it will take me to pay off a debt or multiple debts given how much I put towards the debts |
| Description | The user can enter how much to put towards the debt, how much interest is on the debt and will return when the debt is paid off, this included multiple debts |
| Priority value | 2 |
| Ideal Person-Hours | The user should be able to learn the system in 30 minutes. |

**Part II: System Requirements in the Form of Tasks**

1. Real Time Expense/Income Entry
2. UI includes expense and income entry boxes, default to zero
3. User is notified immediately that the entry was successful or not
4. Budgets and sub-budgets are updated to the user
5. Savings Calculator
6. User enters amount needing to save and how much they can put back
7. Results includes the amount of time to reach that goal
8. User can put this goal in their overall budget as a monthly expense
9. Budget Plan
   1. Give the user detailed information of their daily, weekly, monthly and yearly budgets
   2. The system tracks the users current saving after every entry
10. Income Tax
    1. User is able to select documents needed for their taxes
    2. All documents are stored in the database
    3. User can also store the necessary information for each documents
    4. The tax emulator uses the information entered for each documents
    5. User receives estimated returned
11. Secure Storage System
    1. Logon system for each user
12. Remote Real Time updates
    1. Web portal so user can logon to the system at any location
    2. Desktop application for the user
       1. If system is down, user can continue working on their budget and the budget will then be updated as soon as the system is running
    3. Mobile application for convenience
13. Email Alerts for Spending Budget
    1. User sets a subset bar of their entire spending account
    2. When subset bar is reached, the user is then notified
14. Stock Simulator
    1. User can enter an “investment amount”
    2. User can track how their “invested amount” is doing in the simulator
    3. No real life results are produced
15. Debt Payoff
    1. User can use a snowball or higher interest method calculator to evaluate their debt payoff
    2. User can use this information and use it as an expense in their budget38/40
    3. Budget and sub-budget will be updated

**Part III: Non-Functional Requirements**

1. Product Requirements
2. Efficiency Requirements
   1. Performance Requirements
      1. For small simple request, the user should get the information almost immediately, no more than 1 second
      2. For bigger request, such as 12 months’ worth of information., the user should expect the results within 15 seconds
   2. Space Requirements
      1. For small request, the information expected to and from the database should be no more than 50KB
      2. For much larger request, the information sent to the user should take about 500KB
3. Dependability Requirements
   1. All information requested by the user should produce correct results based on the information the user has sent to the database
   2. Specific algorithm will be used to produce the correct results and in timely manner, 1-15 seconds
4. Security Requirements
   1. All users will require a logon security system to identify the user
   2. Any request to and from the database will be encrypted
5. Usability Requirements
   1. All request will be possible with the mouse and keyboard
   2. Any delays longer than 1 second will produce a “please wait” message
   3. Desktop Application
      1. All option will be accessible within toolbars and sub options
      2. The desktop application will be usable while offline
   4. User will be able to use a desktop application and a web portal or a mobile application (in later iterations).
6. Organizational Requirements
7. Environmental Requirements
   1. The user must have access to a computer with Java installed in order to use the desktop app
8. Operational Requirements
   1. Users of the system must create a unique login and password
9. Development Requirements
   1. The desktop application must be built first with the web portal/mobile app to model after
10. External Requirements
11. Regulatory Requirements - None
12. Ethical Requirements
    1. The system should safeguard the users’ information—the user will expect a certain level of trust that their data is not being sold to third-parties.20/20
13. Legislative Requirements
    1. Accounting Requirements
    2. Safety/Security Requirements
       1. The system should only be accessible for a particular user and should prevent any others from accessing another user’s information