Requirements Document Revision 0
Expense Tracker

Volere Template

Radhika Sharma
Alexander Jackson
Zachary Bazen

McMaster University

Revision 1 - December 8th, 2015

Contents

List of Tables

List of Figures

Revision History

Revision #	Revision Date	Description of Change	Author(s)
Revision 0	October 9, 2015	Functional Requirments Specification	Zachary Bazen
Revision 1	October 10, 2015	Non Functional Requirements Specificatoins	Alexander Bazen
Revision 2	October 12, 2015	Introduction and Conclusion	Radhika Sharma
Revision 3	December 3, 2015	Edits	Radhika Sharma

Table 1: **Revision History**

Project Drivers

1 The Purpose of the Product

The purpose of this project is to provide small businesses with a program that will help keep track of expenses when multiple projects are being executed at once. Traditional paper bookkeeping methods can be tedious and can contain lots of errors due to errors in human input. In addition to these issues, searching through bookkeeping records is time consuming. Often times on-line banking systems and computerized bookkeeping programs lack tools for handling these issues in an optimized manner. On-line bank sites have limited search and sub-search capabilities and lack the the functionality of modifying transitions.Computerized bookkeeping programs often bundle these tasks into a program that is difficult to use. Learning how to use these programs is often too difficult for users that have limited computer knowledge. This program aims to provide users with these functionalities in a simple and intuitive program.

2 The Stakeholders

Stakeholders

Stakeholder	**Rationale**
Small business owners	These people will be the target client and user. End users provide insight into problems with current off the shelf solutions and what features should and should not be incorporated into the expense tracker. They also provide valuable information as to how current programs functions and the challenges that arise from using these programs.
Small Business Owner	Stakeholder; the main user of the program, owns and operates a business on a smaller scale relative to other businesses.
Software Developer	Software Developers are those who will be designing and building the program. Their role in the development process is key to the success of the program. For the program to effectively eradicate the issues that the clients face, Software Developers must provide basic functionality in the program and implement optimized search algorithms. All of this must be accomplished while still listening to the needs and wants of the client. It is also essential that the Software Developers provide a user interface that the clients find easy to use.
Manager	Managers will oversee the entire process of creating the expense tracker. They will provide valuable input on what the expense tracker must do to ensure functionality for the end user.

Marketing Team	The marketing team will be able to provide input into how the program should be marketed such that users find it interesting. The marketing team will also be able to define how the program must look so that it appeals to end users. In addition the marketing team may be able to provide input on what will ensure the ease of use for the user.

Table 2: **Stakeholders**

Product Constraints

3 Mandated Constraints

A mandated constraint for this project states that this project must be completed by December 8, 2015. During the process of implementation, the program must be fully documented. The program must be able to compile and run for the Managers and Users the technology. The final product must be a self contained program that can run on Windows and Apple machines. The input for this project must be a comma separated value file code in ASCII or UTF-8 encoding.

Users must be consistent with respect to currency when using this program. The expense tracker will not have conversion tools, therefore for correct output, the user must stick to one currency when entering expenses.

4 Name Convention and Terminology

The following are definitions of the terminology that are used in this project. Further definitions and terminology will be added as the project continues to bedeveloped.

Definitions

Term	**Definition**
Stakeholder	Individual or group of individuals that have a stake or interest in the development of the project.
Small Business Owner	Stakeholder; the main user of the program, owns and operates a business on a smaller scale relative to other businesses.
Software Developer	Stakeholder, a person that designs, documents and maintains a piece of software code.
Manager	Stakeholder; will provide input into the design and of the program, will notify Software Developers of what the business owners will need and want in the program.

Expense Tracker	a piece of software that allows a user to track a history of expenses for single or multiple projects. It also allows the user to search and modify the history of transactions.
Constraint	A boundary that the project must be developed within.
Operating System	A low level computer program that manages the available resources on a computer.
Windows	An operating system that is developed by Microsoft
Apple	Technology company that developed the Macintosh operating system.

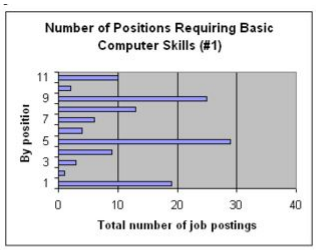
Table 3: **Definitions**

5 Relevant Facts and Assumptions

Assumption

The main assumption for this project is that the end users have a basic knowledge of how to use command-line. It is assumed they will be able to interact and navigate the program using commands dictated by the command-line and the software developers.

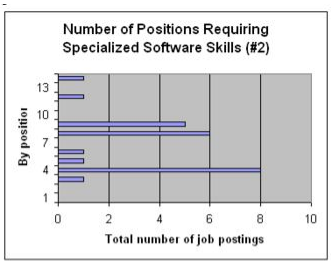
Figure 1: Jobs Requiring Basic Computer Skills



Number of Position Requiring Basic Computer Skills (#1)

AAM Position Category	Internet/Web Skills (#3)
Directors/Administrators	1
Accounting/Finance	0
IT/Web	7
Exhibitions	3
Programming/Education	12
PR/Marketing	1
Development/Membership	3
Collections Mgt/Registrar	0
Curatorial	1
Conservation	0
Administrative/Clerical	2
Visitor/Customer Services	3
Human Resources	0
Consultants	0

Figure 2: Jobs Requiring Specialized Computer Skills



Number of Positions Requiring Specialized Software Skills (#2)

AAM Position Category	Specialized IT Skills (#2)
Directors/Administrators	0
Accounting/Finance	0
IT/Web	1
Exhibitions	8
Programming/Education	1
PR/Marketing	1
Development/Membership	0
Collections Mgt/Registrar	6
Curatorial	5
Conservation	0
Administrative/Clerical	0
Visitor/Customer Services	1
Human Resources	0
Consultants	1

As shown in the above figures, more jobs require basic computer skills but do not require specialized computer skills. It is therefore safe to assume that any end users will have basic knowledge of how to use a computer, however their knowledge of computers may be limited. From this limitation rises the necessity of ease of use.

Facts

The project being developed already exists as an open source software project onGitHub and can be found by using the search function and searching expense tracker.

Functional Requirements

6 The scope of the Work

The expense tracker will work by allowing the user to input transactions into a command line using a predetermined set of commands. The command line will then store these transactions in a CSV file for later use. The expense tracker will allow the user to search the transaction by either project type or date via commands inputed into the command line.

7 Business Data Model and Data Dictionary

Currently the business model is to have the final product available to users by December 8th 2015 and marketed such that users can use the program for tax purposes before the tax season in April.

8 The scope of the product

The product will run as a executable jar file and will be compatible with the major operating systems (Windows and OSxYosemite).

9 Inputs and Outputs to the System

The inputs to the system will be one of three items: a file of existing entries, user input, and user actions. The program will require an initial file of existing entries in a specific format. The number of entries in the initial file is irrelevant. Once the file is loaded into the system, the program will then wait for user actions (mouse clicks on drop down menus) and user input (new entries or modifications to existing entries) The following requirements will outline how the user inputs will be handled by the system. The outputs of the system will be a file that contains the entries that were present at the start of the session, the entries that were added during the session, the entries that were modified during the session, and the file will not have the entries that were deleted during the session.

10 Functional Requirements

Requirement # 1
Description: The program must read a file from the file path specified by the user
Rationale: The information read from the input will be used asin the program
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Requirement # 2
Description: The program must write to a text/comma separated value file
Rationale: The user needs to be able to retrieve the information that was modified at a future date
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Requirement # 3
Description: The program must save the text/comma separated value file to the location specified by the user
Rationale: The user wants to be able to know where to retrievetheir information at a later date
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Requirement # 4
Description: The program must handle file not found exceptions

Rationale: The file may not exist or the user gave the wrong file path
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Requirement # 5
Description: The program must allows the user to view all transactions in the file
Rationale: The user may want to view all transactions recorded in the file
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Requirement # 6
Description: The program must allow the user to search the document for a transaction
Rationale: E.g. Users may want to look for transactions on a given date
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Requirement # 7
Description: The program must allow the user to search on a second field
Rationale: E.g. Users may want to look for a specific transaction on certain days
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Requirement # 8
Description: The program must allow the user to modify existing transactions
Rationale: The users may want to modify comments or correct errors
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Requirement # 9
Description: The program must support multiple text file encodings
Rationale: E.g. ASCII or UTF-8, users may be operating in a different country or language that uses different characters
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Requirement # 10
Description: The program must be able to parse multiple category comma separated value files
Rationale: The transactions will be stored under different headings

Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Requirement # 11
Description: The program must display to the user the full file path where there previous file was stored
Rationale: The user may not remember the path to the file
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

~~**Requirement # 12**~~
~~**Description**: The input from the user must be taken from the console~~
~~**Rationale**: Console is a supplement for the GUI~~
~~**Originator**: Zachary Bazen~~
~~**Priority**: High~~
~~**History**: Created October 5, 2015~~

Requirement # 12
Description: The input from the user must be accepted
Rationale: The GUI must relay the user input to the back - end program
Originator: Radhika Sharma
Priority: High
History: Created December 3, 2015

~~**Requirement # 13**~~
~~**Description**: The program must allow the user to re-enter incorrect input~~
~~**Rationale**: User input may contain mistakes~~
~~**Originator**: Zachary Bazen~~
~~**Priority**: High~~
~~**History**: Created October 5, 2015~~

Requirement # 13
Description: The program must allow the user to modify incorrect input
Rationale: User input may contain mistakes
Originator: Radhika Sharma
Priority: High
History: Created December 3, 2015

Requirement # 14
Description: The program must tell the user when a search item is not found
Rationale: User input the data incorrectly, the item is not found and the user needs to be told
Originator: Zachary Bazen

Priority: High
History: Created October 5, 2015

Requirement # 15
Description: The program must tell the user when a sub search item is not found
Rationale: User inputs the data incorrectly, the item is not found and the user needs to be told
Originator: Zachary Bazen
Priority: High
History: Created October 5, 2015

Non - Functional Requirements

11 Look and Feel Requirements

11 Appearance Requirements

The interface of the program must be simple and intuitive for all users who have a basic knowledge of computers.

11 Style Requirements

The program must remain a simple, command-based application and keep graphics to a minimum.

12 Usability requirements

12 Ease of Use Requirements

The interface of the program must be easy to use and understand. the commands ofthe program must be simple and self-explanatory, even to new users. To verifythat the commands are easy to use, four Software Engineering / Comupter Science students and four students from other faculties will test the interface. Both groups of students must be able to use the program and its commands without any outside help.

12 Personalization and Internationalization Requirements

The interface of the program will be written in English. The currency will remain unitless so that the program can be used with any currency system, and the metric system will be used for all other units of measurement.

12 Learning Requirements

The commands of the program must be easy for users to learn. A help command must be always available to clearly and concisely explain what commands are available and what each command does. It must take no longer than 30 seconds for a user to figure out what command to use next.

12 Understandability and Politeness Requirements

The program must have a high degree of understandability and simplicity. A first-time user must be able to easily familiarize them-self with how to use the program and its commands using the program's help function.

12 Accessibility Requirements

A user of the program needs to be able to read text off a computer screen and be able to type the program's commands.

12 Install-ability Requirements

The program will be an executable jar file and therefore will not need to be installed. The program will simply need to be downloaded on to the users computer, however it will require that the users computer will have Java on it. It is a requirement that any user with basic computer skills will be able to download the program and start it, provided that the users computer has Java.

13 Performance Requirements

13 Speed and Latency Requirements

As the program deals mainly with text and text files, speed is expected to be high and mainly dependent on the user's computer. The program must respond to the user's input within 100 milliseconds.

13 Safety-Critical Requirements

The program should not view or otherwise interact with any files on the user's computer other than the text or CSV files associated with the program.

The program shall not bring harm to any individual.

13 Precision or Accuracy Requirements

All monetary value calculations must be accurate within two decimal points. It is critical that the user's finances be accurate, and the smallest currency denomination is the largest acceptable amount of error.

13 Reliability or Availability Requirements

The program should be usable 24 hours per day, 365 days per year (366 on leap years).

13 Robustness or Fault-Tolerance Requirements

The program should handle incorrect user input alert the user with an error message. The program should also allow the user to reenter the input.

13 Capacity Requirements

The program stores inputted expenses as text, so the only limit to how many expenses the program can track at once would be the maximum possible size of the text file.

13 Scalability or Extensibility Requirements

The program manages expenses as text, so scalability, in terms of increasing number of total expenses over time, is not a concern.

13 Longevity Requirements

Does not apply because their is no metric to measure the longevity requirement.

14 Operational and Environment Requirements

14 Expected Physical Environment

The program can be used for business expenses as well as personal/home expenses.

14 Requirements for Interacting with Adjacent Systems

The program is a self contained, this requirement does not apply.

14 Production Requirements

The program can be downloaded as an executable from GitLab and can be used on any computer that can run Java.

14 Release Requirements

A new version of the program must allow previous versions to still function properly.

15 Maintainability and Portability Requirements

15 Maintenance requirements

Maintenance of the program would be done solely by the developers.

15 Supportability Requirements

The program is self-contained and self-supporting.

15 Adaptability Requirements

The program is a self-contained Java program, thus it is platform-independent.

16 Security Requirements

16 Access Requirements

The program has access only to the expense information inputted by the user. The program has access to no other information about the user, and the user ultimately decides what expenses the program should track. The tracked expenses are saved locally and the developer has no access to this information.

16 Integrity Requirements

The program must be able to identify improper input data and alert the user and allow them to retry.

16 Privacy Requirements

The program only has access to the information the user chooses to input, all inputted info is stored locally, and the program can only access the files that the user commands it to.

16 Audit Requirements

N/A

16 Immunity Requirements

N/A

17 Cultural and Political Requirements

17 Cultural Requirements

The program can be used by anyone regardless of cultural background.

18 Legal Requirements

18 Compliance Requirements

N/A

18 Standards Requirements

The program must meet the standards of the Software Engineering 3XA3 project.

Project Issues

19 Open issues

Optimal Search Algorithms: It is yet to be determined what search algorithms should be used such that the system will be optimized and perform a search in a reasonable amount

of time.
Incorrect User Input : It is yet to be determined how the system will handle searches where the user input is incorrect.

20 Off-the-shelf solutions

Microsoft Excel: This existing microsoft program may aid in the storing of information. If the information is stored in the excel file, then it may be easier for the program to search queries.

21 New Problems

Put your text here.

22 Tasks

Continue programming the project
Begin the construction of the design document
Demonstrate proof of concepts
Revise design document
Create a user reference guide
Identify test cases and test them against the system
Revision of the user reference guide
Revision of Test Cases
Demonstrate final project
Revise final document

23 Migration to the New Product

The migration to the New Product will be done in a systematic manner. First it will be the focus of the software developers to implement the functional requirements. After extraneous testing, it will be checked whether or not the non-functional requirements have been met. If they have not been met, then the program will be revised until it does meet the non functional requirements.

24 Risks

Improper security provisions can result in a breach of information. This will be catastrophic for private companies using this software and wish to keep their information private. In addition, if the program does not complete tasks correctly, this can result in error in the data which can then lead to tremendous errors for the end user. In the proof of concept demo, it will be shown that the application does in fact complete tasks such as add, modify, search and delete correctly.

25 Costs

There are no direct monetary costs associated with the development of this project however various time and space commitments will be required of the developers and of the systems respectively.

26 User Documentation and Training

It is assumed that all users are trained in the basic use of computers, however a list of commands and corresponding descriptions will be made available to the user via text document, for reference purposes.

27 Waiting room

Future releases will be expected to have additional computational properties such as calculating the total expenses for a given time period. Although not in the current plan, eventually a version of this program will ideally be developed for individual people who wish to track their personal expenses.

28 Ideas for Solutions

To mitigate risks, perhaps the program should be password protected. This will ensure partial security.