
Software Requirements Specification

for

Finance Tracker

Version 1.1

Prepared by Fatima Nadeem and Chinonyerem Ukaegbu

NYU Abu Dhabi

10th May, 2022

Table of Contents

Revision History	5
Introduction	1
Purpose	1
Document Conventions	1
Intended Audience and Reading Suggestions	1
Product Scope	2
References	2
Description	3
Product Perspective	3
Product Functions	3
User Classes and Characteristics	4
Operating Environment	4
Design and Implementation Constraints	4
User Documentation	4
Assumptions and Dependencies	5
External Interface Requirements	6
User Interfaces	6

	3
Hardware Interfaces	6
Software Interfaces	6
Communications Interfaces	6
System Features	7
Registration	7
Description and Priority	7
Stimulus/Response Sequences	7
Functional Requirements	7
Login	7
Description and Priority	7
Stimulus/Response Sequences	7
Functional Requirements	8
Create Journal	8
Description and Priority	8
Stimulus/Response Sequences	8
Functional Requirements	8
Create Transaction	8
Description and Priority	8
Stimulus/Response Sequences	8
Functional Requirements	9

	4
Edit Transaction	9
Description and Priority	9
Stimulus/Response Sequences	9
Functional Requirements	9
Delete Transaction	10
Description and Priority	10
Stimulus/Response Sequences	10
Functional Requirements	10
Logout	10
Description and Priority	10
Stimulus/Response Sequences	10
Functional Requirements	11
Delete Account	11
Description and Priority	11
Stimulus/Response Sequences	11
Functional Requirements	11
Other Nonfunctional Requirements	12
Performance Requirements	12
Safety Requirements	12
Security Requirements	12

	5
Software Quality Attributes	12
Business Rules	12
Logical Data Management	13
Memory Constraints	13
Notes about Implementation	13
Appendix A: Glossary	13
Appendix B: Analysis Models	13
Appendix C: To Be Determined List	14

Revision History

Name	Date	Reason For Changes	Version
Fatima Nadeem & Chinonyerem Ukaegbu	10 May 2022	Adding new info to section 8 - Notes about Implementation	1.1

1. Introduction

1.1 Purpose

The Finance tracker web application described in this SRS document will be the very first version released to the New York University Abu Dhabi students. Currently, being a part of a course project, the implementation scale of this product will be limited and would therefore only allow access to students. This SRS is intended to extensively describe all the functional requirements and all the challenges and expectations related to them.

1.2 Document Conventions

SRS - System Requirements Specification

NYUAD - New York University Abu Dhabi

app - Web Application of Finance Tracker

FAQs - Frequently Asked Questions

HTML- Hypertext Markup Language

PDF - Portable Document Format

HTTP(S) - (Secure) Hypertext Transfer Protocol

TCP - Transmission Control Protocol

IP - Internet Protocol

activities - money-spending activities/ transactions

1.3 Intended Audience and Reading Suggestions

This SRS is comprehensive and contains information for all audiences in one location. Everyone in the audience will benefit from the initial introduction. The product's scope and specifications can be examined in depth through this SRS. This documentation will also assist us in finalizing the product's specifications. This document may be used by student users to learn more about the app's features and interface, as well as how to utilize it.

This document will be used by us and testing users to aid in the development process. This document will also be used to create a complete user manual. This SRS will be beneficial to consolidate all of the requirements and action plans in one place. To really get the most out of this document, read this in the order that it has been prepared. You will first be able to learn about the product's fundamental scope and goals, following that you will be able to read in-depth about the features and interface offered.

1.4 Product Scope

College is an amazing experience - definitely an expensive one. Currently in NYUAD, where more than 85% of students are international, living and studying in a foreign country can be very expensive. Budgeting is a skill that will be never dated and it will not only be helpful for college years but also life after university. Budgeting and tracking one's finances will allow the user to distinguish between their needs and wants. Identifying expenses is always helpful in planning ahead and can help reduce stress.

Our app will allow users to seamlessly create a record of all their expenses, view their spending trends and habits. This sense of organization all at one place, provided by our app, will help users understand and be consistent with their finances. Exposure to the skill of budgeting during university education will serve the students far in the future. The easy user interface of our app will make the process of budgeting simpler, smoother, and fun for our users. Refer to the [System Request](#) documentation to learn more about the scope of the product - the expected values and motivation.

1.5 References

Software Engineering A Practitioner's Approach, 10th edition, Roger S Pressmen
System Request, Fatima Nadeem; Chinonyerem Ukaegbu,
<https://github.com/ChinoUkaegbu/SE_ExpenseTracker/blob/main/System%20Request.pdf
>

2. Description

2.1 Product Perspective

The finance tracker and budgeting web tool as described in this SRS document is a new, self-contained product. Previously, most of the students interested in tracking their finances or keen on budgeting would rely on recording their transactions and activities through paper-based systems - journals, dairies, and notebooks. Some would use many mobile applications available for this purpose. However, the free apps would only allow limited access to services and functionalities or be only on a short trial period. To have unlimited access to similar convenience and functions offered in our app, they would be required to pay outrageous monthly/ yearly subscription fees to the current paid versions of finance trackers like our intended web app. The proposed product in this SRS will allow students of NYUAD to easily and seamlessly track their finances and budget their university life without having to through worrisome existing options available to them.

2.2 Product Functions

1. Student Users

- a. The student users must be able to register/create a new account and login to an already existing account
- b. The student users must be able to make accounts and have their information stored
- c. The student users will be able to add, modify or delete the information provided by them when they register.
- d. The student users must be able to input their categorized periodic transactions
- e. The student users must be able to view their categorized expenses for defined period of time
- f. The student users must be able to set a categorized budget/ expected expenses
- g. The student users must be able to view their spending trends over a given time period
- h. The student users must be able to compare their spending trends to other users of the system
- i. The student users must be able to retrieve previously-stored data
- j. The student users must be able to access tutorials/ FAQs on navigating the interfaces
- k. The student users must be able to logout of the system or delete their account.

2. Administrators

- a. The administrators must have access to all student user functions while simultaneously managing the system

2.3 User Classes and Characteristics

Users of this software include product administrators (authors of this documentation) and NYUAD students with the need for a tracker to maintain their finances who we will refer to in this document as student users. Student users will have access to customers' functions and the administrators will have access to both customer functions and management functions.

2.4 Operating Environment

The client-side component of the application must operate within common web browser environments. The application runs in the latest version of Google Chrome or Mozilla Firefox or Internet Explorer browser. The server-side operates on Windows, Linux, or Mac. Basic knowledge of using computers is adequate to use this application. Knowledge of how to use a mouse or keyboard and internet browser is necessary. The user interface will be friendly enough to guide the user.

2.5 Design and Implementation Constraints

- The information of all users must be stored in a database that is accessible to the website
- The fitness tracker system is running 24 hours a day.
- Users must have correct usernames and passwords in order to access and perform functions on the site.
- Access to an internet connection is required.

2.6 User Documentation

The student users are the target audience for all user documentation. Guidelines, tutorials, and Frequently Asked Questions (FAQs) will be provided in the user manual (in PDF or HTML format) that will be compiled with during the app implementation stage.

2.7 Assumptions and Dependencies

In addition to the other assumptions and dependencies mentioned earlier, the users should be able to navigate Github as the website will be hosted through its platform. Also, the system assumes all transactions and listings are in United Arab Emirates Dirhams (AED). However, the system will direct users to a currency converter that can be used before inputting the data.

3. External Interface Requirements

3.1 User Interfaces

The users will be able to access the product through a web application - accessible from Desktop web browsers. The website will be organized in different sections, pertaining to the different user interfaces available. Upon loading the site for the first time, the user will be redirected to the Login UI. After being verified, they would be redirected to the Dashboard UI. From there, they can access the Transaction UI and Trends UI. All the interfaces will be based on forms to collect information from the user and process it through the hosted python program. A description of all interfaces can be found below:

Login UI - Enter username and password to allow authentication

Dashboard UI - allow navigation on the web app to various functions available

Transaction UI - add new transactions/ activities and modify existing transactions activities

Trends UI - users view their spending habits/ history and trends and compare to other users

3.2 Hardware Interfaces

All server-side components must operate on server-class computers. All client-side components must operate on personal computers.

3.3 Software Interfaces

The back-end python program will control all functions of the finance tracking and budgeting tool. Through the available software, the python program will be able to communicate with the web interface developed. The website will then be hosted on GitHub and it will use GitHub pages to view it on web browsers on personal devices. At this point, the web application of our product will only be available for access using Windows or Mac Desktop versions - laptops and PCs. Currently, due to software development constraints, it will be not accessible on mobile/ tablet devices.

3.4 Communications Interfaces

- Client on Internet will be using HTTP/HTTPS protocol
- Client on Intranet will be using TCP/IP protocol

4. System Features

4.1 Registration

4.1.1 Description and Priority

Provides a mode of registering new users. Priority:High

4.1.2 Stimulus/Response Sequences

Stimulus: Individual clicks on registration button

Response: The individual is directed to the registration interface where they are prompted to enter in a username and password

Stimulus: Individual enters username and password

Response: If the username is already in use, the individual is alerted and asked to enter in a new username. Else, an account is created and the individual becomes a user.

4.1.3 Functional Requirements

REQ-1: Users can only create accounts with unique usernames

REQ-2: The system will store the credentials of the new user in the database

4.2 Login

4.2.1 Description and Priority

Users will be asked to enter their valid username to be authenticated into the system.
Priority: High

4.2.2 Stimulus/Response Sequences

Stimulus: Individual clicks on login button

Response: The individual is directed to the login interface where they are prompted to enter in their username and password

Stimulus: Individual enters an invalid username or password

Response: The system notifies the user that either the username or password is incorrect and prompts the user to try again. After 5 tries, the individual is restricted from attempting to log in.

Stimulus: Individual enters a valid username and password

Response: The system displays the dashboard interface upon successful authentication

4.2.3 Functional Requirements

REQ-3: Users shall only be able to login with valid credentials

REQ-4: The system shall authenticate user credentials

4.3 Create Journal

4.3.1 Description and Priority

Users will be able to create multiple journals in order to track transactions over a stated time period

4.3.2 Stimulus/Response Sequences

Stimulus: User clicks on 'Create Journal' button

Response: The individual is prompted to enter a name (mandatory), start date (mandatory) and further description of the journal (optional)

Stimulus: Individual enters the name of already existing journal

Response: The system notifies the user that the journal name is already in use and prompts them to enter a new name

Stimulus: Individual enters valid details

Response: The system displays the empty journal. The user is then able to input transactions and edit them

4.3.3 Functional Requirements

REQ-5: Users shall only be able to create journals with unique names

4.4 Create Transaction

4.4.1 Description and Priority

Users will be able to create transactions within a journal

4.4.2 Stimulus/Response Sequences

Stimulus: User clicks on 'Create Transaction' button

Response: The individual is prompted to select an already existing journal to save the transaction in or to create a new journal to save the transaction in

Stimulus: User creates new journal

Response: The system executes the 'Create Journal' process

Stimulus: User selects journal to save transaction in

Response: The system prompts the user to enter transaction details and to save the transaction

Stimulus: User clicks on 'Save' button

Response: The transaction is saved in the journal

4.4.3 Functional Requirements

REQ-6: The system shall allow a user to provide the type of transaction

REQ-7: The system shall allow the user to provide details of the transaction

REQ-8: The system shall allow the user to categorize the transaction as either expenditure or revenue

4.5 Edit Transaction

4.5.1 Description and Priority

Users will be able to edit transactions within a journal

4.5.2 Stimulus/Response Sequences

Stimulus: User clicks on transaction within a journal

Response: A button appears prompting the user to edit the transaction

Stimulus: User clicks on 'Edit' button

Response: The system prompts the user to select which detail they would like to edit.

Stimulus: User selects journal to save transaction in

Response: The system prompts the user to enter transaction details and to save the transaction

Stimulus: User clicks on 'Save' button

Response: The transaction is saved in the journal

4.5.3 Functional Requirements

REQ-9: The system shall allow a user to select particular detail to edit

4.6 Delete Transaction

4.6.1 Description and Priority

Users will be able to delete transactions within a journal

4.6.2 Stimulus/Response Sequences

Stimulus: User clicks on transaction within a journal

Response: A button appears prompting the user to delete the transaction

Stimulus: User clicks on 'Delete' button

Response: The system displays a message prompting the user to confirm if they would like to permanently delete the transaction.

Stimulus: User selects 'Cancel'

Response: The system returns to the journal

Stimulus: User selects 'Delete'

Response: The transaction is deleted from the journal

4.6.3 Functional Requirements

REQ-10: The system shall allow a user to delete a transaction in a journal

4.7 Logout

4.7.1 Description and Priority

This feature is to enable users who have logged into the system to log out of it

4.7.2 Stimulus/Response Sequences

Stimulus: User clicks on log out button

Response: The system displays a message prompting the user to confirm if they would like to log out.

Stimulus: User selects 'Cancel'

Response: The system returns to the main interface

Stimulus: User selects 'Logout'

Response: The user is logged out of the system.

4.7.3 Functional Requirements

REQ-11: The system shall provide a mechanism for logged in users to log out

4.8 Delete Account

4.8.1 Description and Priority

This feature is to enable users to permanently delete their accounts

4.8.2 Stimulus/Response Sequences

Stimulus: User clicks on 'Delete Account' button

Response: The system displays a message prompting the user to confirm if they would like to permanently delete their account.

Stimulus: User selects 'Cancel'

Response: The system returns to the main interface

Stimulus: User selects 'Delete'

Response: The user's account is deleted from the system.

4.8.3 Functional Requirements

REQ-11: The system shall provide a mechanism for users to permanently delete their account

REQ-12: The details of the deleted account shall be removed from the database.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The app must perform to the highest quality and performance criteria. The web app's server must allow a seamless experience for all users. This must be achieved by the server by handling all the users at all instances. The performance of the app website shall not drop by 15% even when they are running at the maximum capacity of the users.

5.2 Safety Requirements

Every user of the app will be granted access to the app through a unique system id and password that will be verified at every new access session. This will only permit verified access to the data.

5.3 Security Requirements

HTTPS connectivity to the web app allows for secure access to all data stored on the server. The app's site will only be accessible to browsers that have the most up-to-date security standards and certifications. Furthermore, the processes outlined in [5.2](#) for meeting safety standards also meet contemporary security needs.

5.4 Software Quality Attributes

In terms of dependability, availability, responsiveness, and scalability, the app is designed to meet high-quality criteria. The software must be available 24 hours a day, seven days a week, due to its functional nature and breadth. Users must be able to connect and use any function at any time that is convenient for them. All users should be notified about scheduled maintenance at least two weeks ahead of time. Data and server updates in real-time must be well-organized and coded. These should have no impact on the user's experience. The web app must be responsive in order to provide seamless and quick access to all resources and functions. During the first few months, the supported user limit capacity must be double the number of possible app users.

5.5 Business Rules

Each student user should have a valid net ID in order to be able to access the system.

6. Logical Data Management

The information for each user will be stored on the webserver - GitHub. The transaction history, journals, and the spending trends of every user will be stored separately. A CSV file will be created to store all this information - a file for each user. All these files will be used as the database alongside other files including information regarding user authentication and general trends and public information.

7. Memory Constraints

As the flow of logical data management is discussed in the [previous](#) section, we have limitations on our memory. Although the backend program will save the information in separate files, there are still chances of data mismanagement. GitHub only allows a maximum of 2GB single file size. CSV files of every user might be very large in size towards a year or so in heavy usage of the system. The public information files might reach the maximum limit much sooner. Therefore, updated versions with these constraints resolved must be released within six months of initial deployment at the start of May 2022.

8. Notes about Implementation

Considering the constraints while implementation, the project has been revised and presented as a terminal application in the live demonstration and presentation. However, the code is very scalable and practical and will be further developed with the assistance of out-sourced team and develop a web application version of the system. The current code can be easily used as the backend script while the web application is being used.

Appendix A: Glossary

This section will be updated if need be during the implementation process.

Appendix B: Analysis Models

This section will be updated if need be during the implementation process.

Appendix C: To Be Determined List

This section will be updated if need be during the implementation process.