College of Computing and Software Engineering

Sweetledger Software Requirements

Specifications

Dr. Ermias Mamo

Devin Perry, Matthew Crowley, Kendal Elison, Kahmin Keller, Connor Oberlin

September 25, 2025

Table of Contents

Introduction

Purpose

The purpose of this software requirements specifications document is to let the shareholders of the project understand what the requirements are. The shareholders (the users of the system, Dr. Ermias Mamo, and the programmers) will refer to this document for future and present clarification of what this system entails. This document will be updated and presented during the presentations of all five sprints during the allotted time period of this semester. The requirements and scope of this system will be housed in this document. The requirements of Sweetledger have been approved and must not be deviated from to ensure proper completion of the system.

Scope

Sweetledger is a web-based accounting ledger book system that will simulate a high-quality accounting system that a company would use. Sweetledger consists of five sprints, with each one building on the previous sprint. Each sprint will be evaluated by the client, Dr. Ermias Mamo, to ensure proper completion of Sweetledger.

Sweetledger will use AWS Elastic Beanstalk as a website host, accompanied by MongoDB as the database. The primary languages utilized will be JavaScript, CSS, HTML, and C#.

Overall Description

This section of the software requirements specifications document pertains to helping the reader understand why the requirements make sense for Sweetledger. Each sprint consists of a different section of Sweetledger.

Sprint 1 is the User Interface Module. Sprint 2 is the Chart of Accounts Module. Sprint 3 is the Journalizing and Ledger Module. Sprint 4 is the Adjusting Entries and Financial Reports Module. Finally, Sprint 5 is the All Users Module.

Sprint 1 is the design and deployment of Sweetledger's user interface and log-in module. It is important to gather requirements and follow them strictly because the user interface must be effective for Sweetledger to succeed.

Performance Requirements and Reliability

It is of the utmost importance that Sweetledger is efficient in its functions. Sweetledger will be efficient in all domains because of the technologies utilized. MongoDB and AWS Elastic Beanstalk are extremely responsive and will not limit the user of Sweetledger whatsoever once the service is online. With AWS Elastic Beanstalk, there is a brief waiting period when the system is connecting to AWS after an extended down period. With that being said, AWS is extremely responsive and efficient when it is online.

We have chosen AWS to host Sweetledger online, which allows for all users to use Sweetledger. There will also be a mobile version of Sweetledger that will release near the final deployment of the system. With proper programming techniques, Sweetledger's back end calculations will be optimized.

Sweetledger will also be reliable. MongoDB is partnered with AWS, and they are tightly connected. As long as AWS is available, Sweetledger will also be available. We have chosen MongoDB as our database to induce a "less moving parts" mindset to Sweetledger, where less can go wrong in terms of system reliability.

With all of these techniques and technologies as the backbone of our system, Sweetledger will perform smoothly, and the user will have no problem exploring the functions efficiently and reliably.

Sprint 1 - User Interface and Log-In Module

1. Allow three types of users – administrator, manager, and regular user (accountant) to login to the system;

2. The administrator user should be able to create users and assign roles;

3. The administrator user should be able to update information about a system user;

4. The administrator user should be able to activate or deactivate each kind of user;

5. Each kind of user should be able to log in to the system once credentials are created in the system

6. The login username, picture, should be displayed clearly on the top right corner of the login page once they have successfully logged into the system;

7. The login page should have:

a. A text box to enter the username
b. A textbox to enter a password which will be hidden as the user keys in the password
c. A submit button
d. A forgot password button
e. A create new user button
f. A logo which will be displayed on all the pages of the application

8. The create a new user button will be used if the user is accessing the system for the first time. Clicking this button should display a user interface where the user will provide personal information such as first name, last name, address, DOB, and click submit to request access to the application. The administrator should receive an email request and must approve or reject the request. If approved, an email should be sent to the user with a link to login to the system;

9. A button for ‘forgot password’. If this button is clicked, the system should prompt the user to enter email address and user id the person provided when his credentials were created in the system and ask security questions to allow him to supply new password;

10. Passwords must be a minimum of 8 characters, must start with a letter, must have a letter, a number and a special character, if this requirement is not satisfied, display at appropriate error message;

11. Password used in the past cannot be used when password is reset;

12. Password must be encrypted;

13. A maximum of three wrong password attempts should be allowed after which the user should be suspended;

14. All login information must be stored in database tables;

15. Three days before a password expires, the user should receive notification that the password is about to expire;

16. The administrator should have a report where he can view all users in the system without going straight to the tables;

17. The administrator should be about to suspend any user from a start date to expiry date such as if the person is on an extended leave;

18. The administrator should get a report of all expired passwords;

19. The administrator should be able to send email for any user from within the system;

20. A username should be made of the first name initial, the full last name, and a four digit (two-digit month and two-digit year) of when the account is created;

These requirements for Sweetledger's user interface and log-in module prioritize user and account security along with a visually pleasing aspect to the system. The database is to be updated to contain all account information.

This concludes the approved requirements for Sweetledger's first sprint. Completion of these requirements will ensure that Sweetledger is on track for a scheduled release after the time frame of the final sprint.