Software Requirements Specification

for

MamoBooks

Version 1.0

**Prepared by Benny Villegas, Eddie Matos,
Stuart Tresslar, Zikomo Bullock, Brian Deguzis**

MamoCoin

**SWE 4713 Application Domain
Spring Semester, 2018
Kennesaw State University
Department of Software Engineering and Game Design**

24-April-2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Brian Deguzis	2/26/2018	Created the template for the document	0.1
Brian Deguzis, Eddie Matos, Stuart Tresslar, Zikomo Bullock, Benny Villegas	2/27/2018	Added Sections 1, 2 and 3.	0.2
Stuart Tresslar, Zikomo Bullock, Brian Deguzis	2/27/2018	Added Use Case Diagrams, descriptions, and the domain model for Module 1.	0.3
Zikomo Bullock	3/15/2018	Adjusted Use Case Diagrams and Descriptions for Module 2	0.4
Stuart Tresslar	3/21/2018	Added Functional Requirements for Module 1 and 2.	0.5
Eddie Matos	3/25/2018	Added Non-functional Requirements	0.6
Zikomo Bullock	3/28/2018	Updated Use Case Diagrams and Descriptions to Module 3.	0.7
Stuart Tresslar	3/28/2018	Added Module 3 requirements.	0.8
Zikomo Bullock	4/16/2018	Updated Use Case Diagrams and Descriptions	0.9
Zikomo Bullock Benny Villegas	4/18/2018	Updated all Requirements to Module 4.	1.0

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) documents the software requirements for the web based accounting application entitled Mamo Books. This specification covers version 1.0 of the system, and will include the entire scope of the system. Details on product scope can be found in Section 1.4 of this document.

1.2 Document Conventions

The basic template to be used is derived from IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans. Sections were eliminated and added as a team.

1.3 Intended Audience and Reading Suggestions

This document is intended to be ready by key stakeholders related to the system. The development team will refer to this document to ensure the product fully meets the system requirements. Whilst it is recommended the manager reads the entire document, special attention should be paid towards the business rules, functional, and nonfunctional requirements sections. The development team should study the use cases and respective diagram, the data flow diagram, and functional/nonfunctional requirements sections.

1.4 Product Scope

This software will be developed as a web based application for purpose of handling various accounting and financial functions. The functionalities provided will be but are not limited to:

- Creating/viewing and editing a chart of accounts
- Journalizing and viewing journalized transactions
- Posting transactions to accounts
- Creating and viewing trial balances
- Creating and viewing various financial statements
    - Income statements
    - Balance sheets
    - Cash flow statements
- Creating ration analysis

1.5 References

"Accounting Database Design." Simple Accounting - Database Design, simpleaccounting.freeservers.com/database.htm.

"Accounting Databases." Data Base Answers, www.databaseanswers.org/data_models/.

DigitalOcean. "LAMP Stack Tutorials | DigitalOcean." Tutorials | DigitalOcean, Digital Ocean, www.digitalocean.com/community/tags/lamp-stack?type=tutorials.

LinkedIn. "Lynda Accounting Principles ." Lynda Business and Accounting Library , LinkedIn, www.lynda.com/.
"WordPress.org." WordPress Codex, WordPress, codex.wordpress.org/.

2. Overall Description

2.1 Product Perspective

This product is a standalone application. The project was given to us by the client Dr. Mamo and is expected to be completed by late April. We are using an agile methodology with two week development cycles.

2.2 Product Functions

Through the software application the user will be able to...
-view the chart of accounts.
- journalize a transaction.
- post to the general ledger
- View individual ledger accounts

2.3 User Classes and Characteristics

User Class	**Pertinent System Functions**
Bookkeepers	Journalizing transactions, viewing chart of accounts
Accountants	Journalizing transactions, posting transactions to accounts, creating/viewing trial balances, creating/viewing financial statements, viewing chart of accounts,viewing ration analysis
Management Team Members	Approving/Rejecting journal entries, viewing trial balances, viewing financial statements, creating/editing/viewing chart of accounts, creating/viewing ration analysis
Marketing Team Members	Viewing financial statements
Finance Team Members	Viewing financial statements
Human Resource Team Members	Viewing financial statements

2.4 Operating Environment

The system will be a web based application and utilize the Digital Ocean programming environment, as well as LAMP framework, FTP Client server repository, Bracket text editor, and Browser Stack.

2.5 Design and Implementation Constraints

The system's primary constraint is being able to be accessed only via web browser. This means that users will never have direct access to the database, all access should be done through the system. User Interfaces must be created to ensure that users have the needed functionality that access to the database would provide.

2.6 User Documentation

The final product will be delivered along with a user manual detailing instructions for use.

2.7 Assumptions and Dependencies

Dependencies:
Our current project depends on modern Web Browsers to run correctly.

Assumptions:
We assume that the user will have general knowledge of Accounting practices.

3. External Interface Requirements

3.1 User Interfaces

The user interface for the software shall be compatible with web browsers such as Internet Explorer, Google Chrome, etc.

The user interface shall be implemented using tools such as DigitalOcean's server-side development, Linux, Apache, MySQL, and PHP (LAMP).

3.2 Hardware Interfaces

Since the application will be able to run on most web browsers currently available in the market, hardware necessary for internet connectivity is required. This includes, but is not limited to

modems and wireless routers. Due to the nature of this application, any device capable of running web browsers available in the market will be supported.

3.3 Software Interfaces

MamoBooks is to be developed for use on any web browser. It will accomplish this using HTML5, CSS, Javascript, MySQL, and PHP.

3.3.1 Incoming and outgoing items

- Outgoing data consists of transaction, account, and user information that are sent by users to the server.
- Incoming data consists of updates regarding accounts, users, and transactions.

3.3.2 Services and Communications

- MamoBooks will use server push and pull protocols
- Communication occurs on a user's web browser when the following situations occur:
    - When a user adds a journal entry
    - When a change is made in the chart of accounts
    - When a change is made in the users page

3.4 Communications Interfaces

MamoBooks will have a SQL database that is held on an Apache HTTP server. The database exists to house all information regarding an organization's financial data. PHP will be used to read and write information to the SQL database.

4. Domain Model

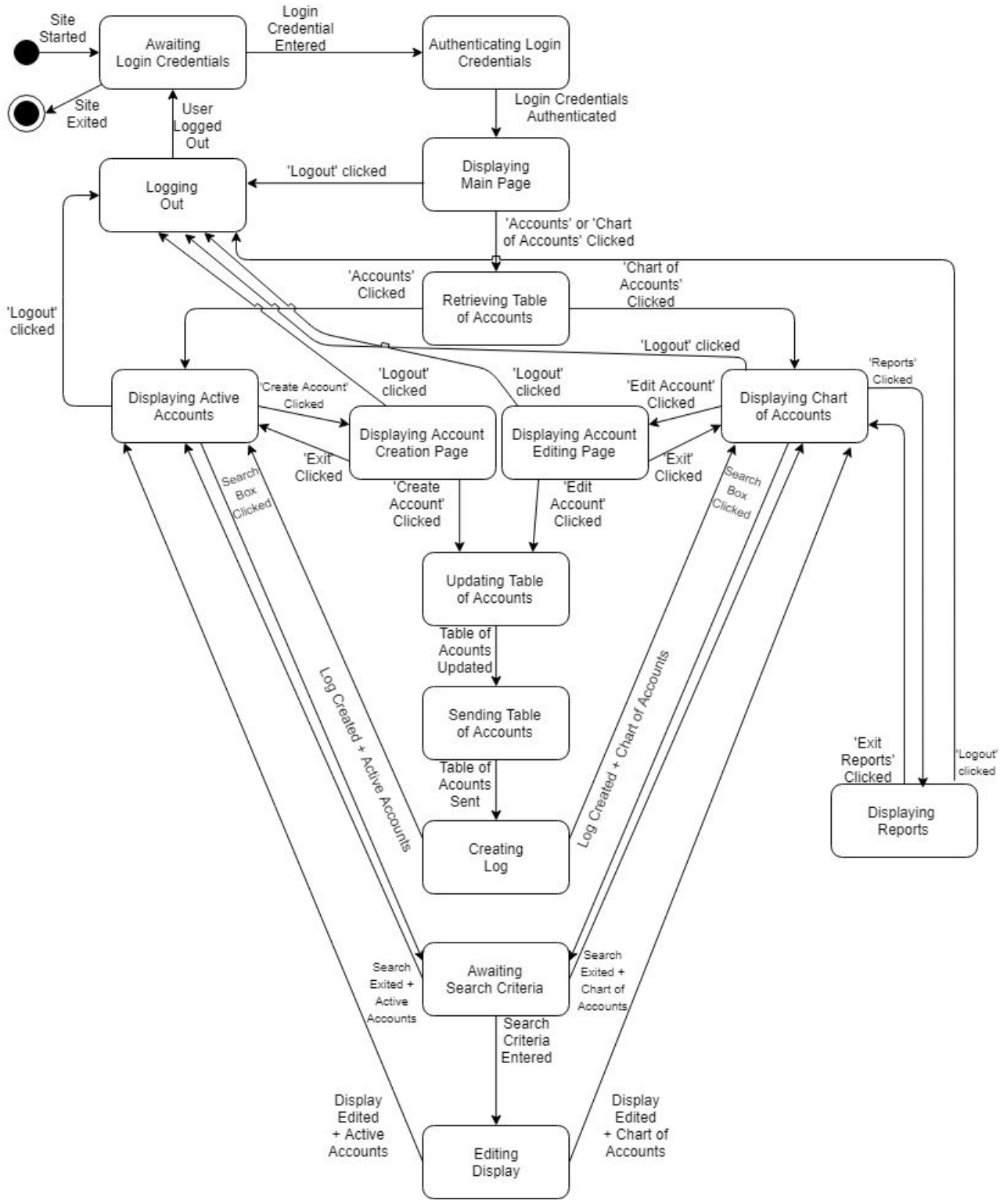

Figure 4.1 - Chart of Accounts State Chart Diagram

5. System Features (Use Cases)

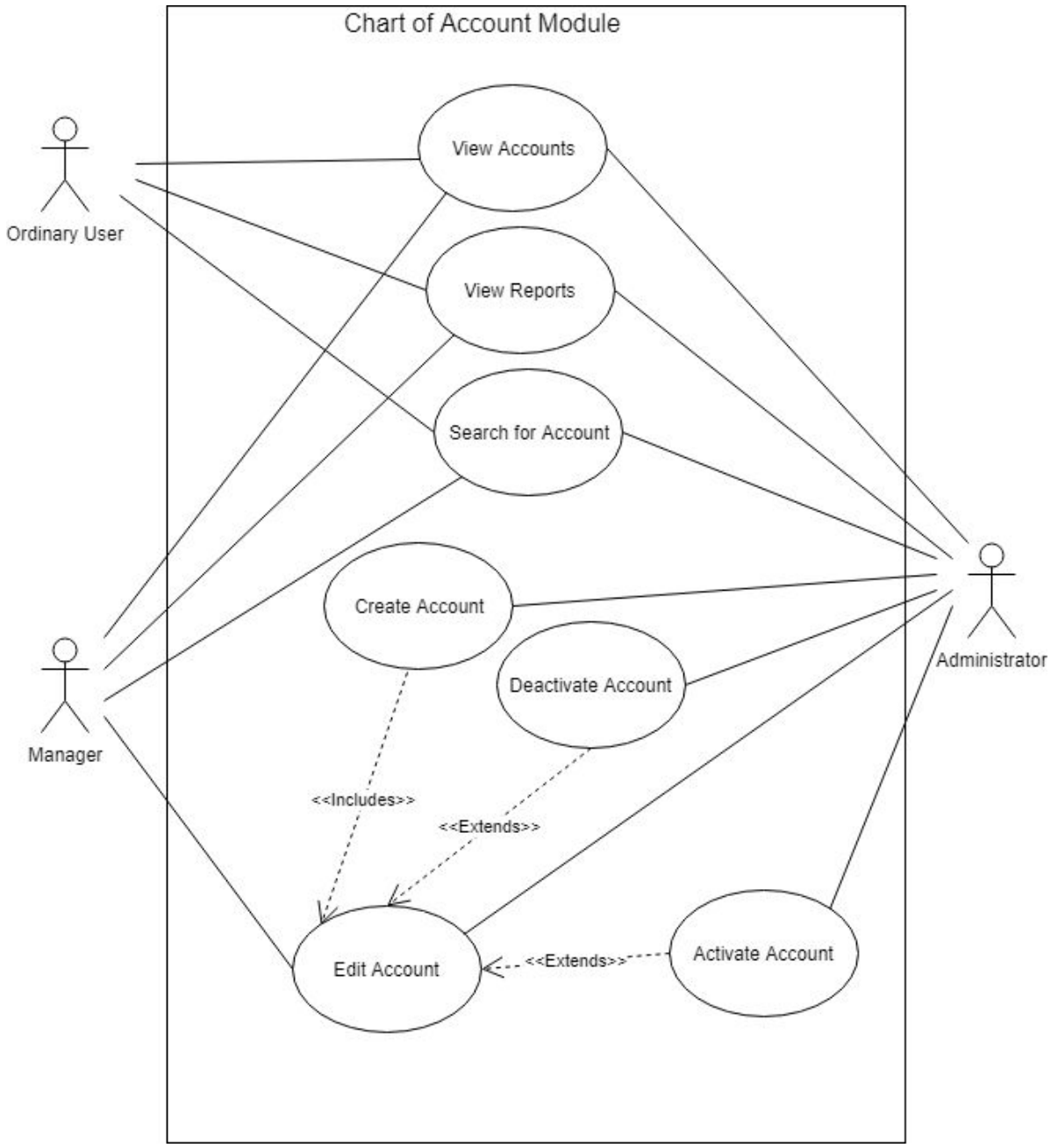

Figure 5.1 – Chart of Accounts Use Case Diagram

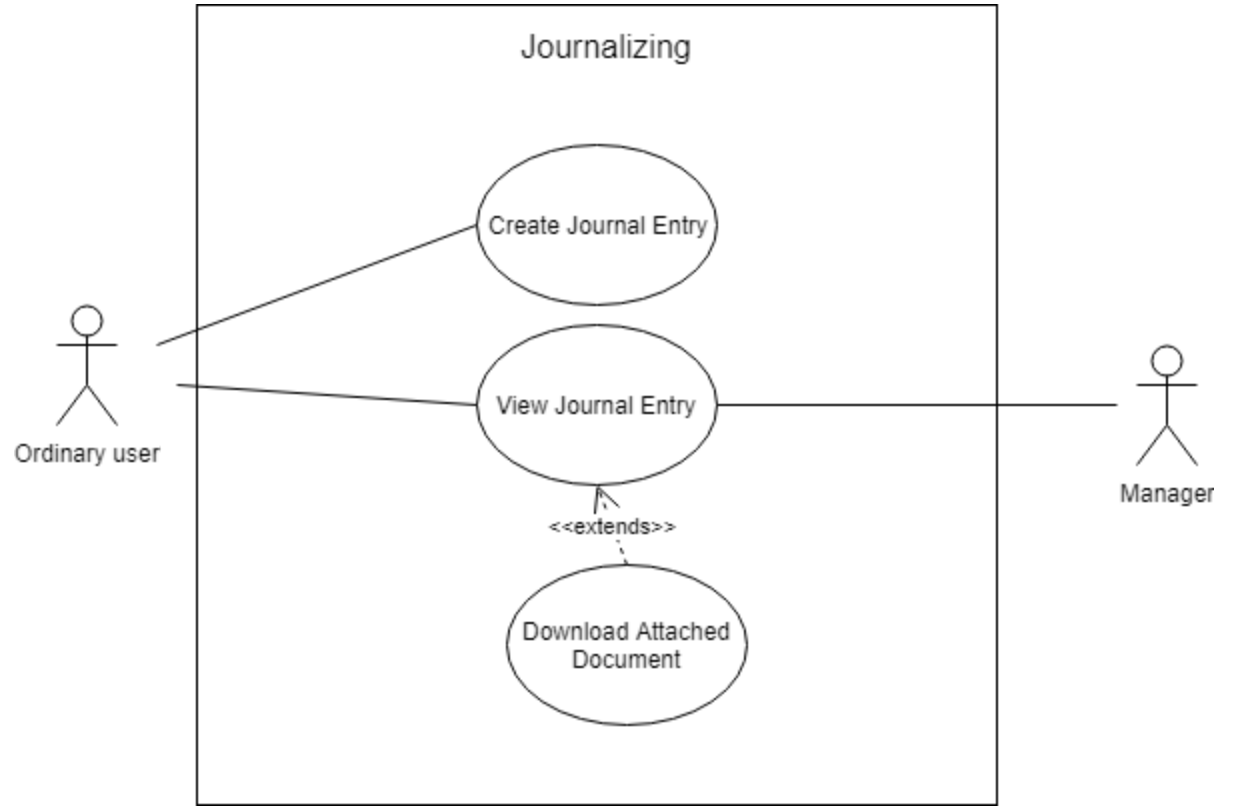

Figure 5.2 – Journalizing Use Case Diagram

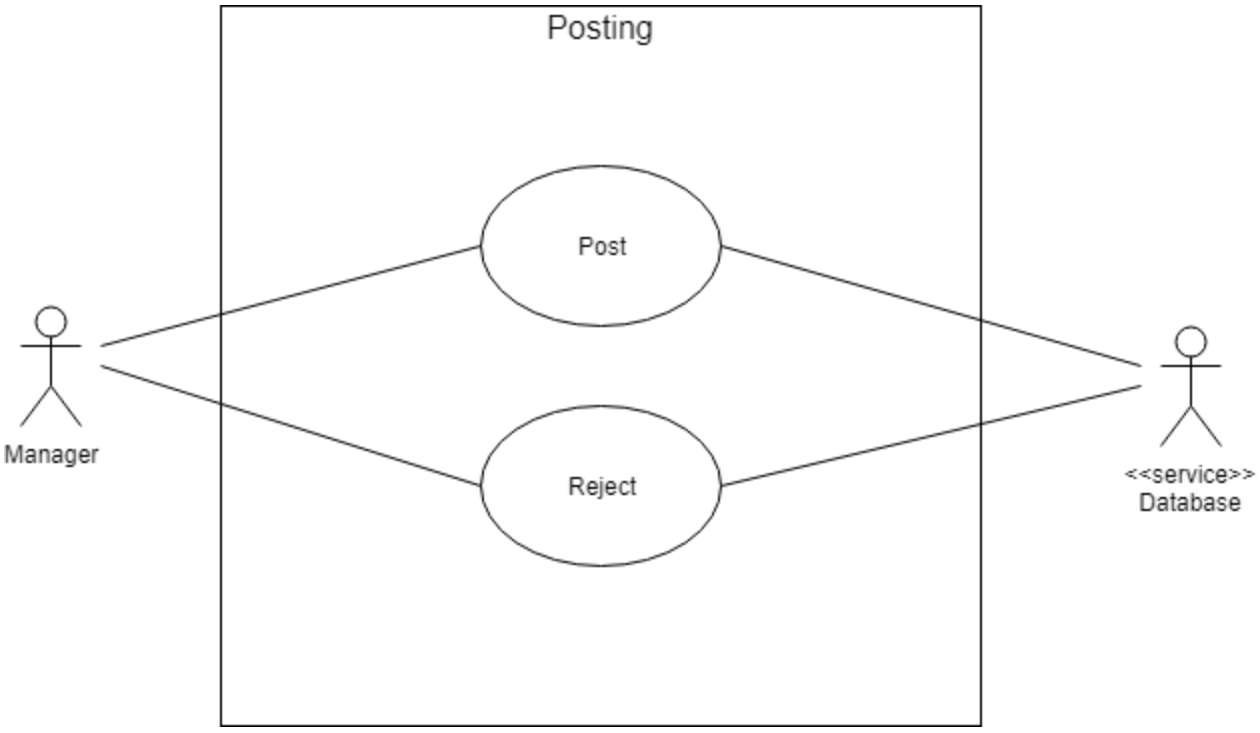

Figure 5.3 – Posting Use Case Diagram

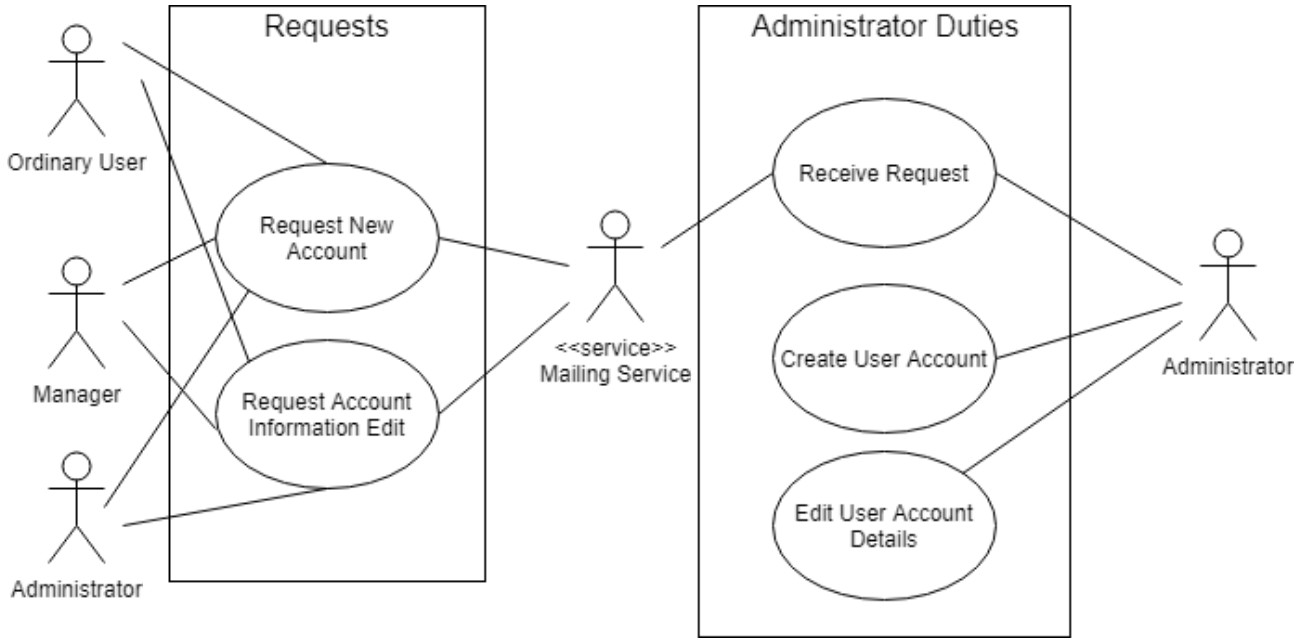

Figure 5.4 – System Administration Use Case Diagram

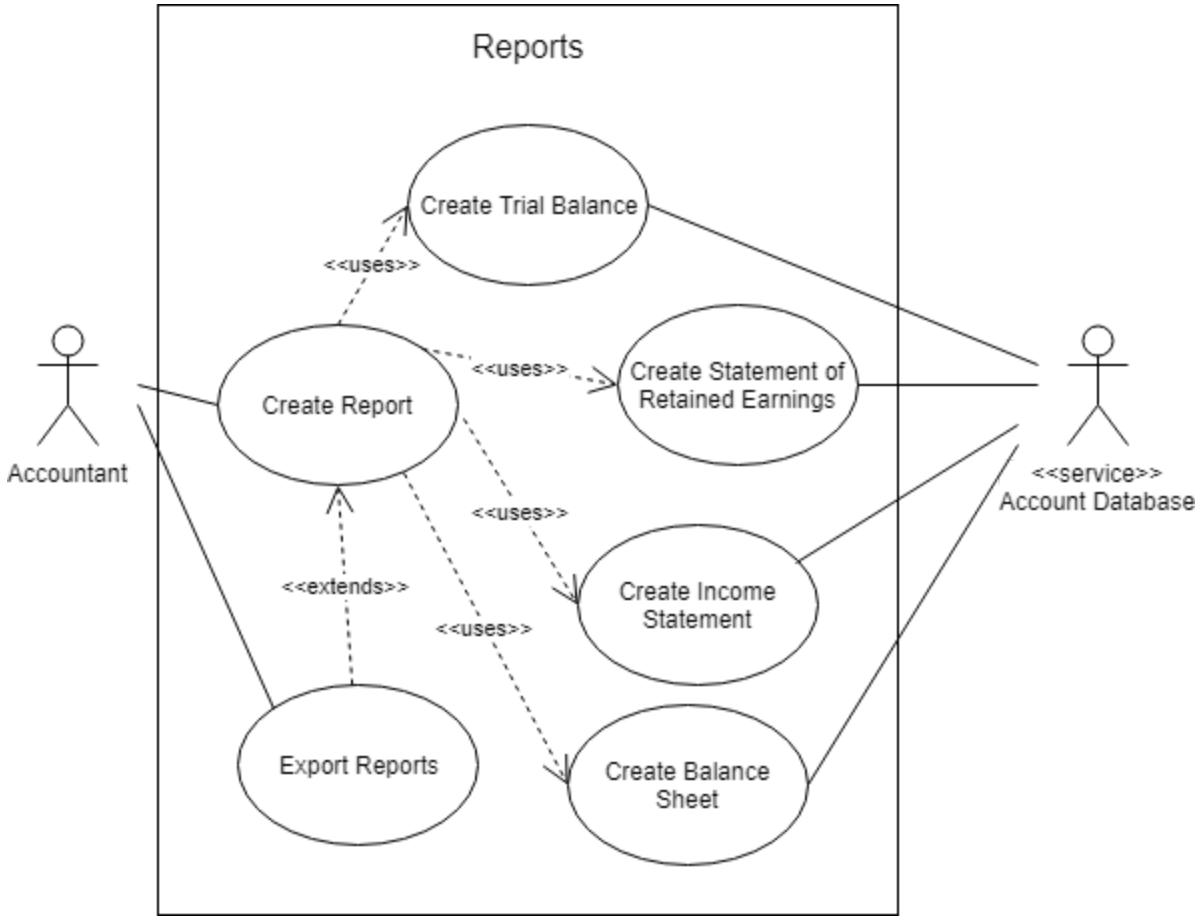

Figure 5.5 – Reports Use Case Diagram

5.1 Use Cases

ID	UC_CoA_001
Name	View Accounts
Author	Brian Deguzis
Date	February 27, 2018
Description	The user views the chart of accounts.
Actors	Ordinary User, Manager, Administrator
Preconditions	The user must be logged into their respective account on MamoBooks.
Assumptions	The chart of accounts in the MamoBooks database is populated with at least one active account.
Triggers	The user navigates to the “Chart of Accounts” page
Normal Flow	1. The website queries the MamoBooks database for the chart of accounts. 2. The website populates a table with all accounts labeled as “active” in the chart of accounts. 3. The website displays the populated table to the user. 4. The user can sort the table by: a. Account number b. Account name c. Category d. Sub-category e. Normal Side f. Initial Balance g. Date/time added h. User author
Exceptional Flow	1. The website queries the MamoBooks database for the chart of accounts. 2. There are no active accounts in the chart of accounts. 3. The website displays a message notifying the user that there are currently no active accounts in the chart of accounts.
Post Conditions	The website will display all accounts in the chart of accounts

	labeled as “active” in the form of a sortable table.

ID	UC_CoA_002
Name	View Reports
Author	Zikomo Bullock
Date	February 27, 2018
Description	The user views a report of all accounts
Actors	Ordinary User, Manager, Administrator
Preconditions	The user must be logged into their respective account on MamoBooks.
Assumptions	The chart of accounts within MamoBooks database is populated with at least one account.
Triggers	The user clicks the “View Report” button
Normal Flow	1. The user clicks on the “View Report” button. 2. A report based on each active account is generated. 3. The user is taken to MamoBook’s Chart of Accounts report page. 4. Reports can be downloaded or printed.
Exceptional Flow	1. The user clicks on the “View Report” button. 2. There are currently no accounts within MamoBook’s database 3. The website displays a message notifying the user that there are currently no active accounts in the chart of accounts.
Post Conditions	The website will display a report of all active accounts

ID	UC_CoA_003
Name	Search for Account
Author	Brian Deguzis
Date	February 27, 2018

Description	The user searches the chart of accounts by account number, account name, category, sub category, or user author.
Actors	Ordinary User, Manager, Administrator
Preconditions	The user must be logged into their respective account on MamoBooks.
Assumptions	None
Triggers	The user enters a search query into the search field on the chart of accounts page.
Normal Flow	1. The user chooses a search parameter to search the chart of accounts with from the following options: a. Account number b. Account name c. Category d. Sub Category 2. The user submits a search query from the search field on the chart of accounts page. 3. The website queries the MamoBooks database for the chart of accounts limited by the user's search query. 4. The website updates the chart of accounts table to show only items of the search parameter specified by the user, limited by the user's search query.
Alternate Flow	1. The user chooses a search parameter to search the chart of accounts with from the following options: a. Account number b. Account name c. Category d. Sub Category e. User Author (Administrator only) 2. The user submits a search query from the search field on the chart of accounts page. 3. The website queries the MamoBooks database for the chart of accounts limited by the user's search query. 4. The website updates the chart of accounts table to show only items of the search parameter specified by the user, limited by the user's search query.
Post Conditions	The website displays the chart of accounts, showing only items matching the user's search query.

ID	UC_CoA_004

Name	Create Account
Author	Brian Deguzis
Date	February 27, 2018
Description	The administrator adds a new account to the chart of accounts.
Actors	Administrator
Preconditions	The administrator must be logged into their respective account on MamoBooks.
Assumptions	None
Triggers	The administrator clicks on the “Create account” button.
Normal Flow	1. The administrator is prompted to fill in the following information about the account they would like to create: a. An account name selected from a dropdown box b. An account number c. A category d. A sub-category e. The normal Side f. The initial Balance 2. The user either cancels the account creation by clicking the “Cancel” button, or clicks on the “Create Account” button to finalize the creation of the account. 3. The account’s “Date/Time Created” and “User Author” values are auto created by MamoBooks. 4. The website queries the MamoBooks database to move the new account into the chart of accounts, and to change its respective values to match the information entered in step 1 of this use case. 5. The website updates the chart of accounts table to reflect the changes in the database.
Post Conditions	A new account is added to the chart of accounts.

ID	UC_CoA_005
Name	Deactivate Account
Author	Brian Deguzis

Date	February 27, 2018
Description	The administrator changes the status of an account in the chart of accounts from “Active” to “Inactive.”
Actors	Administrator
Preconditions	1. The administrator must be logged into their respective account on MamoBooks. 2. There exists at least one active account in the chart of accounts that has no transaction history.
Assumptions	None
Triggers	The administrator clicks on the “Deactivate Account” button for an account.
Normal Flow	1. The website queries the MamoBooks database to change the “Status” variable of the respective account from “Active” to “Inactive” 2. The website updates the chart of accounts table to reflect the changes in the database. 3. The “Deactivate Account” button is changed to the “Activate Account” button
Post Conditions	The status of the chosen account in the chart of accounts is changed from “Active” to “Inactive.”

ID	UC_CoA_006
Name	Edit Account
Author	Brian Deguzis
Date	February 27, 2018
Description	The manager or administrator is prompted to change any or all of the following: the account number, account name, category, sub-category, normal side, or user author of an account in the chart of accounts.
Actors	Manager, Administrator
Preconditions	1. The manager or administrator must be logged into their respective account on MamoBooks. 2. There exists at least one account in the chart of accounts.

Assumptions	None
Triggers	The manager or administrator click on the “Edit Account” button for an account.
Normal Flow	1. The website displays the edit screen for changing account information. 2. The website prompts the user to change any of the following account information: a. The account name selected from a dropdown box b. The account number c. The category d. The sub-category e. The normal Side 3. The user either clicks the “Cancel” button to cancel editing the account, or the user clicks the “Save” button to save the edits to the account. 4. The website queries the MamoBooks database to change the account requested by the user, to reflect the changes made in step 2 of this use case. 5. The website updates the chart of accounts table to reflect the changes in the database.
Post Conditions	The account information is modified according to user’s input.

ID	UC_CoA_007
Name	Activate Account
Author	Brian Deguzis
Date	February 27, 2018
Description	The administrator changes the status of an account in the chart of accounts from “Active” to “Inactive.”
Actors	Administrator
Preconditions	1. The administrator must be logged into their respective account on MamoBooks. 2. There exists at least one inactive account in the chart of accounts.
Assumptions	None
Triggers	The administrator clicks on the “Activate Account” button for an

	account.
Normal Flow	1. The website queries the MamoBooks database to change the “Status” variable of the respective account from “Inactive” to “Active” 2. The website updates the chart of accounts table to reflect the changes in the database. 3. The “Activate Account” button is changed to the “Deactivate Account” button
Post Conditions	The status of the chosen account in the chart of accounts is changed from “Inactive” to “Active.”

ID	UC_J_001
Name	Create Journal Entry
Author	Zikomo Bullock
Date	March 15, 2018
Description	The user enters in data regarding the transaction entry for the journal.
Actors	Ordinary User, Manager
Preconditions	1. A transaction has taken place. 2. An Ordinary User or Manager must be logged into their account on MamoBooks.
Assumptions	None
Triggers	The user clicks on the “Add Journal Entry” button.
Normal Flow	1. The user adds the date of the transaction, which takes place on either the current date or a date in the future. 2. The user clicks on the “Add journal Entry button” 3. The user selects the two accounts involved in the transaction 4. The user then adds a monetary amount to either the Debit and Credit accounts of each account in the transaction 5. A comment is left regarding the transaction. 6. The user clicks on the “Save Journal Entry” button, and the transaction is saved to the General Journal.
Alternate Flow	1. The user adds the date of the transaction, which takes place on either the current date or a date in the future.

	2. The user clicks on the “Add journal Entry button” 3. The user selects the two accounts involved in the transaction 4. The user then adds a monetary amount to either the Debit and Credit accounts of each account in the transaction 5. A comment is left regarding the transaction. 6. A document relevant to the transaction is attached 7. The user clicks on the “Save Journal Entry” button, and the transaction is saved to the General Journal.
Post Conditions	The transaction has been journalized.

ID	UC_J_002
Name	Attach Document
Author	Zikomo Bullock
Date	March 15, 2018
Description	The user attaches a transaction document to a journal entry
Actors	Ordinary User
Preconditions	1. A transaction document is available 2. The user is currently creating a journal entry
Assumptions	None
Triggers	The user clicks on the “Attach document” button within the “Journal Entry Creation” page
Normal Flow	1. The System prompts the user to supply the document a. A window displays all directories within the users’ systems to select the file 2. Mamobooks database is queried with a request to update the journal entry with the attached document
Post Conditions	The document has been attached to the journal entry

ID	UC_J_003
Name	View Journal Entry
Author	Zikomo Bullock

Date	March 15, 2018
Description	The user wishes to view a specific journal entry
Actors	Ordinary User, Manager
Preconditions	1. A journal entry must be present 2. An Ordinary User or Manager must be logged into their account in MamoBooks
Assumptions	None
Triggers	The user clicks on a journal entry within the General Journal
Normal Flow	1. The Mamobooks database is queried with a request to view the specific journal entry 2. The specific information of the journal entry are provided as a response
Post Conditions	The journal entry and its specifics are now visible

ID	UC_J_004
Name	Download Attached Document
Author	Zikomo bullock
Date	March 15, 2018
Description	The document attached to a journal entry is downloaded
Actors	Ordinary User, Manager
Preconditions	1. A document must have been previously attached to a journal entry 2. An Ordinary User or Manager must be logged into their account in MamoBooks
Assumptions	None
Triggers	When viewing a journal entry, the user selects the “Download attached file” button.
Normal Flow	1. The user is prompted to select a location to save the document

	2. The system receives the save location for the file. a. Mamobooks database is queried with a request to retrieve the attached file 3. The document begins downloading to the save location
Post Conditions	The attached document has been downloaded

ID	UC_Post_001
Name	Post
Author	Brian Deguzis
Date	March 15, 2018
Description	The journal entry is posted to the corresponding account ledger.
Actors	Manager
Preconditions	1. The manager must be logged into their respective account on MamoBooks. 2. A journal entry is recorded.
Assumptions	None
Triggers	The Manager clicks on the “Post” button.
Normal Flow	1. The credit and debit accounts in the journal entry are recorded to their respective ledgers. a. The MamoBooks database is queried with a request to update the respective account ledgers with the transactions from the journal. 2. The corresponding ledger on the accounts page is updated to reflect the changes made by the journal entry.
Post Conditions	The journal entry has been posted.

ID	UC_Post_002
Name	Reject
Author	Zikomo Bullock
Date	March 15, 2018

Description	A journal entry is rejected from being added to the corresponding account ledger.
Actors	Manager
Preconditions	1. The manager must be logged into their respective account on MamoBooks. 2. A journal entry is recorded.
Assumptions	None
Triggers	The Manager clicks the “Reject” button.
Normal Flow	1. The manager leaves a comment as to why the entry has been rejected. 2. The system prompts the user to confirm the rejection. 3. The journal entry has been removed from the MamoBooks database.
Post Conditions	The journal entry in question has been rejected.

ID	UC_Req_001
Name	Request New Account
Author	Zikomo Bullock
Date	April 13, 2018
Description	The user requests the Administrator to make a new account.
Actors	Ordinary User, Manager, Administrator, Mailing Service
Preconditions	The user must not previously have an account with MamoBooks.
Assumptions	Administrators are responsible for the creation of all accounts.
Triggers	The user clicks on the “Request account” button.

Normal Flow	1. An Account Request window is shown to the user requiring a request summary and email address 2. The user inputs data into these fields 3. The submit button is clicked and the mailing service sends this request to the Administrator. 4. The Account Request window is closed.
Alternate Flow	1. An Account Request window is shown to the user requiring a request summary and email address. 2. The cancel button is clicked. 3. The Account Request window is closed.
Post Conditions	A request has been made for a new account.

ID	UC_Req_002
Name	Request Account Information Edit
Author	Zikomo Bullock
Date	April 13, 2018
Description	The user requests the Administrator to edit information associated with the account.
Actors	Ordinary User, Manager, Administrator, Mailing Service
Preconditions	The user must currently possess an account with MamoBooks.
Assumptions	Administrators are responsible for the editing of information for accounts.
Triggers	The user clicks on the “Request Account Editing” button.
Normal Flow	1. An Account Editing request window appears requiring a request summary and email address, and the fields to be changed. 2. The user inputs data into these fields. 3. The submit button is clicked and the mailing service sends this request to the Administrator. 4. The Account Editing request window is closed.
Alternate Flow	1. An Account Editing request window appears requiring a request summary, email address, and the fields to be changed. 2. The user clicks the “cancel” button. 3. The Account Editing Request window is closed.

Post Conditions	A request has been made for a new account.

ID	UC_AD_001
Name	Receive Request
Author	Zikomo Bullock
Date	April 13, 2018
Description	The Administrator receives the request from the mailing service.
Actors	Administrator and Mailing Service
Preconditions	The user must be an Administrator of MamoBooks and requests must have previously been made.
Assumptions	N/A
Triggers	The mailing service forwards requests to the Administrator
Normal Flow	1. The requests arrive in the Administrator's mailbox within MamoBooks. a. Requests are ordered by date received b. Can be ordered by other factors such as request status, and by sender 2. The administrator views these requests.
Post Conditions	A request has been made for a new account.

ID	UC_AD_002
Name	Create User Account
Author	Zikomo Bullock
Date	April 13, 2018
Description	The Administrator creates a new account with MamoBooks
Actors	Administrator
Preconditions	The user must be an Administrator of MamoBooks and requests must have previously been made.

Assumptions	N/A
Triggers	The “Create User” Button is clicked.
Normal Flow	1. A prompt opens with fields for the new user to be completed. This includes a: a. Username b. Password c. Email Address d. User Type such as “Accountant”, “Manager”, or “Administrator” 2. The “Create” button is clicked. 3. A request is made to the MamoBooks user database. 4. The database responds, and a new entry is input into the database containing the data from the new user’s fields. 5. A new user has been added to the database. 6. The prompt is closed. 7. A message stating that the User has been created appears on the web page.
Alternate Flow	Ex. 1: 1. A prompt opens with fields for the new user to be completed. 2. The user clicks “cancel”. 3. The prompt is closed. Ex. 2: 1. A prompt opens with fields for the new user to be completed. This includes a: a. Username b. Password c. Email Address d. User Type such as “Accountant”, “Manager”, or “Administrator” 2. The “Create” button is clicked. 3. A request is made to the MamoBooks user database. 4. The database is unable to be reached. 5. The prompt is closed. 6. An error code and description appear on the web page.
Post Conditions	A new account has been created.

ID	UC_AD_003
Name	Edit User Account Details
Author	Zikomo Bullock

Date	April 13, 2018
Description	The Administrator edits an account with provided details from an Account Editing Request.
Actors	Administrator
Preconditions	The user must be an Administrator of MamoBooks and an Account Edit Request must be present.
Assumptions	N/A
Triggers	The “Edit” Button beside a user is clicked.
Normal Flow	1. A window opens with fields for the user to be edited. This includes a: a. Username b. Password c. Email Address d. User Type such as “Accountant”, “Manager”, or “Administrator” 2. The fields needing editing are changed. 3. The “Edit” button is clicked. 4. A request is made to the MamoBooks user database. 5. The database responds, and the entry is edited. 6. The selected user’s information has been changed. 7. The prompt is closed. 8. A message stating that the User has been edited appears on the web page.
Alternate Flow	Ex. 1: 1. A window opens with fields for the user to be edited. 2. The user clicks “cancel”. 3. The window is closed. Ex. 2: 1. A prompt opens with fields for the user to be edited. This includes a: a. Username b. Password c. Email Address d. User Type such as “Accountant”, “Manager”, or “Administrator” 2. The “Edit” button is clicked. 3. A request is made to the MamoBooks user database. 4. The database is unable to be reached. 5. The window is closed. 6. An error code and description appear on the web page.

Post Conditions	The User has been edited.

ID	UC_Reports_001
Name	Create Report
Author	Zikomo Bullock
Date	April 13, 2018
Description	An Accountant is able to create a Report using Accounts within MamoBooks.
Actors	Accountant, Manager, Account Database
Preconditions	There are currently accounts within the Account Database.
Assumptions	N/A
Triggers	The Accountant clicks on the “Reports” tab within MamoBooks.
Normal Flow	1. The web page prompts the user to enter information regarding the report to be generated. This includes: a. Time Period b. End Date c. Report Type 2. The “Create Report” button is clicked. 3. A request is made to the Accounts Database within MamoBooks. 4. The request is received by the database, and all information regarding the accounts is provided for the reports to be generated. 5. A report is generated using information stored in the Accounts database.
Alternate Flow	1. The web page prompts the user to enter information regarding the report to be generated. 2. The “Create Report” button is clicked. 3. A request is made to the Accounts Database within MamoBooks. 4. The request is received by the database, and all information regarding the accounts is provided for the reports to be generated. 5. A report is generated using information stored in the Accounts database.
Post Conditions	A Report has been prepared.

ID	UC_Reports_002
Name	Create Trial Balance
Author	Zikomo Bullock
Date	April 13, 2018
Description	An Accountant is able to create a Trial Balance using Accounts within MamoBooks.
Actors	Accountant, Manager, Account Database
Preconditions	There are currently accounts within the Account Database.
Assumptions	N/A
Triggers	The Accountant clicks on the “Reports” tab within MamoBooks.
Normal Flow	6. The web page prompts the user to enter information regarding the report to be generated. This includes: a. Time Period b. End Date 7. The Report type is “Unadjusted Trial Balance” 8. The “Create Report” button is clicked. 9. A request is made to the Accounts Database within MamoBooks. 10. The request is received by the database, and all information regarding the accounts is provided for the reports to be generated. 11. A report is generated using information stored in the Accounts database.
Alternate Flow	Ex.1: 6. The web page prompts the user to enter information regarding the report to be generated. 7. The Report Type is “Adjusted Trial Balance” 8. The “Create Report” button is clicked. 9. A request is made to the Accounts Database within MamoBooks. 10. The request is received by the database, and all information regarding the accounts is provided for the reports to be generated. 11. A report is generated using information stored in the Accounts database. Ex. 2:

	1. The web page prompts the user to enter information regarding the report to be generated. 2. The Report Type is “Post-Closing Trial Balance” 3. The “Create Report” button is clicked. 4. A request is made to the Accounts Database within MamoBooks. 5. The request is received by the database, and all information regarding the accounts is provided for the reports to be generated. 6. A report is generated using information stored in the Accounts database. Ex. 3: 1. The web page prompts the user to enter information regarding the report to be generated. 2. The “Create Report” button is clicked. 3. A request is made to the Accounts Database within MamoBooks. 4. There is no response from the database. 5. An error code and description are produced.
Post Conditions	A Trial Balance has been prepared.

ID	UC_Reports_003
Name	Create Statement of Retained Earnings.
Author	Zikomo Bullock
Date	April 13, 2018
Description	An Accountant is able to create a Statement of Retained Earnings using Accounts within MamoBooks.
Actors	Accountant, Manager, Account Database
Preconditions	There are currently accounts within the Account Database.
Assumptions	N/A
Triggers	The Accountant clicks on the “Reports” tab within MamoBooks.
Normal Flow	1. The web page prompts the user to enter information regarding the report to be generated. This includes: a. Time Period b. End Date 2. The Report type is ”Statement of Retained Earnings”

	3. The “Create Report” button is clicked. 4. A request is made to the Accounts Database within MamoBooks. 5. The request is received by the database, and all information regarding the accounts is provided for the reports to be generated. 6. A report is generated using information stored in the Accounts database.
Alternate Flow	Ex.1: 1. The web page prompts the user to enter information regarding the report to be generated. 2. The “Create Report” button is clicked. 3. A request is made to the Accounts Database within MamoBooks. 4. There is no response from the database. 5. An error code and description are produced.
Post Conditions	A Statement of Retained Earnings is created.

ID	UC_Reports_004
Name	Create Income Statement
Author	Zikomo Bullock
Date	April 13, 2018
Description	An Accountant is able to create an Income Statement using Accounts within MamoBooks.
Actors	Accountant, Manager, Account Database
Preconditions	There are currently accounts within the Account Database.
Assumptions	N/A
Triggers	The Accountant clicks on the “Reports” tab within MamoBooks.
Normal Flow	1. The web page prompts the user to enter information regarding the report to be generated. This includes: a. Time Period b. End Date 2. The Report type is “Income Statement” 3. The “Create Report” button is clicked. 4. A request is made to the Accounts Database within MamoBooks.

	5. The request is received by the database, and all information regarding the accounts is provided for the reports to be generated. 6. A report is generated using information stored in the Accounts database.
Alternate Flow	Ex. 1: 1. The web page prompts the user to enter information regarding the report to be generated. 2. The “Create Report” button is clicked. 3. A request is made to the Accounts Database within MamoBooks. 4. There is no response from the database. 5. An error code and description are produced.
Post Conditions	An Income Statement is created.

ID	UC_Reports_005
Name	Create Balance Sheet
Author	Zikomo Bullock
Date	April 13, 2018
Description	An Accountant is able to create a Balance Sheet using Accounts within MamoBooks.
Actors	Accountant, Manager, Account Database
Preconditions	There are currently accounts within the Account Database.
Assumptions	N/A
Triggers	The Accountant clicks on the “Reports” tab within MamoBooks.
Normal Flow	1. The web page prompts the user to enter information regarding the report to be generated. This includes: a. Time Period b. End Date 2. The Report type is “Balance Sheet” 3. The “Create Report” button is clicked. 4. A request is made to the Accounts Database within MamoBooks. 5. The request is received by the database, and all information regarding the accounts is provided for the reports to be generated.

	6. A report is generated using information stored in the Accounts database.
Alternate Flow	Ex. 1: 1. The web page prompts the user to enter information regarding the report to be generated. 2. The “Create Report” button is clicked. 3. A request is made to the Accounts Database within MamoBooks. 4. There is no response from the database. 5. An error code and description are produced.
Post Conditions	A Balance Sheet is created.

ID	UC_Reports_006
Name	Export Reports
Author	Zikomo Bullock
Date	April 13, 2018
Description	An Accountant is able to export or download Reports for record keeping.
Actors	Accountant, Manager, Account Database
Preconditions	There are currently accounts within the Account Database.
Assumptions	N/A
Triggers	The Accountant clicks on the “Export” option when a Report has already been generated.
Normal Flow	1. A window appears prompting the user to choose a save location. 2. A name and type for the file is selected 3. When the save button is clicked, the file will begin downloading
Post Conditions	The report has been downloaded.

6. Functional Requirements

6.1 Chart of Accounts

The Chart of Accounts feature allows for the users to display all accounts, create, edit, activate, deactivate, and search for accounts based on various criteria. The accessibility of these features depends on the user's level within the system and permissions.

6.1.1 Interface Requirements

INT.1.1 – The system should be able to display all created accounts.
INT.1.2 – The system should only display the Account Number, Account Name, Category, Sub-Category, Current Balance, and Is Active entries to normal users and managers.
INT.1.3 – In addition to the entries listed in INT.1.2, the system should display the Normal Side, Initial Balance, Date/Time Added, and User Author entries to administrators.
INT.1.4 – The system should display the Is Active column as a Y for active and N for inactive.
INT.1.5 – The system should display a message to the user that there are no accounts for the specified sort/search parameters.
INT.1.6 – The system should have a View Report button.
INT.1.7 – On account creation, the system should only allow users to input the account category through a list of pre-created categories.
INT.1.8 – On account creation, the system should only allow users to enter the Is Active entry through a checkbox.
INT.1.9 – The system should have a Cancel button on the account creation page.
INT.1.10 - The system should have a Create Account button the the account creation page.
INT.1.11 - The system should have an Edit button next to each listed account on the Chart of Accounts that takes the user to the Edit Account page.
INT.1.12 - The system should have a Cancel button on the Edit Account page that takes the user back to the previous page.

6.1.2 Operational Requirements (a.k.a. Functional Capabilities)

OPR.1.1 – The system should have a database that holds the different accounts.
OPR.1.2 – The system should have an accounts table in the database that has the following columns: Account Number, Account Name, Category, Sub-Category, Normal Side, Initial Balance, Current Balance, Is Active, Date/Time Added, and Account Author. The system should be able to access the database to retrieve the accounts table.
OPR.1.3 – The system should be able to retrieve the accounts table from the database.
OPR.1.4 – The system should allow users to create a new account without having direct access to the database.
OPR.1.5 – The system should be able to sort the accounts table by any of its displayed headings.
OPR.1.6 – The system should allow the user to type into a search bar.

OPR.1.7 – The system should sort through the table according to whatever is entered into the search bar.
OPR.1.8 – The system should allow any user to create an account by entering the following info: Account Category, Account Name, Normal Side, Initial Balance, and Is Active.
OPR.1.9 – The system should have a separate account creation page.
OPR.1.10 – The system should allow the user to click the Cancel button to be taken back to the previous page.
OPR.1.11 – The system should allow the user to click the Create Account button to create an entry into the database based on the entered information.
OPR.1.12 – The system should allow the user to edit an account from the Accounts page.
OPR.1.13 – The system should take the user to an Edit Account page when the user wants to edit an account.
OPR.1.14 – The system should have an Edit Account page where the user can change the Account Name, Account Number, Category, Sub-Category, Normal Side, and IsActive sections of the Account.
OPR.1.15 – The system should be able to change the edited account info in the database when the user is done editing the account info from the Edit Account page.

6.1.3 Data Requirements

DAT.1.1 – On account creation, the system should interpret a check in the Is Active checkmark box as Y and no check as N.
DAT.1.2 – When an account is created and entered into the database, the system should enter in the current time as the Date/Time Added section in the Accounts Table.
DAT.1.3 – When an account is created and entered into the database, the system should enter in the user who created the account as the Account Author section in the Accounts Table.

6.2 Journalizing

6.2.1 Interface Requirements

INT.2.1 – The main task bar at the top of the site should have a “Journalize” button that takes the user to the Journal page.
INT.2.2 – The Journal page should have an “Add Journal Entry” button that takes the user to the New Journal Entry page.
INT.2.3 – The New Journal Entry page should have a drop-down calendar for inputting the date.
INT.2.4 – The dropdown calendar for the date should only allow users to click on dates seven days before the current date or any dates after the current date, including the current date.
INT.2.5 – The New Journal Entry page should have a “Cancel” button that takes the user back to the previous page.
INT.2.6 – The date entry should be the top-most selection in the New Journal Entry page.
INT.2.7 – The summary section should be the bottom-most field in the New Journal Entry page.

INT.2.8 – In the New Journal Entry page, any accounts that are being debited should be listed before the accounts that are being credited.
INT.2.8 – In the New Journal Entry page, there should be a “Submit Journal Entry” button that uploads the journal entry to the Journal Table and takes the user to the Journal page.
INT.2.9 – The “Submit Journal Entry” button should not be able to be clicked if the total values being debited do not match the total values being credited.
INT.2.10 – An error should be displayed if the total values being debited do not match the total values being credited.
INT.2.11 – The “Submit Journal Entry” button should not be able to be clicked if the same account is present twice or more in a journal entry.
INT.2.12 – An error should be displayed if the the same account is present twice or more in a journal entry.
INT.2.13 – There should be a way to denote how many journal entries you want to see on each page of the Journal web-page.

6.2.2 Operational Requirements

OPR.2.1 – The system should allow users to view all created journal entries on a Journal page.
OPR.2.2 – The system should be able to create journal entries and store them in a Journal Table in the database.
OPR.2.3 – The system should have a New Journal Entry page where users can select a date and input an account, whether it is being Debited or Credited, for which amount, a summary of the transaction, and any files that need to be included with it.
OPR.2.4 – The system should have the ability to add in more than one account in the New Journal Entry page.
OPR.2.5 – The system should have the ability to upload documents.
OPR.2.6 – The system should have the ability to attach documents to a journal entry.
OPR.2.7 – The system should have the ability to check if the total values being debited match the total values being credited.
OPR.2.8 – The system should have the ability to check if the same account is present twice or more in a journal entry.
OPR.2.9 – The system should be able to send created journal entries to all managers to be approved or denied.
OPR.2.10 – The system should be able to display all approved and pending journal entries on the Journal page.
OPR.2.11 – The system should be able to search/filter by any data field in the Journal page.
OPR.2.12 – The system should allow users to download any attached documents to a journal entry from the Journal page.
OPR.2.13 – The system should be able to limit how many journal entries are displayed on each page of the Journal web-page.
OPR.2.14 – The system should be able to move between journal entry pages on the Journal web-page.

6.2.3 Data Requirements

DAT.2.1 – The Journal Table should include columns for Account, Debit/Credit, Amount, Summary, Files, Date, and Approval Status.

6.3 Posting

6.3.1 Interface Requirements

INT.3.1 – For managers, there should be an “Post” button next to Journal entries displayed on the Journal page that have yet to be approved or denied.
INT.3.2 – For managers, there should be an “Deny” button next to Journal entries displayed on the Journal page that have yet to be approved or denied.
INT.3.3 – When the “Deny” button is clicked, a prompt should appear asking if the manager is sure they want to be denied, along with the ability to add a reason why they are denying it.
INT.3.4 – When the “Deny” button is clicked, it is mandatory that there is a stated reason why they are denying the journal entry.

6.3.2 Operational Requirements

OPR.3.1 – The system should allow managers to input a reason for why they denied a journal entry.

6.3.3 Data Requirements

DAT.3.1 – When the “Post” button is clicked on the Journal page, the Approval Status of the journal entry should be changed to “Approved”.
DAT.3.2 – When the “Deny” button is clicked on the Journal page, the Approval Status of the journal entry should be changed to “Denied” with the addition of the reason stated for denying the journal entry.

6.4 Financial Reporting

6.4.1 Interface Requirements

INT.4.1 – The Reports page should have a dropdown for inputting the Report Time Period.
INT.4.2 – The Reports page should have a dropdown calendar for inputting the date of the report’s creation.
INT.4.3 – The Reports page should have a dropdown for inputting the Report Type.
INT.4.4 – The Reports page should have a “Create Report” button that creates a report in the page.
INT.4.6 – The report’s total balances should have two bars below them.
INT.4.7 – The report’s final debit and credit values should have a bar below them.

INT.4.8 – The report should have all debits to the left, and credits to the right.
INT.4.9 – The report should have all assets come before all liabilities and equity.
INT.4.10 – The report should have all revenue and expenses come below the liability and equity.
INT.4.11 – The Reports page should have a “Export Report” button that allows the user to download the report.
INT.4.12 – An error should be displayed if the “Create Report” button is clicked when there are no accounts within the system.

6.4.2 Operational Requirements

OPR.4.1 – The system should be able to create different types of financial statements.
OPR.4.2 – The system should allow for financial statements to be downloaded.
OPR.4.3 – The system should be able to order the accounts present in the report by type.

6.4.3 Data Requirements

DAT.4.1 – The Reports should include a column for Account Type, Account Name, Debit, and Credits.

6.5 System Administration

6.5.1 Interface Requirements

INT.5.1 – The login page should have a button to request a new account.
INT.5.2 – The login page should have a button to request a change on an account
INT.5.3 – The button to request tickets should open a window for an account request.
INT.5.4 – The request window should have a text box for request summaries.
INT.5.5 – The request window should have a text box for the user’s email.
INT.5.6 – The request window should have a “Cancel” button that closes the request window.
INT.5.7 – The request window should have a “Submit” button that sends the request to an administrator.
INT.5.8 – For Administrators, the main task bar should have a “Users” button that takes the user to the Users page.
INT.5.9 – The Users page should have a “Create User” button that takes the user to the Create user page.
INT.5.10 – The Create User page should have a field for entering the username, password, and email address of the new user.
INT.5.11 – The Create User page should have a dropdown for selecting User Type for the new user.
INT.5.12 – The Create User page should have a “Cancel” button that takes the user back to the previous page.
INT.5.13 – The Create User page should have a “Create” button that adds the new user to the user database and takes the Administrator back to the Users page.
INT.5.14 – The Create User page should display an error if all fields are not filled, but the “Create” button has been clicked.

INT.5.15 – The Create User page should display an error if the email address field does not contain a valid email address.
INT.5.16 – When a user has been created, a message should display at the top of the Users page showing which user has been created.
INT.5.17 – The Users page should have an “Edit” button to the right of each user that opens the “Edit User” window.
INT.5.18 – The “Edit User” window should have a field for changing the username, password, email address, and user type for a user.
INT.5.19 – The “Edit User” window should have a “Cancel” button that takes the user back to the previous page.
INT.5.20 – The “Edit User” window should display an error if the email address field does not contain a valid email address.
INT.5.21 – When the “Edit” button is clicked within the “Edit User” window, it should submit the changes and take the administrator back to the Users page.
INT.5.22 – When a user has been edited, a message should display at the top of the Users page showing which user has been edited.
INT.5.23 – For Administrators, the main task bar should have a “Logs” button that takes the user to the “Logs” page.
INT.5.24 – The “Logs” page should have a drop down for starting and end date of logs.
INT.5.25 – The “Logs” page should have a drop down for number of entries per page
INT.5.26 – The “Logs” page should have a search bar.
INT.5.27 – The “Logs” page should have a “Previous Page” button to go back one page within the Logs.
INT.5.28 – The “Logs” page should have a “Next Page” button to go forward one page within the Logs.
INT.5.29 – The “Logs” page should display all events and changes within the system.

6.5.2 Operational Requirements

OPR.5.1 – The system should allow users to create different types of financial statements.
OPR.5.2 – The system should be able to create reports and allow for them to be downloaded.
OPR.5.3 – The system should have a “Reports” page where users can select a time period, end date, and report type for creating a report.
OPR.5.4 – The “Account Request” window should be able to send Administrators requests via email.
OPR.5.5 – The system should have the ability to check that a valid email has been entered.
OPR.5.6 – The system should have the ability to log all events that occur with before and after values of what was changed.
OPR.5.7 – The system should be able to limit how many events are displayed on each page of the “Logs” page.
OPR.5.8 – The system should be able to limit how many users are displayed on each page of the “Users” page.
OPR.5.9 – The system should be able to search/filter by any data field in the “Logs” page.
OPR.5.10 – The system should be able to search/filter by any data field in the “Users” page.
OPR.5.11 – The system should have the ability to check if a username already exists when creating a new user.

6.5.3 Data Requirements

DAT.5.1 – The “User Table” should include columns for username, password, email address, and user type.
DAT.5.2 – The “Log Table” should include columns for Log ID, Date/Time, Before Event, After Event, and User ID.
DAT.5.3 – When a change in the system is made, the user responsible should be used for the User ID within the “Logs” page.

7. Nonfunctional Requirements

7.1 Performance Requirements

NPR.1.1 - The system shall be able to respond to user requests within 5 seconds.
NPR.1.2 - The system database should handle requests in under 5 seconds.
NPR.1.3 - The system should support over 100 concurrent users at any given time.
NPR.1.4 - The system should display an error message within 1 second of when the error occurs.

7.2 Safety Requirements

SR.1.1- The system should not allow access to unverified users.
SR.1.2- The system should store all important information on a secure private server.
SR.1.3- The system should always require a login in a new session.
SR.1.4 - The system should allow for the immediate restriction of access of a user by an administrator.
SR.1.5 - The system should not allow the user to upload a file that is corrupted.
SR.1.6 - The system should only allow certain access to certain user types.
SR.1.7 - The system should logout after 10 minutes of inactivity,

7.3 Security Requirements

SCR.1.1 - The system should restrict functionality of the system based on user groups.

7.4 Software Quality Attributes

SQA.1.1 - The system should be available every hour of every day.
SQA.1.2 - The system should log any error encountered in use for maintenance and tracking purposes.
SQA.1.3 - The system should be compatible with all modern internet browsers.
SQA.1.4 - The software should provide correct results one hundred percent of the time.
SQA.1.5 - The software should be easily testable.

SQA.1.6 - The software should allow for updates to be installed remotely.
SQA.1.7 - The software should be designed so that any modification can be easily made
SQA.1.8 - The software should display an error message whenever an error occurs.

8. Other Requirements

OR.1.1- The system should abide by all accounting principles.

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Term	Definition
SRS	Software requirements specification- a description of a software system to be developed. It lays out functional and nonfunctional requirements, and may also include a set of use cases that describe user interactions that the software must provide.
Journal	A transaction to be stored within the system.
Account	Accounts within the Accounting system keep track of all transactions for an organization. They can be divided into categories such as Assets, Liabilities, Equity, Revenue, and Expenses.
Posting	A Manager has the ability to post or accept journal entries, in which the journal entries appear in the general ledger.
Rejecting	A Manager has the ability to reject journal entries, and are not added to the general ledger.
Report	Financial Reports are statements that possess information that can be interpreted by accountants.

Accountant	Standard user of the accounting system that has the ability to add journal entries, view journal entries, accounts, and reports.
Manager	A user that has all abilities of an accountant, but can also accept or reject journal entries.
PHP	Server side web language
HTTP	Headers required to accept, request, or send data
Apache	Server type for internet communications.
MySQL	Open sourced database management