ALTEAN
SYSTEM NAME: PROCION
PROCUREMENT SYSTEM

THE TEAM: TEAM 11

WERNER SCHUTTE
EMAIL: U21797189@TUKS.CO.ZA
CONTACT NUMBER: 0736179722
STUDENT NUMBER: U21797189

EMIL WONIGKEIT
EMAIL: U19022949@TUKS.CO.ZA
CONTACT NUMBER: 0820601750
STUDENT NUMBER: U19022949

LEON COMBRINCK
EMAIL: U21514951@TUKS.CO.ZA
CONTACT NUMBER: 0838430876
STUDENT NUMBER: U21514951

BUPE CHINDONGO
EMAIL: U21566641@TUKS.CO.ZA
CONTACT NUMBER: 0658793401
STUDENT NUMBER: U21566641

JASON VAN DER MERWE
EMAIL: U21502987@TUKS.CO.ZA
CONTACT NUMBER: 0793731393
STUDENT NUMBER: U21502987

FROM LEFT TO RIGHT: BUPE CHINDONGO || EMIL WONIGKEIT || JASON VD MERWE || LEON COMBRINCK || WERNER SCHUTTE

DESIGN & DEVELOPMENT: ITERATION 6

THIS DOCUMENT OUTLINES THE UPDATED FUNCTIONAL REQUIREMENTS LIST AS WELL AS NUMEROUS TECHNICAL DESIGNS FOR OUR CLIENT (MOYO BUSINESS ADVISORY) SUCH AS TECHNICAL USE CASE DIAGRAMS, TECHNICAL ERD, ACTIVITY, SEQUENCE AND DATAFLOW DIAGRAMS FOR VARIOUS ASPECTS OF THE SYSTEM SUCH AS REPORTS AND FURTHER FUNCTIONALITY IMPLEMENTATION. THE PURPOSE OF THIS ITERATION IS TO IFINALISE THE SYSTEM BY CONCLUDING REPORTING, FURHTER IMPLEMENTATION OF FUNCTIONS AS WELL AS TECHNICAL BUG FIXES.

MOYO
Driving Significance Together

Altean

Contents

TABLE OF TABLES

TABLE OF FIGURES

1.DOCUMENT INTRODUCTION

In iteration 6 we focus on the Design and the development of the Report sub-system of the Procion system that we are developing. Here we try and gather insight about the updated functional requirements, Logical Use Case Narratives, Logical context diagrams for those narratives, Screen designs for those narratives, Technical Use Case Narrative, Technical Context diagram for those narratives, Technical primitive level diagram for those narratives, Activity diagram for those narratives, Sequence diagram for those narratives, State diagram for those narratives(only complex narratives that has a change in state involved), and Demonstrate Functionality of the use cases, and the updated complexity matrix with the updated ERD to help us develop the basis for the final system This will bring us closer towards our end goal of the final solution of the Procion system.

2. UPDATED FUNCTIONAL REQUIRMENTS

2.1 INTRODUCTION

In this section a list of the second and last updated functional requirements will be listed for our proposed system Procion. The second and last updated functional requirements are grouped into the following sub-systems: The user sub-system, Admin sub-system, Procurement sub-system, Finance sub-system, Inventory sub-system, Vendor sub-system and a Report sub-system.

2.2 SUMMARY OF UPDATED FUNCTIONAL REQUIREMENTS

With regards to the to our functional requirements changes have been made to the following:

2.37 View Audit log

4.10 Export Budget Allocation

2.3 FUNCTIONAL REQUIREMENT LIST

SUBSYSTEM 1 – USER SUBSYSTEM

1.1 Login

1.2 Logout

1.3 Forgot password.

1.4 Update password

1.5 View profile.

1.6 Update profile

1.7 View user manual.

1.8 View Notification

SUBSYSTEM 2 – ADMIN SUBSYSTEM

2.1 Create employee.

2.2 View employee

2.3 Edit employee

2.4 Delete employee.

2.5 Create employee roles.

2.6 View employee roles

2.7 Update employee roles

2.8 Edit employee roles.

SUBSYSTEM 3 – PROCUREMENT SUBSYSTEM

SUBSYSTEM 4 – FINANCE SUBSYSTEM

SUBSYSTEM 5 – INVENTORY SUBSYSTEM

5.1 Create Consumable

5.2 View Consumables

5.3 Update Consumable

5.4 Delete Consumable

5.5 Create Consumable category.

5.6 View Consumable categories

5.7 Update Consumable category.

5.8 Delete Consumable category.

5.9 Export inventory details

5.10 Complete Stock Take

SUBSYSTEM 6 – VENDOR SUBSYSTEM

6.1 Create Vendor onboard Request

6.2 View Vendor onboard Request

6.3 Update Vendor Onboard Request

6.4 Delete Vendor Onboard Request

6.5 Approve onboard request

6.6 Create vendor

6.7 View vendor

6.8 Update Vendor

6.9 Delete vendor

6.10 Export Due Diligence Vendor Checklist

6.11 Generate Sole Supplier Review Notification

6.12 Generate BEE Certificate Expiry Notification

6.13 Update Approved onboard request

6.14 View Approved Or Pending onboard request.

SUBSYSTEM 7 – REPORT SUBSYSTEM

7.1 Generate Approved Vendor Report.

7.2 Generate BEE Spend Report.

7.3 Generate Vendor Spend Report

7.4 Generate Consumable Inventory Management Report.

7.5 Generate Business Unit Allocation Report

7.6 Generate Budget Variance Report

7.7 View Generate Report

2.4 FUNCTIONAL REQUIREMENTS DESCRIPTION

SUBSYSTEM 1 – USER SUBSYSTEM

Functional Requirement	Explanation
Requirement number:	1.1
Requirement name (use case name):	Login
Requirement short description:	The system should allow the user to log on to the system
Requirement detailed description and constraints:	**Description:** Each user will log on to the system with their username and password. The information is then validated using the system's database and will end either by providing user access to what their authority level allows or by displaying a error message indicating the entered details are incorrect. **Constraint(s):** • The user requires internet access
Business rules applicable to this requirement:	• All passwords must be encrypted and hashed. • The administrative must have added user into the system.
Revision date and Revision number:	03/17/2023 Version 2.0
Priority:	High

Table 1: 1.1 Login functional requirement

Functional Requirement	Explanation
Requirement number:	1.2
Requirement name (use case name):	Logout
Requirement short description:	The system should allow the user to log out of the system
Requirement detailed description and constraints:	**Description:** The user will request to log out of the system. The system will then log the user out and the use case concludes once the user is returned to the login page successfully. **Constraint(s):** None
Business rules applicable to this Requirement:	• The user requires internet access • User must be logged into the system. • The user must be able to log out from any page of the system.
Revision date and revision Number:	03/17/2023 Version 2.0
Priority:	High

Table 2: 1.2 Logout functional requirement

Functional Requirement	Explanation
Requirement number:	1.3
Requirement name (use case name):	Forgot password
Requirement short description:	The system should allow the user to reset their password if they forgot their password.
Requirement detailed description and constraints:	**Description:** The user will request that he forgot his password. The system will prompt the user to enter the following details: • Username/ email The system will use two factor authentication to send a code to the user's email address. The system will prompt the user to enter the security code. Once the security code is entered the user will be prompted to enter a new password. The system will validate that the new password is not the same as the previous password. The system will notify the user that his password has been changed successfully. **Constraint(s):** None
Business rules applicable to this Requirement:	• The user requires internet access • User must be logged out of the system. • The user must be registered on the system. • The new password cannot be the same as any previous passwords set by the user.
Revision date and revision Number:	03/17/2023 Version 2.0
Priority:	High

Table 3:1.3 Forgot password functional requirement

Functional Requirement	Explanation
Requirement number:	1.4
Requirement name (use case name):	Update password
Requirement short description:	The system should allow the user to reset their password
Requirement detailed description and constraints:	**Description:** The user should be able to request a new password should they want to change their password. The user will request to change password in which the system will request their previous password, new password. The previous password will be validated using the system database if the validation is correct the password will be replaced. The user will then be notified of the successful password update. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The user requires internet access • The user details must be verified before any changes are made. • The new password cannot be the same as any previous passwords set by the user.
Revision date and revision Number:	03/17/2023 Version 2.0
Priority:	High

Table 4: 1.4 Update password functional requirement

Functional Requirement	Explanation
Requirement number:	1.5
Requirement name (use case name):	View profile
Requirement short description:	The system should allow the user to view their own profile.
Requirement detailed description and constraints:	**Description:** The user requests to view their profile. The system will retrieve the user's details from the database and display the following details: • Username • Name • Surname • Telephone Number • Email • Branch • Department • Mandate Limit • Profile Picture The complete user profile will be displayed to the user. **Constraint(s):** None.
Business rules applicable to this Requirement:	• User must be logged in to the system • The user requires internet access.
Revision date and revision Number:	03/08/2023 Version 2.0
Priority:	High

Table 5:1.5 View profile functional requirement

Functional Requirement	Explanation
Requirement number:	1.6
Requirement name (use case name):	Edit profile
Requirement short description:	Users should be able to view their own profile and edit some parts of their details.
Requirement detailed description and constraints:	**Description:** The user requests to update their profile. The user will provide the specific details that they wish to update. The system will capture these details and validate the captured data. If the data validation returns a success the system will capture the newly updated details on the database. **Constraint(s):** Certain details cannot be updated without the proper access level.
Business rules applicable to this Requirement:	• User must be logged in to the system. • The user requires internet access.
Revision date and revision Number:	03/17/2023 Version 2.0
Priority:	High

Table 6: 1.6 Update profile functional requirement

Functional Requirement	Explanation
Requirement number:	1.7
Requirement name (use case name):	View user manual
Requirement short description:	The system should allow users to view the user manual.
Requirement detailed description and constraints:	**Description:** The user will request to view the user manual. The user will navigate to the user manual using the system. This will provide users with the user manual that will help explain certain parts of the system and how to work with them. **Constraint(s):** None
Business rules applicable to this Requirement:	• User must be logged in to the system.
Revision date and revision Number:	03/21/2023 Version 2.0
Priority:	High

Table 7:1.7 View user manual functional requirement

Functional Requirement	Explanation
Requirement number:	1.8
Requirement name (use case name):	View Notifications
Requirement short description:	The user of the system would like to view any notifications that specifically they might have.
Requirement detailed description and constraints:	**Description:** They system would allow the user to view all the notifications that are specifically meant for them, this could be found by the user at the top right of the navbar at the bell icon. When the user clicks on the icon the system will display a screen list of notifications for that user. Once the user clicks on one item of the list the system will navigate them to the appropriate location within the system where they can complete the task of that notification. Example The user clicks on a notification that has too with procurement status approval then the system will navigate that user to the procurement approval page for them to complete that task. **Constraint(s):** None
Business rules applicable to this Requirement:	• User must be logged in to the system. • User must have a notification before they can click on a specific notification.
Revision date and revision Number:	04/10/2023 Version 3.0
Priority:	High

Table 8: 1.9 View Notifications

SUBSYSTEM 2 – ADMIN SUBSYSTEM

Functional Requirement	Explanation
Requirement number:	2.1
Requirement name (use case name):	Create employee
Requirement short description:	The system should allow an admin with the right authority to create a new employee account.
Requirement detailed description and constraints:	**Description:** The admin with the right authority should be able to create a new employee account. Where they will need to provide all the information such as: • email address • phone number • name • surname The system will send the user an email containing a temporary login, allowing them to activate their account. The details are then validated by the system and added to the database. **Constraint(s):** A duplicate account cannot be made.
Business rules applicable to this Requirement:	• The admin must be logged into the system. • All needed details must be provided. • A strong password should be given. • The user requires internet access.
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 9:2.1 Create employee functional requirement.

Functional Requirement	Explanation
Requirement number:	2.2
Requirement name (use case name):	View employee
Requirement short description:	The system should allow an admin with the right authority to view and search all employee accounts on the system.
Requirement detailed description and constraints:	**Description:** The admin with the right authority will request to either search or view all employee accounts on the system. The admin will select the employee account they would like to view. The system will then display all the related details for that account. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin with the right authority must be logged into the system.
Revision date and revision Number:	03/18/2023 Version 2.0
Priority:	High

Table 10:2.2 View employee functional requirement

Functional Requirement	Explanation
Requirement number:	2.3
Requirement name (use case name):	Edit employee
Requirement short description:	The system should allow the admin with the right authority to update an employee's account detail.
Requirement detailed description and constraints:	**Description:** The admin with the right authority will request to view all employee accounts on the system. The admin will select the employee account they would like to update. The system will then display all the related details for that account where the admin can then edit any of the details. • email address • phone number • name • surname Once finished the system will validate the new details and either successfully update the details in the system database or it will display an error message regarding the details. The system then notifies the employee that their account has been updated. **Constraint(s):** The admin must "view" the employee account first before they will be able to update it.
Business rules applicable to this Requirement:	• The admin with the right authority must be logged into the system.
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 11:2.3 Update employee functional requirement

Functional Requirement	Explanation
Requirement number:	2.4
Requirement name (use case name):	Delete employee
Requirement short description:	The system should allow an admin with the right authority to delete an employee account.
Requirement detailed description and constraints:	**Description:** The admin wishes to delete an employee profile. The admin views the profile they wish to delete and requests deletion. The system will validate that the employee can be deleted. If the validation returns a success the system will remove the employee profile from the database. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The user with the right authority must be logged into the system. • The employee must not be with the company anymore.
Revision date and revision Number:	03/18/2023 Version 2.0
Priority:	High

Table 12:2.4 Delete employee functional requirement

Functional Requirement	Explanation
Requirement number:	2.5
Requirement name (use case name):	Create employee roles
Requirement short description:	The system should allow an admin with the right authority to add new employee roles
Requirement detailed description and constraints:	**Description:** Admin should be able to create new employee roles on the system. The admin will request to create new employee role. The system will request all relevant details • Name • Description The admin will provide the details and it will then be validated by the system. The system will notify the user if their role has been successfully added. **Constraint(s):** No duplication of roles will be allowed
Business rules applicable to this Requirement:	• Only users with the correct authority will be able to view employee roles.
Revision date and revision Number:	03/18/2023 Version 2.0
Priority:	High

Table 13:2.5 Create employee roles functional requirement

Functional Requirement	Explanation
Requirement number:	2.6
Requirement name (use case name):	View employee roles
Requirement short description:	The system should allow the admin to view all employee roles
Requirement detailed description and constraints:	**Description:** The admin will request to view all the employee roles saved on the system. The system will prompt the user to enter the role that they wish to search for. The user will provide the following details: • Name The system will search the system database for the record, retrieve the data and display it to the user. **Constraint(s):** None.
Business rules applicable to this Requirement:	• Only users with the correct authority will be able to view employee roles
Revision date and revision Number:	03/08/2023 Version 2.0
Priority:	High

Table 14:2.6 View employee roles functional requirement

Functional Requirement	Explanation
Requirement number:	2.7
Requirement name (use case name):	Edit employee roles
Requirement short description:	The system should allow the admin to update an employee's roles.
Requirement detailed description and constraints:	**Description:** The user will request to update an employee role. The user will select the employee role they would like to update. The system will then display the following details: • Name • Description The system will capture these details and perform validation in order to ensure that the details can be updated. Once the validation returns a success the system will update the relevant details in the systems database. The system will then notify the user that their role has been updated. **Constraint(s):** The admin must "view" the employee role first before they will be able to update it.
Business rules applicable to this Requirement:	• Only users with the correct authority will be able to view employee roles
Revision date and revision Number:	03/18/2023 Version 2.0
Priority:	High

Table 15:2.7 Update employee roles functional requirement

Functional Requirement	Explanation
Requirement number:	2.8
Requirement name (use case name):	Delete employee roles
Requirement short description:	The system should allow an admin to delete an employee role.
Requirement detailed description and constraints:	**Description:** The admin will request to delete an employee role. The system will prompt the user to confirm if they are sure that they want to continue with the deletion. The system will validate the deletion request and if it returns a success the employee role will be removed from the database. The user will then be notified by the system that their role has been removed. **Constraint(s):** None.
Business rules applicable to this Requirement:	• Only users with the correct authority will be able to delete employee roles
Revision date and revision Number:	03/18/2023 Version 2.0
Priority:	High

Table 16:2.8 Delete employee roles functional requirement

Functional Requirement	Explanation
Requirement number:	2.9
Requirement name (use case name):	Create Department
Requirement short description:	The system should allow the admin to create a Department section on the system.
Requirement detailed description and constraints:	**Description:** Admin should be able to create new department on to the system. The admin will request to create new department. The system will request the following details: • Name • Description The admin will provide the details and it will then be validated by the system. System will notify the admin that the department has successfully been created. **Constraint(s):** No duplicate departments allowed.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	03/20/2023 Version 2.0
Priority:	High

Table 17:2.9 Create department functional requirement

Functional Requirement	Explanation
Requirement number:	2.10
Requirement name (use case name):	View Department
Requirement short description:	The system should allow an admin to view all the departments.
Requirement detailed description and constraints:	**Description:** The admin will request to view a department. The system will prompt the user to enter the following details: • Name The system will search for the relevant record in the database, and it will retrieve the data associated with it. The system will display the department data to the user. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	03/20/2023 Version 2.0
Priority:	High

Table 18:2.10 View department functional requirement

Functional Requirement	Explanation
Requirement number:	2.11
Requirement name (use case name):	Edit Department
Requirement short description:	The system should allow an admin to update a department.
Requirement detailed description and constraints:	**Description:** The admin will request to update a department on the system. The user will select the department they would like to update. The system will then display all the related details for that account where the admin can then edit any of the details. • Name • Description The system will capture these details and perform validation in order to ensure that the details can be updated. Once the validation returns a success the system will update the relevant details in the systems database. The system will then notify the user that their department has been updated. **Constraint(s):** Admin must first view all departments before they can select and update the required department.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	03/20/2023 Version 3.0
Priority:	High

Table 19:2.11 Update department functional requirement

Functional Requirement	Explanation
Requirement number:	2.12
Requirement name (use case name):	Delete Department
Requirement short description:	The system should allow an admin to delete a department.
Requirement detailed description and constraints:	**Description:** Admin should be able to delete a selected department from the system. The system will confirm whether to continue with the deletion or not. All information for the department will then be deleted. The system notifies the admin that the department has been successfully removed. **Constraint(s):** Admin must first view all departments before they can select and delete the required department.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	03/20/2023 Version 3.0
Priority:	High

Table 20:2.12 Delete department functional requirement

Functional Requirement	Explanation
Requirement number:	2.13
Requirement name (use case name):	Create Branch
Requirement short description:	The system should allow an admin to create a branch on the system.
Requirement detailed description and constraints:	**Description:** Admin should be able to create new branch on to the system. The admin will request to create new branch. The system will request the following details: • Name • Street • City • Postal Code • Province The admin will capture the details and it will then be validated by the system. If the validation returns a success the system will create the new branch in the database. System will notify the admin that the branch has successfully been created. **Constraint(s):** No duplicate branches allowed
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	03/20/2023 Version 3.0
Priority:	High

Table 21:2.13 Create Branch functional requirement

Functional Requirement	Explanation
Requirement number:	2.14
Requirement name (use case name):	View Branch
Requirement short description:	The system should allow an admin to view all the branches on the system.
Requirement detailed description and constraints:	**Description:** The user will request to view a branch on the system. The system will prompt the user to enter the following details: • Name The user will select the branch they would like to view. The system will retrieve the relevant data from the system database. The system will then display all the related details for that branch. • Name • Street • City • Postal Code • Province **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	03/20/2023 Version 3.0
Priority:	High

Table 22:2.14 View Branch functional requirement

Functional Requirement	Explanation
Requirement number:	2.15
Requirement name (use case name):	Edit Branch
Requirement short description:	The system should allow an admin to update a branch
Requirement detailed description and constraints:	**Description:** The admin will request to update a branch on the system. The user will select the branch they would like to update. The system will then display all the relevant details of the branch. • Name • Street • City • Postal Code • Province The system will capture these details and perform validation in order to ensure that the details can be updated. Once the validation returns a success the system will update the relevant details in the systems database. The system will then notify the user that their branch has been updated. **Constraint(s):** Admin must first view all branches before they can select and update the required branches.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	03/20/2023 Version 3.0
Priority:	High

Table 23:2.15 Update Branch functional requirement

Functional Requirement	Explanation
Requirement number:	2.16
Requirement name (use case name):	Delete Branch
Requirement short description:	The system should allow an admin to delete a branch.
Requirement detailed description and constraints:	**Description:** Admin will request to delete a selected branch from the system. The system will confirm whether to continue with the deletion or not. All information for the branch will then be deleted. The system notifies that the admin that the branch has been successfully removed. **Constraint(s):** Admin must first view all branches before they can select and delete the required branch.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	03/20/2023 Version 3.0
Priority:	High

Table 24:2.16 Delete Branch functional requirement

Functional Requirement	Explanation
Requirement number:	2.17
Requirement name (use case name):	View delegation of authority
Requirement short description:	The admin should be able to view the list of all the delegation of authority that took place in the system.
Requirement detailed description and constraints:	**Description:** The admin clicks on the delegation of authority tab then the system will navigate to the list of all the delegation of authority that took place in the system. The amin will then be able to click on the view button then the system will display the details of that specific delegation of authority. **Constraint(s):** None
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	04/10/2023 Version 3.0
Priority:	High

Table 25: 2.17 View Delegation of Authority

Functional Requirement	Explanation
Requirement number:	2.18
Requirement name (use case name):	Assign delegation of authority
Requirement short description:	The admin should be able to setup a delegation of authority
Requirement detailed description and constraints:	**Description:** The admin will request to assign a delegation of authority. The admin will provide the details of when it will start and end, who will receive the authority and the completed Delegation of Authority document. **Constraint(s):** Must be approved before anything authority is given. The admin will not be able to complete this process until the completed Delegation of Authority document has been uploaded.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	24/07/2023 Version 3.0
Priority:	High

Table 26:2.18 Assigning delegation of authority functional requirement

Functional Requirement	Explanation
Requirement number:	2.19
Requirement name (use case name):	Edit delegation request
Requirement short description:	The admin should be able to setup a delegation of authority
Requirement detailed description and constraints:	**Description:** The admin with the right authority will request to view all delegations on the system. The admin will select the delegation they would like to update. The system will then display all the related details for that delegation where the admin can then edit any of the details. • Delegate user • From Date • To Date • Delegation Document Once finished the system will validate the new details and either successfully update the details in the system database or it will display an error message regarding the details. **Constraint(s):** The admin must “view” the delegation first before they will be able to update it.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	24/07/2023 Version 4.0
Priority:	High

Table 27:2.19 Edit delegation of authority functional requirement

Functional Requirement	Explanation
Requirement number:	2.20
Requirement name (use case name):	Revoke Access
Requirement short description:	The system will revoke a certain delegation of authority if the specified time period has passed
Requirement detailed description and constraints:	**Description:** The system request to revoke a certain delegation of authority. The system will access the delegation of authority data in the systems database to continuously check and compare the date values given. If the end date has passed the system will revoke the access and authority of the delegated user back to their original access level. The system will then notify both the requester and the receiver about the shift in delegation of authority. **Constraint(s):** Time period set must be passed.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	24/07/2023 Version 3.0
Priority:	High

Table 28: 2.20 Revoke Access functional requirement

Functional Requirement	Explanation
Requirement number:	2.21
Requirement name (use case name):	Create mandate limits
Requirement short description:	The system should allow the admin to set mandate limits for different departments for handling procurements without management approval
Requirement detailed description and constraints:	**Description:** The admin will request to create a new mandate limit for a department. The system will request the user to select a role and provide the mandated limit. The required details are provided • Amount • Department The system will validate that the roles have no other set mandate limits and then update it as needed. The system then displays a success message for the user. **Constraint(s):** Only departments that does not have a non-default mandate limit should be displayed.
Business rules applicable to this requirement:	• User must be registered on the system. • The user must have authority to access the create mandate limits. • The different departments must be stored in the database
Revision date and Revision number:	03/19/2023 Version 2.0
Priority:	High

Table 29:2.21 Create mandate limits functional requirement

Functional Requirement	Explanation
Requirement number:	2.22
Requirement name (use case name):	View Mandate limits
Requirement short description:	The system should allow the admin to view the different employees with a mandate limited.
Requirement detailed description and constraints:	**Description:** The admin will request to view all the mandated limits created for employees. The system will retrieve all required details from the relevant tables and display all the mandate limits. **Constraint(s):** None.
Business rules applicable to this requirement:	• User must be registered on the system. • The user must have the correct authority to view mandate limits
Revision date and Revision number:	05/18/2023 Version 3.0
Priority:	High

Table 30:2.22 View mandate limits functional requirement

Functional Requirement	Explanation
Requirement number:	2.23
Requirement name (use case name):	Update Mandate limits
Requirement short description:	The system should allow the admin to update the mandated limits in any employee.
Requirement detailed description and constraints:	**Description:** The admin will request to update a mandate limit for an employee. The system prompt the user to select the mandate limit that they wish to update. The user will select the limit and provide the following details: • Amount • Date The system will validate the details that need to be updated. If the validation returns a success the details will be updated on the system. The system finally displays a success message for the user. **Constraint(s):** None.
Business rules applicable to this requirement:	• User must be registered on the system. • The user must have authority to access the create mandate limits.
Revision date and Revision number:	05/18/2023 Version 3.0
Priority:	High

Table 31:2.23 Update mandate limits functional requirement

Functional Requirement	Explanation
Requirement number:	2.24
Requirement name (use case name):	Delete Mandate limits
Requirement short description:	The system should allow the admin to delete any mandated limits of an employee or group.
Requirement detailed description and constraints:	**Description:** The admin will request to delete a mandate limit for an employee. The system will request the selection of an employee for the deletion of the mandate limit. The system will validate the request in order to ensure that the data can be deleted. Once the validation returns a success the data will be removed from the system. The system finally displays a success message for the user. **Constraint(s):** None.
Business rules applicable to this requirement:	• The administrative must have added user into the system
Revision date and Revision number:	05/18/2023 Version 3.0

Priority:	High

Table 32:2.24 Delete mandate limits functional requirement

Functional Requirement	Explanation
Requirement number:	2.25
Requirement name (use case name):	Backup system data
Requirement short description:	The admin should be able to make a backup of all the data on the system.
Requirement detailed description and constraints:	**Description:** The admin should be following a scheduled time for making a backup of the system data. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	03/19/2023 Version 2.0
Priority:	High

Table 33:2.25 Backup system data functional requirement

Functional Requirement	Explanation
Requirement number:	2.26
Requirement name (use case name):	Restore data to system
Requirement short description:	The admin should be able to restore all data to the system using a backup.
Requirement detailed description and constraints:	**Description:** The admin request to restore system to a previous version. The system will then restore all data to the point that it was last backed up. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	03/19/2023 Version 2.0
Priority:	High

Table 34:2.26 Restore data to system functional requirement

Functional Requirement	Explanation
Requirement number:	2.27
Requirement name (use case name):	Add Help
Requirement short description:	The admin should be able to add a help section.
Requirement detailed description and constraints:	**Description:** The admin will request to create a help section. The system will prompt the user to enter the following: • Name • Description • Video • User Manual • Category The system will capture these details and create the help function on the system. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin must be logged into the system. • Only the admin can make use of the add help function
Revision date and revision Number:	07/24/2023 Version 3.0
Priority:	Medium

Table 35:2.27 Add help functional requirement

Functional Requirement	Explanation
Requirement number:	2.28
Requirement name (use case name):	Update Help
Requirement short description:	The admin should be able to update a help section.
Requirement detailed description and constraints:	**Description:** The admin will request to update a help section. The system will prompt the user to update any of the following: • Name • Description • Video • User Manual • Category The system will capture these details and update the help function on the system’s database. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin must be logged into the system. • Only the admin can make use of update help.
Revision date and revision Number:	07/24/2023 Version 3.0
Priority:	Medium

Table 36:2.28 Update help functional requirement

Functional Requirement	Explanation
Requirement number:	2.29
Requirement name (use case name):	Delete Help
Requirement short description:	The admin should be able to Delete a help section.
Requirement detailed description and constraints:	**Description:** The admin will request to delete a help section. The system will prompt the user to confirm that they want to delete the specified help section. The system will validate the request and once it retrieves a success the help section along with the following details will be deleted and removed from the system: • Name • Description • Video • User Manual • Category **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin must be logged into the system. • Only the admin can make use of Delete help. • The admin must confirm the deletion of the help.
Revision date and revision Number:	07/24/2023 Version 3.0
Priority:	Medium

Table 37:2.29 Delete help functional requirement

Functional Requirement	Explanation
Requirement number:	2.30
Requirement name (use case name):	View all help
Requirement short description:	The admin should be able to view all the help sections.
Requirement detailed description and constraints:	**Description:** The admin will request to view a help section. The system will prompt the user to select a help section that they would like to view. The system will retrieve the help section from the system's database and display it to the user along with the following details: • Name • Description • Video • User Manual • Category **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin must be logged into the system. • Only the admin can make use of View all help.
Revision date and revision Number:	07/24/2023 Version 3.0
Priority:	Medium

Table 38:2.30 View all help functional requirement

Functional Requirement	Explanation
Requirement number:	2.31
Requirement name (use case name):	Create admin
Requirement short description:	The system should allow an admin with the right authority to create a new admin account.
Requirement detailed description and constraints:	**Description:** The admin with the right authority should be able to create a new admin account. Where they will need to provide all the information such as: • email address • phone number • name • surname The details are then validated by the system and added to the database. **Constraint(s):** A duplicate account cannot be made.
Business rules applicable to this Requirement:	• The admin must be logged into the system. • All needed details must be provided. • A strong password should be given. • The user requires internet access.
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 39: 2.31 Create Admin

Functional Requirement	Explanation
Requirement number:	2.32
Requirement name (use case name):	View admin
Requirement short description:	The system should allow an admin with the right authority to view and search all admin accounts on the system.
Requirement detailed description and constraints:	**Description:** The admin with the right authority will request to either search or view all admin accounts on the system. The admin will select the admin account they would like to view. The system will then display all the related details for that account. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin with the right authority must be logged into the system.
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 40: 2.31 View Admin

Functional Requirement	Explanation
Requirement number:	2.33
Requirement name (use case name):	Edit admin
Requirement short description:	The system should allow the admin with the right authority to update an admin's account detail.
Requirement detailed description and constraints:	**Description:** The admin with the right authority will request to view all admin accounts on the system. The admin will select the admin account they would like to update. The system will then display all the related details for that account where the admin can then edit any of the details. • email address • phone number • name • surname Once finished the system will validate the new details and either successfully update the details in the system database or it will display an error message regarding the details. The system then notifies the admin that their account has been updated. **Constraint(s):** The admin must "view" the admin account first before they will be able to update it.
Business rules applicable to this Requirement:	• The admin with the right authority must be logged into the system.
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 41: 2.33 Update Admin

Functional Requirement	Explanation
Requirement number:	2.34
Requirement name (use case name):	Delete admin
Requirement short description:	The system should allow an admin with the right authority to delete an admin account.
Requirement detailed description and constraints:	**Description:** The admin wishes to delete an admin profile. The admin views the profile they wish to delete and requests deletion. The system will validate that the admin can be deleted. If the validation returns a success the system will remove the admin profile from the database. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The user with the right authority must be logged into the system. • The admin must not be with the company anymore.
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 42: 2.34 Delete Admin

Functional Requirement	Explanation
Requirement number:	2.35
Requirement name (use case name):	Delete Delegation of Authority
Requirement short description:	The system should allow an admin with the right authority to delete a delegation request.
Requirement detailed description and constraints:	**Description:** The admin wishes to delete a delegation request. The admin views the delegation they wish to delete and requests deletion. The system will validate that the delegation can be deleted. If the validation returns a success the system will remove the delegation request from the database. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The user with the right authority must be logged into the system.
Revision date and revision Number:	24/07/2023 Version 1.0
Priority:	High

Table 43: 2.35 Delete Delegation of Authority

Functional Requirement	Explanation
Requirement number:	2.36
Requirement name (use case name):	Activate Delegation of Authority
Requirement short description:	The system will activate a certain delegation of authority if the specified time period begins
Requirement detailed description and constraints:	**Description:** The system request to activate a certain delegation of authority. The system will access the delegation of authority data in the systems database to continuously check and compare the date values given. If the start date has been reached the system will activate the access and authority of the delegated user back to the additional temporary access level. **Constraint(s):** Time period set must begin.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	24/07/2023 Version 1.0
Priority:	High

Table 44: 2.36 Activate Delegation of Authority

Functional Requirement	Explanation
Requirement number:	2.37
Requirement name (use case name):	View Audit Log
Requirement short description:	The system should allow an admin to view Audit Log.
Requirement detailed description and constraints:	**Description:** The use case describes the event where a Admin would like to view Audit log on the system. The user will then request the system to view the Audit Log on the system. The system will then load all Audit log that are currently on the system and display them with the following information: • Time Completed • User • Action Taken The system promts the Admin to enter sort by criteria.The Admin will then request the system to be able to Sort by for the audit log. The system sorts the audit log in order of the options that the admin chose and load the sorted audit log to the Admin. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The admin must be logged into the system.
Revision date and revision Number:	08/08/2023 Version 1.0
Priority:	High

Table 45:2.37 View Audit Log functional requirement

SUBSYSTEM 3 – PROCUREMENT SUBSYSTEM

Functional Requirement	Explanation
Requirement number:	3.1
Requirement name (use case name):	Create procurement request
Requirement short description:	Allows user to make a new request for procurement.
Requirement detailed description and constraints:	**Description:** The user can request to make a new request for procurement. The system will request the following details: • Vendor name (It can be an approved vendor selected from a dropdown list, or a other vendor from which the name should be specified) • Quote options The system will validate the request. In the case of a "other" vendor that was selected, the system will validate the request to check if that vendor has been used more than 2 times. Once the request returns a success the request will be stored in the database and given the status of "Submitted" is assigned. **Constraint(s):** None.
Business rules applicable to this Requirement:	• User must be logged into the system.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 46: 3.1 Create Procurement request.

Functional Requirement	Explanation
Requirement number:	3.2
Requirement name (use case name):	View procurement request
Requirement short description:	User will be able to view a request for procurement which has been made.
Requirement detailed description and constraints:	**Description:** The user requests to view a procurement request. When the view button is pressed, the procurement request should be displayed, which will detail the • Vendor name • Quote options The system will search the systems database and retrieve the relevant procurement request records and display it to the user where the user will be able to view the Procurement request details that have been stored on the system. **Constraint(s):** At least one procurement request must be present on the system.
Business rules applicable to this Requirement:	• User must be logged into the system.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 47: 3.2 View procurement request.

Functional Requirement	Explanation
Requirement number:	3.3
Requirement name (use case name):	Update procurement request
Requirement short description:	The user should be able to update a request for procurement that has not yet been approved or rejected.
Requirement detailed description and constraints:	**Description:** The user will request to update a procurement request. The user will select the request they would like to update. The system will then display all the related details for that account where the user can then edit any of the details. • Vendor name • Quote options Once finished the system will validate the new details and either successfully updated the details in the system database or it will display an error message regarding the details. **Constraint(s):** The request for procurement should be temporary disabled from being rejected or approved.
Business rules applicable to this Requirement:	• User must be logged into the system.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 48: 3.3 Update procurement request

Functional Requirement	Explanation
Requirement number:	3.4
Requirement name (use case name):	Delete procurement request
Requirement short description:	The user should be able to delete a request for procurement that has not yet been approved or rejected.
Requirement detailed description and constraints:	**Description:** The user will request to delete a procurement request. The user will select the request they would like to delete. The system will prompt the user to confirm the deletion of the request. The request of deletion will be validated and if the validation returns a success, all information for the request will then be deleted. The system notifies that the request has been successfully removed. **Constraint(s):** None.
Business rules applicable to this Requirement:	• User must be logged into the system
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 49: 3.4 Delete procurement request.

Functional Requirement	Explanation
Requirement number:	3.5
Requirement name (use case name):	Approve Procurement Request
Requirement short description:	The system should allow procurement request created to be approved or rejected.
Requirement detailed description and constraints:	**Description:** The Budget owner requests to approve or reject a request. The system will prompt the user to select the request they would like to approve or reject. The user will select the request and subsequently either approve or reject it. Once the request has been accepted or rejected the relevant buyer will be notified that the approval has been made **Constraint(s):** A procurement request must exist, and at least 1 quotes must be uploaded for this procurement request.
Business rules applicable to this Requirement:	• Budget Owner should be logged into the system
Revision date and revision Number:	07/26/2023 Version 3.0
Priority:	High

Table 50: 3.5 Approve Procurement Request

Functional Requirement	Explanation
Requirement number:	3.6
Requirement name (use case name):	Place procurement details
Requirement short description:	The system should allow the user to place order details into the system.
Requirement detailed description and constraints:	**Description:** The user request to place order details. The system will request the following details: • Buyer Name • Buyer Email • Item Type • Consumable item/Asset Name • Quantity/Description • Account Code • Payment Type • Deposit Amount • Deposit Due Date • Full payment due date • Receipt Document • Total amount • Proof of payment • Comment The provided details are then validated by the system and then saved into the database. The system then displays a success message for the user. **Constraint(s):** The procurement request must have been approved.
Business rules applicable to this Requirement:	• User must be logged into the system.
Revision date and revision Number:	07/23/2023 Version 3.0
Priority:	High

Table 51: 3.6 Place procurement request.

Functional Requirement	Explanation
Requirement number:	3.7
Requirement name (use case name):	Approve Flagged Procurement Details
Requirement short description:	The system will flag a procurement details when its cost is over a mandated limit and allow the *budget owner* to review request to either reject or accept request.
Requirement detailed description and constraints:	**Description:** The system flags a procurement for going over mandated limit and notify the budget owner of the procurement request. The budget owner will request to review request. The system will display the procurement request and request that the user provide either accept or reject input. The input is provided by the user and either a reject or approved notification is sent to the requestor. **Constraint(s):** The procurement request must be larger than the requestor mandate limit.
Business rules applicable to this Requirement:	• The user must be logged in the system.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 52: 3.7 Management mandate approval

Functional Requirement	Explanation
Requirement number:	3.8
Requirement name (use case name):	Receive Procurement item
Requirement short description:	The user should be allowed to update stock items for when receiving procurement items.
Requirement detailed description and constraints:	**Description:** The user will request to add the stock received on the system. The system will retrieve the data from the procurement request and prompt the user to enter the quantity received. The user will enter the following details relative to the procurement request item: • Amount The system will capture the relevant details and the amount will be saved on the system's database. An on-screen notification will be displayed to the user to inform them of the successful stock capture. **Constraint(s):** None
Business rules applicable to this Requirement:	• User must be logged into the system. • The request must be approved.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 53: 3.8 Receive Procurement item.

Functional Requirement	Explanation
Requirement number:	3.9
Requirement name (use case name):	Upload tax Invoice
Requirement short description:	A user should be allowed to upload a tax invoice.
Requirement detailed description and constraints:	**Description:** The user would like to upload a tax invoice for a procurement request. The system will request the user to provide the required file. The user will select the file they want to upload from their device onto the system. The system will then save the files with into the database. The system will display a success message if the file has been successfully added to the database. **Constraint(s):** The system will not allow the user to continue with the procurement process until the invoice has been uploaded to the system.
Business rules applicable to this Requirement:	• User must be logged into the system.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 54: 3.9 Upload tax Invoice

Functional Requirement	Explanation
Requirement number:	3.10
Requirement name (use case name):	Budget owner Requisition approval
Requirement short description:	The system should allow the budget owner to review the requisition request and approve or reject it.
Requirement detailed description and constraints:	**Description:** The user request to review the requisition request. The system will display all the details from the procurement request. The budget owner will verify the uploaded invoice, the total amount due, and the selected budget line. After verification the budget owner will either accept or reject the procurement request and notify the appropriate parties. If the request is approved the system will notify the Finance department to continue with payment. If the request is rejected the buyer will be notified to make the requested changes. **Constraint(s):** None

Business rules applicable to this Requirement:	• User must be logged into the system. • The request must be approved.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 55: 3.10 Budget owner Requisition approval

Functional Requirement	Explanation
Requirement number:	3.11
Requirement name (use case name):	Upload receipt
Requirement short description:	A user should be allowed to upload a receipt.
Requirement detailed description and constraints:	**Description:** The user would like to upload a receipt for a procurement request. The system will request the user to provide the required file. The user will select the file they want to upload from their device onto the system. The system will then save the file into the database. The system will display a success message if the file has been successfully added to the database. **Constraint(s):** None
Business rules applicable to this Requirement:	• User must be logged into the system.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 56: 3.11 Upload receipt

Functional Requirement	Explanation
Requirement number:	3.12
Requirement name (use case name):	Upload credit card Payment
Requirement short description:	A user should be allowed to upload a credit card payment.
Requirement detailed description and constraints:	**Description:** The user would like to upload a credit card payment for a procurement request. The system will request the user to provide the required file. The user will select the file they want to upload from their device onto the system. The system will then save the file into the database. The system will display a success message if the file has been successfully added to the database. **Constraint(s):** None
Business rules applicable to this Requirement:	• The request must have a status of approved
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 57: 3.12 Upload credit card Payment

SUBSYSTEM 4 – FINANCE SUBSYSTEM

Functional Requirement	Explanation
Requirement number:	4.1
Requirement name (use case name):	Finalize procurement requests
Requirement short description:	The user should be able to search or view all invoices on the system.
Requirement detailed description and constraints:	**Description:** The user would like to finalize a procurement request. The user would select the procurement request that is awaiting payment. The request would be reviewed and paid outside of the system. The user will then upload the proof of payment (optional) and finalize the procurement request. The system will then automatically deduct the amount of the invoice from the requesters procurement items specific budget line. A notification would then be sent to the requester that their procurement request has been finalized. **Constraint(s):** Procurement request must exist beforehand
Business rules applicable to this Requirement:	• User must be logged into the system. • A procurement item request must be approved. • Invoice must be uploaded to the system
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 58: 4.1 Finalize procurement requests.

Functional Requirement	Explanation
Requirement number:	4.2
Requirement name (use case name):	Create budget allocation
Requirement short description:	The user should be able to create a new budget allocation for a department.
Requirement detailed description and constraints:	**Description:** The user would like to add a budget allocation to the system. The system will gather the following details. • Date • Total • Year for budget • Department name The system will capture the data on the system and save it in the systems database. The system will notify the relevant budget owner that the budget has been created. **Constraint(s):** None
Business rules applicable to this Requirement:	• User must be logged into the system. • The user must have the right authority to create a budget allocation.
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 59: 4.3 Create Budget allocation

Functional Requirement	Explanation
Requirement number:	4.3
Requirement name (use case name):	View budget allocation
Requirement short description:	The user should be able to view all budget allocation for the different departments.
Requirement detailed description and constraints:	**Description:** The user requests to view all budget allocation for the different departments. The system will retrieve the following details: • Date • Total • Year for budget • Department name This information will be displayed to the user. **Constraint(s):** None
Business rules applicable to this Requirement:	• User must be logged into the system. • The user must have the right authority to access budget allocation.
Revision date and revision Number:	05/18/2023 Version 2.0
Priority:	High

Table 60: 4.4 View budget allocation

Functional Requirement	Explanation
Requirement number:	4.4
Requirement name (use case name):	Update budget allocation
Requirement short description:	The user should be able update the budget allocation
Requirement detailed description and constraints:	**Description:** The user will request to view all budget allocations on the system. The user will select the request they would like to update. The user has the option to update the relevant details on the system which includes: • Adding and removing of line items • Changing figures The relevant changes will be stored on the system's database. **Constraint(s):** There must be a budget allocation created on the system
Business rules applicable to this Requirement:	• User must be logged into the system. • The user must have the right authority to access budget allocation.
Revision date and revision Number:	05/18/2023 Version 2.0
Priority:	High

Table 61: 4.5 Update budget allocation

Functional Requirement	Explanation
Requirement number:	4.5
Requirement name (use case name):	Delete budget allocation
Requirement short description:	The system should allow a user to delete a budget allocation.
Requirement detailed description and constraints:	**Description:** The user would like to delete a budget allocation. The user would view the budget that they would like to remove. Once the user has chosen to remove the budget allocation, the system will validate the request. If the validation returns a success the user will be notified that the budget has been successfully deleted and the data will be removed from the system. **Constraint(s):** There must be a budget created on the system.
Business rules applicable to this Requirement:	• The user must be logged into the system. • The user must have the right authority to access budget allocation.
Revision date and revision Number:	05/18/2023 Version 2.0
Priority:	High

Table 62: 4.6 Delete budget allocation.

Functional Requirement	Explanation
Requirement number:	4.6
Requirement name (use case name):	Create budget category
Requirement short description:	The user should be able to create a new budget category.
Requirement detailed description and constraints:	**Description:** The user would like to add a budget category to the system. The system will prompt the user to enter the following details for the budget category: • Name • Description **Constraint(s):** None
Business rules applicable to this Requirement:	• The user must be logged into the system. • The user must have the right authority to access budget category.
Revision date and revision Number:	05/18/2023 Version 1.0
Priority:	High

Table 63: 4.7 Create Budget Category

Functional Requirement	Explanation
Requirement number:	4.7
Requirement name (use case name):	View budget category
Requirement short description:	The user should be able to view all budget categories on the system.
Requirement detailed description and constraints:	**Description:** The user requests to view all budget categories The system will retrieve the following details: • Name • Description This information will be displayed to the user. **Constraint(s):** None
Business rules applicable to this Requirement:	• User must be logged into the system. • The user must have the right authority to access budget category.
Revision date and revision Number:	05/18/2023 Version 1.0
Priority:	High

Table 64: 4.8 View Budget Category

Functional Requirement	Explanation
Requirement number:	4.8
Requirement name (use case name):	Update budget category
Requirement short description:	The user should be able update the budget category
Requirement detailed description and constraints:	**Description:** The user will request to view all budget categories on the system. The user will select the category they would like to update. The user has the option to update the relevant details on the system which includes: • Name • Description The relevant changes will be stored on the system's database. **Constraint(s):** There must be a budget category created on the system
Business rules applicable to this Requirement:	• User must be logged into the system. • The user must have the right authority to access budget category.
Revision date and revision Number:	05/18/2023 Version 1.0
Priority:	High

Table 65: 4.9 Update Budget Category

Functional Requirement	Explanation
Requirement number:	4.9
Requirement name (use case name):	Delete budget category
Requirement short description:	The system should allow a user to delete a budget category.
Requirement detailed description and constraints:	**Description:** The user would like to delete a budget category. The user would view the budget category that they would like to remove. Once the user has chosen to remove the budget category, the system will validate the request. If the validation returns a success the user will be notified that the budget category has been successfully deleted and the data will be removed from the system. **Constraint(s):** There must be a budget category created on the system.
Business rules applicable to this Requirement:	• The user must be logged into the system. • The user must have the right authority to access budget category.
Revision date and revision Number:	05/18/2023 Version 1.0
Priority:	High

Table 66: 4.10 Delete Budget Category

Functional Requirement	Explanation
Requirement number:	4.10
Requirement name (use case name):	Export Budget Allocation
Requirement short description:	The system will allow the user to export the details of the Budget allocation to a Excel file
Requirement detailed description and constraints:	**Description:** The use case describes the event where the user would like to export the details contained in the Budget allocation system to a .xlsx file.. The user will export the Budget Allocation list and the .xlsx will be generated as response and downloaded to the users device. **Constraint(s):** none
Business rules applicable to this Requirement:	• The user with the right authority must be logged into the system.
Revision date and revision Number:	03/08/2023 Version 1.0
Priority:	High

Table 67: 4.10 Export Budget Allocation

SUBSYSTEM 5 – INVENTORY SUBSYSTEM

Functional Requirement	Explanation
Requirement number:	5.1
Requirement name (use case name):	Create Consumable
Requirement short description:	The system should allow a procurement consumable item to be created
Requirement detailed description and constraints:	**Description:** The use case describes the event where the user will want to create a new consumable. The system will prompt the user to enter the product details. Including: • Item name • Item description • Item Category Once this is complete and the item is saved the item will be added to the system's database. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The user must be logged in to the system.
Revision date and revision Number:	05/18/2023 18/08/2023 Version 2.0
Priority:	High

Table 68: 5.1 Create inventory item

Functional Requirement	Explanation
Requirement number:	5.2
Requirement name (use case name):	View Consumables
Requirement short description:	The system should allow the user to view consumables on the system
Requirement detailed description and constraints:	**Description:** The use case describes the event where a user would want to search for a a consumable. The item can be searched for with the following details: • Item name • ItemID The system will search through all of the records in the database and retrieve the correct item and display it to the user based on their search criteria. **Constraint(s):** There needs to be an inventory item created on the system.
Business rules applicable to this Requirement:	• The user must be logged in to the system
Revision date and revision Number:	18/08/2023 Version 2.0
Priority:	High

Table 69: 5.2 View inventory item

Functional Requirement	Explanation
Requirement number:	5.3
Requirement name (use case name):	Update Consumable
Requirement short description:	The system should allow the user to update a consumable
Requirement detailed description and constraints:	**Description:** The use case describes the event where a user would want to update certain consumable The user will view the item that they wish to update. They will enter the relevant details that they wish to change and once updated the data will be validated on the system. If the validation returns a success, the details will be saved on the systems database. **Constraint(s):** There needs to be an inventory item on the system.
Business rules applicable to this Requirement:	• The user must be logged into the system.
Revision date and revision Number:	18/08/2023 Version 2.0
Priority:	High

Table 70: 53 Update inventory item

Functional Requirement	Explanation
Requirement number:	5.4
Requirement name (use case name):	Delete Consumable
Requirement short description:	The system should allow the user to delete a consumable
Requirement detailed description and constraints:	**Description:** This use case describes the event where the user requests to delete a consumable. The user will select the inventory item they want to delete. The system will ask the user for confirmation of deletion. The system will validate the request of deletion. If the validation returns a success the system will remove the procurement inventory item from the system. **Constraint(s):** An inventory item needs to be on the system.
Business rules applicable to this Requirement:	• User must be logged in to the system.
Revision date and revision Number:	18/08/2023 Version 2.0
Priority:	High

Table 71: 5.4 Delete inventory item

Functional Requirement	Explanation
Requirement number:	5.5
Requirement name (use case name):	Create Consumable Category
Requirement short description:	The system should allow a category to be created
Requirement detailed description and constraints:	**Description:** The use case describes the event where a user wants to create a category on the system. The system will prompt the user to enter the category details such as: • Category name • Category Description Once this is complete and the category is saved the category will be added to the system's database. **Constraint(s):** None.
Business rules applicable to this Requirement:	• The user must be logged in to the system.
Revision date and revision Number:	18/08/2023 Version 2.0
Priority:	High

Table 72: 5.5 Create item category

Functional Requirement	Explanation
Requirement number:	5.6
Requirement name (use case name):	View Consumable categories
Requirement short description:	The system should allow the user to view a consumable category
Requirement detailed description and constraints:	**Description:** The use case describes the event where a user would want to search for a specific category. The item can be searched for with the following details: • category name • Category ID The system will search through all of the records in the database and retrieve the correct category and display it to the user based on their search criteria. **Constraint(s):** There needs to be an inventory item category created on the system.
Business rules applicable to this Requirement:	• User must be logged in to the system.
Revision date and revision Number:	18/08/2023 Version 2.0
Priority:	High

Table 73: 5.6 View item category

Functional Requirement	Explanation
Requirement number:	5.7
Requirement name (use case name):	Update Consumable Category
Requirement short description:	The system should allow the user to update a consumable category
Requirement detailed description and constraints:	**Description:** The use case describes the event where a user would want to update a certain category. The user will view the category that they wish to update. They will enter the relevant details that they wish to change and once updated the data will be validated on the system. If the validation returns a success, the details will be saved on the systems database. **Constraint(s):** There needs to be an inventory item category on the system.
Business rules applicable to this Requirement:	• User must be logged in to the system.
Revision date and revision Number:	18/08/2023 Version 2.0
Priority:	High

Table 74: 5.7 Update Item Category

Functional Requirement	Explanation
Requirement number:	5.8
Requirement name (use case name):	Delete Consumable Category
Requirement short description:	The system should allow the user to delete a consumable Category
Requirement detailed description and constraints:	**Description:** This use case describes the event where the user requests to delete an consumable category. The user will select the category they want to delete. The system will ask the user for confirmation of deletion. The system will validate the request of deletion. If the validation returns a success the system will remove the procurement inventory category from the system. . **Constraint(s):** An inventory item category needs to be on the system.
Business rules applicable to this Requirement:	• User must be logged in to the system.
Revision date and revision Number:	18/08/2023 Version 2.0
Priority:	High

Table 75: 5.8 Delete Item Category

Functional Requirement	Explanation
Requirement number:	5.9
Requirement name (use case name):	Export inventory details
Requirement short description:	The system will allow the user to export the details of the procurement item inventory to a CSV file
Requirement detailed description and constraints:	**Description:** The use case describes the event where the user would like to export the details contained in the procurement item inventory system (Item name, stock on hand) to a .csv file in the structure supported by Xero. The user will export the inventory list and the .csv will be generated as response. **Constraint(s):** There must be inventory loaded on the system.
Business rules applicable to this Requirement:	• The user with the right authority must be logged into the system.
Revision date and revision Number:	03/08/2023 Version 1.0
Priority:	High

Table 76: 5.9 Export inventory details

Table 77: 5.10 Generate reorder stock notification.

Functional Requirement	Explanation
Requirement number:	5.10
Requirement name (use case name):	Complete stock take
Requirement short description:	The system should allow the user to complete a stock take.
Requirement detailed description and constraints:	**Description:** The use case describes the event where the user completes a stock take, and the stock levels of different consumables are entered and updated. The user will select the items for which they intend to perform a stock take, and the system will prompt them to enter the amount of stock which exists for these items, after which this data will be captured and saved to the system database. **Constraint(s):** There must be inventory loaded on the system.
Business rules applicable to this Requirement:	• Only the appropriate users will be able to perform a stock take
Revision date and revision Number:	03/08/2023 Version 1.0
Priority:	High

Table 78: 5.10 Complete stock take

SUBSYSTEM 6 – VENDOR SUBSYSTEM

Functional Requirement	Explanation
Requirement number:	6.1
Requirement name (use case name):	Create Vendor onboard Request
Requirement short description:	The system should allow the user to create a vendor onboard request.
Requirement detailed description and constraints:	**Description:** The system should allow the user to create a vendor onboard request. The system should allow the user to select between a normal vendor or a sole supplier. In the case of a normal vendor: The user would have had to collect 3 quotes from 3 different companies. The system will prompt the user to upload the documents from their computer's file system. The user will be prompted to enter the company name and email related to each quote. The user will then confirm the request which will notify the Procurement department for approval. In the case of a sole supplier: The user will provide the company name and reason for this vendor to be approved with the option of providing a quote. The user will then confirm the request which will notify the Managing director for approval. **Constraint(s):** A number of 3 Quotes and company names must be added to the request for the user to be able to proceed with the process if the sole supplier option is NOT selected.
Business rules applicable to this Requirement:	• The user must be logged into the system. • The business requires a minimum of 3 quotes for a vendor to be onboarded in the case of a normal vendor. • Sole suppliers do not need to have quotes as supporting documents but require the managing director's approval
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 79: 6.1 Create Vendor onboard Request.

Functional Requirement	Explanation
Requirement number:	6.2
Requirement name (use case name):	View Vendor onboard Request
Requirement short description:	The system should allow the user to view the vendor onboard request
Requirement detailed description and constraints:	**Description:** Under the vendor section, the requester (user) should be able view their current vendor onboard requests. When the view button is pressed, the request should be displayed along with the uploaded quotes, company names, and the current status of the request. **Constraint(s):** An existing request needs to have been created.
Business rules applicable to this Requirement:	• The user must be logged into the system. • Only the user can update their own request.
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 80: 6.2 View Vendor onboard Request

Functional Requirement	Explanation
Requirement number:	6.3
Requirement name (use case name):	Update Vendor onboard Request
Requirement short description:	The system should allow the requester (user) to update their vendor onboard request.
Requirement detailed description and constraints:	**Description:** The requester(user) requests to edit a vendor onboard request. Under the vendor section, the requester (user) should be able update their current vendor onboard requests. The system will prompt the user to enter the details that they wish to update. The user would update the details and validation will be performed. If the validation returns a success the details will be updated on the system. The system will notify the Procurement department of the update and request approval. **Constraint(s):** An existing request needs to have been created. A number of 3 Quotes and company names and emails must be added to the request for the user to be able to proceed with the process.
Business rules applicable to this Requirement:	• The user must be logged into the system. • Only the user can update their own request. • The business requires a minimum of 3 quotes for a vendor to be onboarded.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 81: 6.3 Edit Vendor onboard request

Functional Requirement	Explanation
Requirement number:	6.4
Requirement name (use case name):	Delete Vendor onboard Request
Requirement short description:	The system should allow the requester (user) to remove their vendor onboard request from the system.
Requirement detailed description and constraints:	**Description:** The user wishes to delete a specific onboard request. The user would select the specific request that they would like to delete. The user will delete the specific onboard request, it will be validated on the system to ensure that there are no associated records to the request and if the validations succeed the data will be removed from the system. **Constraint(s):** The user must first view the vendor onboard requests before they will be able to remove a request from the system. An existing request needs to have been created.
Business rules applicable to this Requirement:	• The user must be logged into the system. • Only the user can remove their own request.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 82: 6.4 Delete Vendor onboard Request

Functional Requirement	Explanation
Requirement number:	6.5
Requirement name (use case name):	Approve onboard request
Requirement short description:	The system should allow the GRC officer to approve a vendor onboarding request.
Requirement detailed description and constraints:	**Description:** The GRC officer requests to approve a vendor onboard request. The GRC officer will select the request that they wish to approve. The system will display the following details which relates to the onboard request: • Name • 3 quotes The GRC officer will approve the correct company on the system. The system will prompt the GRC officer to add the vendor's name, BEE rating, BEE certificate, and BEE expiration date to a Vendor Setup form. In the case of a sole supplier: The managing director will request to view the onboard request on the system. The following detail will be displayed: • Vendor Name • Relevant Reason • Optional quote The managing director will approve the setup request. The system will notify the GRC officer to add the vendor's name, BEE rating, BEE certificate, and BEE expiration date to a Vendor Setup form. In both cases of a sole supplier and a normal vendor the system will send a notification, along with the uncompleted Vendor Setup form containing the vendor's name, BEE rate, BEE certificate, and BEE expiration date, to the buyer/requester notifying them that they can proceed with the vendor setup process. **Constraint(s):** None
Business rules applicable to this Requirement:	• The GRC officer/ managing director must be logged into the system. • The vendor's name, BEE rating, BEE certificate, and BEE expiration date must be added to the form approval process to be completed.
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 83: 6.5 Approve onboard request

Functional Requirement	Explanation
Requirement number:	6.6
Requirement name (use case name):	Create vendor
Requirement short description:	The system should allow a vendor to be added on the system once the Vendor onboard request has been approved.
Requirement detailed description and constraints:	**Description:** The user wishes to create a vendor on the system. The user will select the vendor that they have requested and that has been approved and they will provide the following details: **Company Contact Information** • Telephone • Fax • Email • Point of Contact Title • Point of Contact Name • Contact Phone Number • Registered Address • Website URL • Contact Email **Company Overview Information** • Details of goods and services • Vat registration number • Vat Registration Document • Company Registration Number • Income tax Number • Tax clearance certificate • Signed Agreement Details • Signed Agreement Document • Insurance Cover Details • Insurance Cover Document • Payment Terms Details • License/ Accreditation Number and document **Bank Name** • Bank Name • Branch Code • Account Name • Account Type • Account Number • Bank Contact Name • Bank Contact Phone Number • Bank stamped confirmation letter The system will take these details and save them on the system's database. **Constraint(s):** • The system will not allow the user to proceed if they have not uploaded the accompanying documents to the following fields: • Company Registration Number and document • Income Tax Number and document
Business rules applicable to this Requirement:	• The user must be logged in to the system. • The vendor will only be able to be created once the onboarding request has been approved
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 84: 6.6 Create vendor

Functional Requirement	Explanation
Requirement number:	6.7
Requirement name (use case name):	View Vendor
Requirement short description:	The system should allow a specific vendor and the details associated with the vendor to be viewed.
Requirement detailed description and constraints:	**Description:** The user wishes to view a specific vendor. The user will search for the user using the following details: • Company name The system will search the systems database and retrieve the relevant vendor record and display it to the user where the user will be able to view the vendor details that have been stored on the system. The user will also be able to view more detail by clicking a view button next to a specific vendor record. **Constraint(s):** There needs to be a vendor added on the system
Business rules applicable to this Requirement:	• The user needs to be logged in to the system.
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 85: 6.7 View Vendor

Functional Requirement	Explanation
Requirement number:	6.8
Requirement name (use case name):	Update Vendor
Requirement short description:	The system should allow the user to update a specific vendor on the system.
Requirement detailed description and constraints:	**Description:** The user wishes to update a specific vendor. The user will select the vendor they wish to update and provide the updated details. The system will capture and validate the updated details. Finally, a success notification is then displayed and return the user back to the view vendor screen. . **Constraint(s):** There needs to be a vendor added onto the system.
Business rules applicable to this Requirement:	• The user needs to be logged in to the system.
Revision date and revision Number:	05/18/2023 Version 3.0
Priority:	High

Table 86: 6.8 Update Vendor

Functional Requirement	Explanation
Requirement number:	6.9
Requirement name (use case name):	Delete Vendor
Requirement short description:	The system should allow the user to delete a specific vendor record
Requirement detailed description and constraints:	**Description:** The user wishes to delete a specific vendor. The user would select the specific vendor that they would like to delete. The user will delete the specific vendor, it will be validated on the system to ensure that there are no associated records to the request and if the validations succeed the data will be removed from the system. **Constraint(s):** There needs to be a vendor added onto the system
Business rules applicable to this Requirement:	• The user needs to be logged into the system • Only a user with the correct access level will be allowed to remove a vendor from the system
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 87: 6.9 Delete Vendor

Functional Requirement	Explanation
Requirement number:	6.10
Requirement name (use case name):	Export due diligence vendor checklist
Requirement short description:	The system should allow the user to export a due diligence vendor checklist.
Requirement detailed description and constraints:	**Description:** The user would like to export a vendor due diligence checklist. The user would search for the vendor by using the following fields: • Company name The user would select the vendor and export the checklist which will be downloaded in a structured pdf format. **Constraint(s):** The due diligence checklist for the specific vendor needs to be completed.
Business rules applicable to this Requirement:	• The user needs to be logged in to the system. • The user with the correct access level will be able to export the due diligence checklist
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	High

Table 88: 6.10 Export due diligence vendor checklist

t

Functional Requirement	Explanation
Requirement number:	6.11
Requirement name (use case name):	Generate Sole Supplier Review Notification
Requirement short description:	The system should generate a notification to indicate that a Sole supplier should be reviewed.
Requirement detailed description and constraints:	**Description:** Notifications should be generated by the system which indicate that a Sole supplier should undergo a performance review. This notification will be sent on the first day of the month in which the vendor was first added to the system. **Constraint(s):** None
Business rules applicable to this Requirement:	• None
Revision date and revision Number:	03/19/2023 Version 1.0
Priority:	Medium

Table 89: 6.11 Generate Sole Supplier Review Notification

Functional Requirement	Explanation
Requirement number:	6.12
Requirement name (use case name):	Generate BEE Certificate Expiry Notification.
Requirement short description:	The system should generate a notification to indicate the expiry of a BEE certificate
Requirement detailed description and constraints:	**Description:** Notifications should be generated by the system which indicate that the expiry date of a BEE certificate which has been added to the system is nearing. This should be done 1 week before, 3 weeks before, as well as once the expiry date for the certificate has officially been reached. This notification should be sent to the GRC officer via email. **Constraint(s):** At least 1 BEE certificate should be present on the system.
Business rules applicable to this Requirement:	• None
Revision date and revision Number:	03/16/2023 Version 2.0
Priority:	Medium

Table 90: 6.12 Generate BEE Expiry Certification Notification

Functional Requirement	Explanation
Requirement number:	6.13
Requirement name (use case name):	Update Approve onboard request
Requirement short description:	The system should allow the user to update the due diligence checklist and other details related to it
Requirement detailed description and constraints:	**Description:** The GRC officer requests to update a vendor onboard request due diligence checklist details. The system will prompt the GRC officer to add the vendor's BEE rating, BEE certificate, and BEE expiration date, Due diligence checklist and POPI related details. The system will then capture and validate the updated details. Finally, a success notification is then displayed and return the user back to the View Approved or Pending onboard request screen. **Constraint(s):** None
Business rules applicable to this Requirement:	• None
Revision date and revision Number:	07/29/2023 Version 1.0
Priority:	Medium

Table 91: 6.13 Update Approve Onboard Request

Functional Requirement	Explanation
Requirement number:	6.14
Requirement name (use case name):	View Approved Or Pending Onboard Request
Requirement short description:	The system should allow the user to view approved or pending onboard request.
Requirement detailed description and constraints:	**Description:** The user wishes to view approved or pending onboard request. The user will search for the user using the following details: • Onboard request details The system will search the systems database and retrieve the relevant onboard request records and display it to the user where the user will be able to view the onboard request details and add, remove or update any due diligence details. **Constraint(s):** There needs to be a onboard request that has been created and not rejected on the system
Business rules applicable to this Requirement:	• None
Revision date and revision Number:	07/29/2023 Version 1.0
Priority:	Medium

Table 92: 6.14 View Approved or Pending Onboard Request

SUBSYSTEM 7 – REPORT SUBSYSTEM

Functional Requirement	Explanation
Requirement number:	7.1
Requirement name (use case name):	Generate approved vendor Report
Requirement short description:	The system should allow the user to generate a list of the approved vendors.
Requirement detailed description and constraints:	**Description:** The GRC officer request to generate an approved vendor report. The system will generate the report which includes a table list of all the vendors which have the status of "approved". The system will display the option to download the report in a PDF format. **Constraint(s):** There must be at one vendor with at least one procurement item they sell on the system.
Business rules applicable to this Requirement:	• Head of governance and compliance (Arina van er Merwe) must be logged into the system. • Only Head of governance and compliance (Arina van er Merwe) can generate this report.
Revision date and revision Number:	04/10/2023 Version 3.0
Priority:	High

Table 93: 7.1 Generate approved vendor report.

Functional Requirement	Explanation
Requirement number:	7.2
Requirement name (use case name):	Generate BEE spend report
Requirement short description:	The system should allow to generate a BEE spending report.
Requirement detailed description and constraints:	**Description:** The owner, financial department, business enablement department and governance and compliance department requests to generate a BEE spend report. The system will state the "Generated on:" followed by the date that the report was generated on and then the "Period:" followed by the period that the report was based on. The system will generate an overall BEE spend summary table, a MBA BEE spend summary table, a VEN BEE spend summary table, a ENG BEE spend summary table, a MTS BEE spend summary table, a CPT BEE spend summary table, with all table containing columns ranging from 1-8, a non-compliant, and a total column as well as the following rows procurement spend, procurement %, and procurement entitlement. This is followed by a total procurement entitlement BEE level This data will be retrieved from the system's database. The system will display the option to the stakeholders to download the report in a structured PDF format. **Constraint(s):** Only The owner, financial department, business enablement department and governance and compliance department will be able to generate this report.
Business rules applicable to this Requirement:	• Stakeholders must be logged into the system.
Revision date and revision Number:	04/10/2023 Version 3.0
Priority:	High

Table 94: 7.2 Generate BEE spend report.

Functional Requirement	Explanation
Requirement number:	7.3
Requirement name (use case name):	Generate Vendor Spend Report
Requirement short description:	The system should allow to generate a vendor spending report.
Requirement detailed description and constraints:	**Description:** The financial department and governance and compliance department requests to generate a vendor spend report. The system will generate a table of the vendors with the following Information: Supplier, account code, account name, budget, branch, level, amount spent, count. This will be retrieved from the system's database. The system will display the option to all the previously mentioned stakeholders to download the specified report in a structured PDF format. **Constraint(s):** Business enablement department will not be able to generate this report.
Business rules applicable to this Requirement:	• Users of the financial department and governance and compliance department must be logged out of the system.
Revision date and revision Number:	04/10/2023 Version 3.0
Priority:	High

Table 95: 7.3 Generate Vendor Spend Report

Functional Requirement	Explanation
Requirement number:	7.4
Requirement name (use case name):	Generate Consumable Inventory management report
Requirement short description:	The system should be able to generate a consumable inventory management report.
Requirement detailed description and constraints:	**Description:** The financial department and Business enablement department requests to generate the consumable inventory management report. The system will state the “Generated on:” followed by the date that the report was generated on and then the “Period:” followed by the time period that the report was based on. The system will generate a table with the following columns with the respective information in each column Item name and ranging from January-December, as well as the following rows based on consumable inventory. This is then followed by a bar chart based on consumable inventory on hand per month. The system will display the option to all the previously mentioned stakeholders to download the report in a structured PDF format. **Constraint(s):** The governance and compliances department will not be able to see this report.
Business rules applicable to this Requirement:	• Users of the financial department and Business enablement department must be logged out of the system.
Revision date and revision Number:	04/10/2023 Version 3.0
Priority:	High

Table 96: 7.4 Generate Consumable Inventory management report

Functional Requirement	Explanation
Requirement number:	7.5
Requirement name (use case name):	Generate Business Unit allocation Report
Requirement short description:	The system should allow to generate a business unit allocation spend report
Requirement detailed description and constraints:	**Description:** The financial department, Business enablement department and governance and compliance department requests to generate the business unit allocation report. The system will state the “Department:” followed by the department chosen to do the report on. The system will generate a table about the business unit spending budget category based on that specific department’s spending. The system will generate a bar chart that clearly indicates what is the spending report of each budget unit is within the department chosen. The system will display the option to all the previously mentioned stakeholders to download the report in a structured PDF format. **Constraint(s):** Only the financial department, Business enablement department and governance and compliance department can generate this report
Business rules applicable to this Requirement:	• Users of the financial department and Business enablement department and governance and compliance department must be logged out of the system.
Revision date and revision Number:	04/10/2023 Version 3.0
Priority:	High

Table 97: 7.5 Generate Business Unit allocation report.

Functional Requirement	Explanation
Requirement number:	7.6
Requirement name (use case name):	Generate Budget Variance Report
Requirement short description:	The system should allow to generate budget variance report.
Requirement detailed description and constraints:	**Description:** The financial department and Business enablement department requests to generate the budget variance report. The system will state the “Generated on:” followed by the date that the report was generated on and then the “Period:” followed by the time period that the report was based on. The system will generate a table with the following columns with the respective information in each column: Month, budget category, budget amount (R), actual spent (R), variance (R), variance (%). This is followed by a line chart to show total expenses on a month-based period (Month by month for a year). Then followed by a bar chart that shows total expenses on a category-based period (each budget category over a year). The system will display the option to all the previously mentioned stakeholders to download the report in a structured PDF format. **Constraint(s):** Governance and compliances will not be able to view this report.
Business rules applicable to this Requirement:	• Users of the financial department and Business enablement department must be logged out of the system.
Revision date and revision Number:	04/10/2023 Version 3.0
Priority:	High

Table 98: 7.6 Generate Budget Variance Report

Functional Requirement	Explanation
Requirement number:	7.7
Requirement name (use case name):	View Generate Report.
Requirement short description:	The system should allow the user to view all the reports that they can generate.
Requirement detailed description and constraints:	**Description:** The user wishes to view which reports they would be able to generate. The system will then navigate to the view Generate report page where all the repost that that user can generate will be. **Constraint(s):** Users will only be able to view reports that they would be able to generate based on their role’s authority.
Business rules applicable to this Requirement:	• Users must be logged into the system.
Revision date and revision Number:	04/10/2023 Version 3.0
Priority:	High

Table 99: UC 7.7 View Generate Report

2.5 CONCLUSION

This Concludes the updated requirement list for the Procion system. The updated requirement list with the following subsystems: The user sub-system, Admin sub-system, Procurement sub-system, Finance sub-system, Inventory sub-system, Vendor sub-system and a Report sub-system with the description and explanation of each.

2. FUNCTIONAL REQUIREMENT CLIENT SIGN OFF

PROCION SYSTEM: Design and Development

Business Name: MOYO Business Advisory

I,______________________, Hereby confirm that I have read the list of updated functional requirements and agree that the information contained therein is complete, accurate and meets the requirements of the project.

BY SIGNING I CONFIRM THAT:

1. **I have reviewed the changes made to the functional requirements.**
2. **I accept the changes as complete and satisfactory.**

Signed by:
Name:
Date:
Signature:

1. TECHNICAL USE CASE DIAGRAM

1.1 INTRODUCTION

In this section we provide a visual representation of the technical use case diagrams for each sub system. This clearly states each actor and the role that they play within the Procion system.

1.2 USE CASE DIAGRAMS

1.2.1 SUBSYSTEM 1 - USER SUBSYSTEM

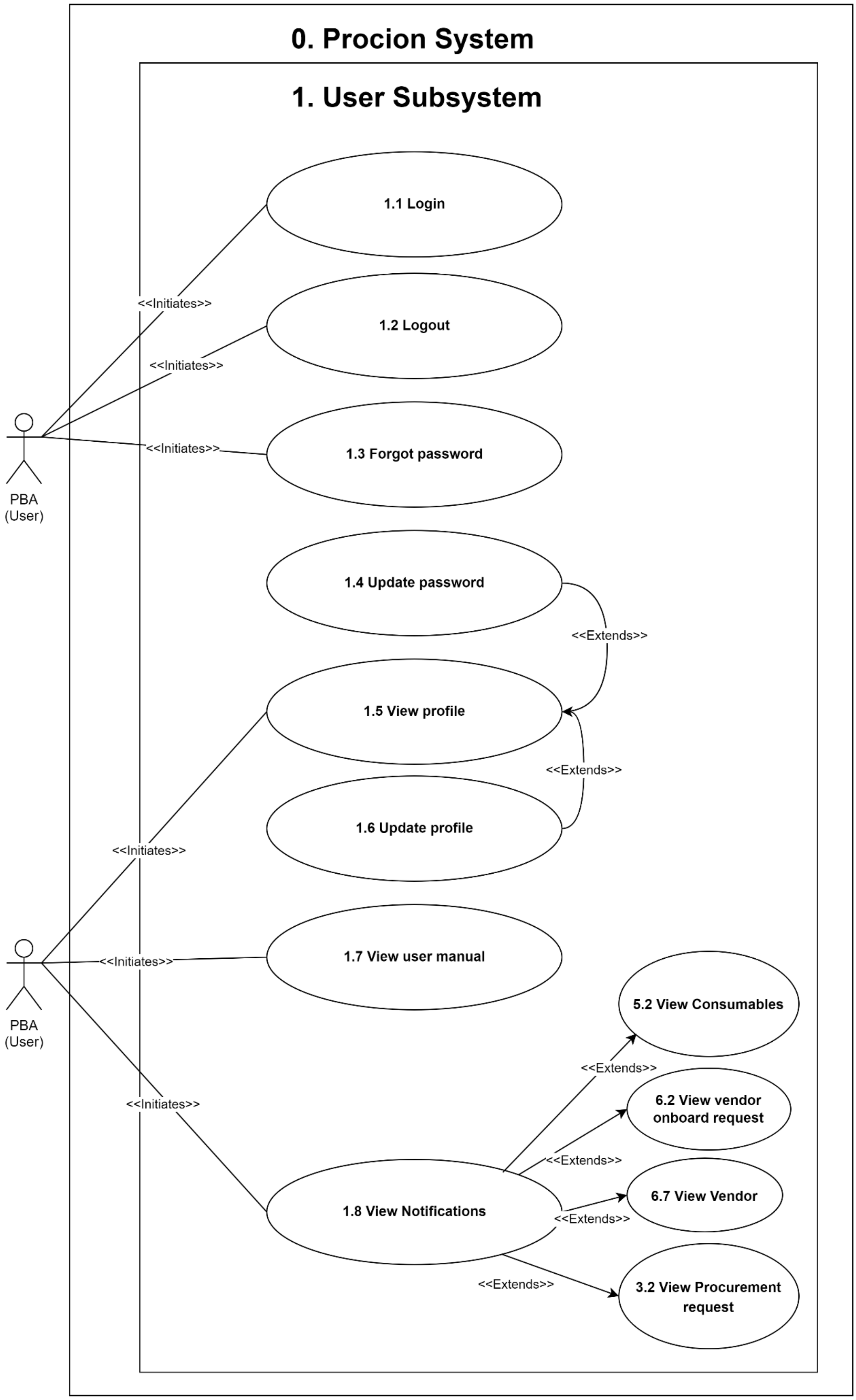


Figure 1 Use Case Diagram - Subsystem 1

1.2.2 SUBSYSTEM 2 – ADMIN SUBSYSTEM

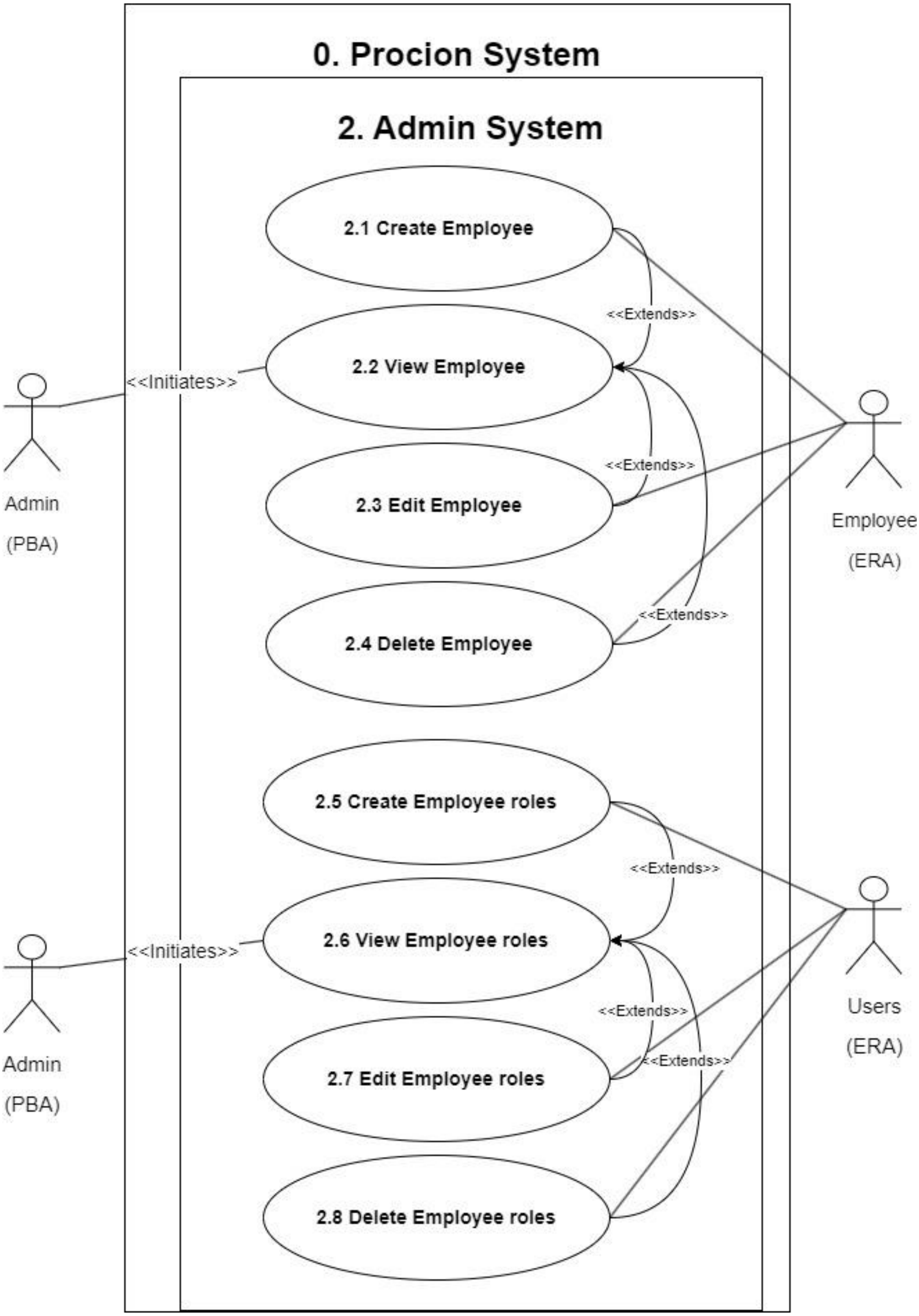


Figure 2 Use Case Diagram - Subsystem 2

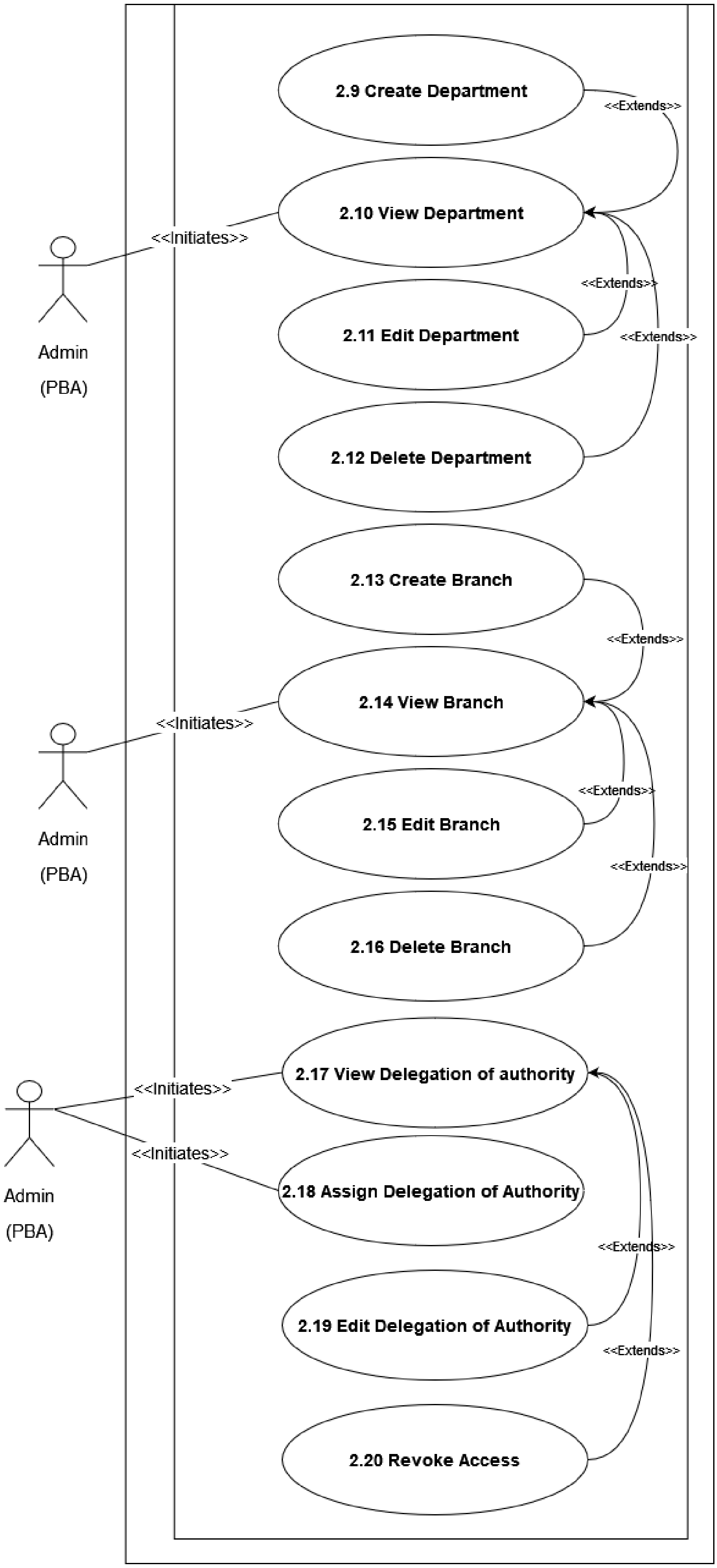


Figure 3 Use Case Diagram - Subsystem 2 Continuation 1

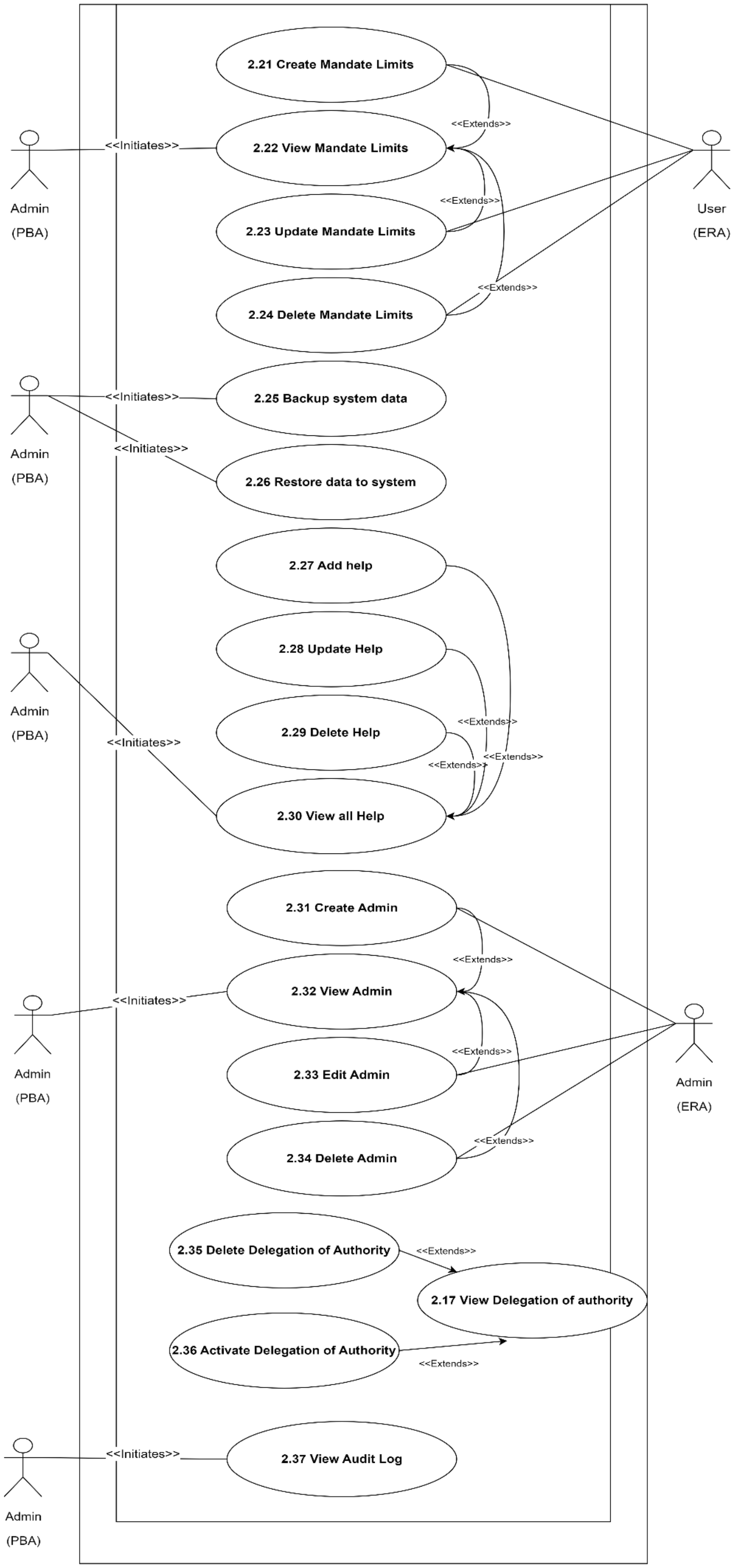


Figure 4 Use Case Diagram - Subsystem 2 Continuation 2

1.2.3 SUBSYSTEM 3 - PROCUREMENT SUBSYSTEM

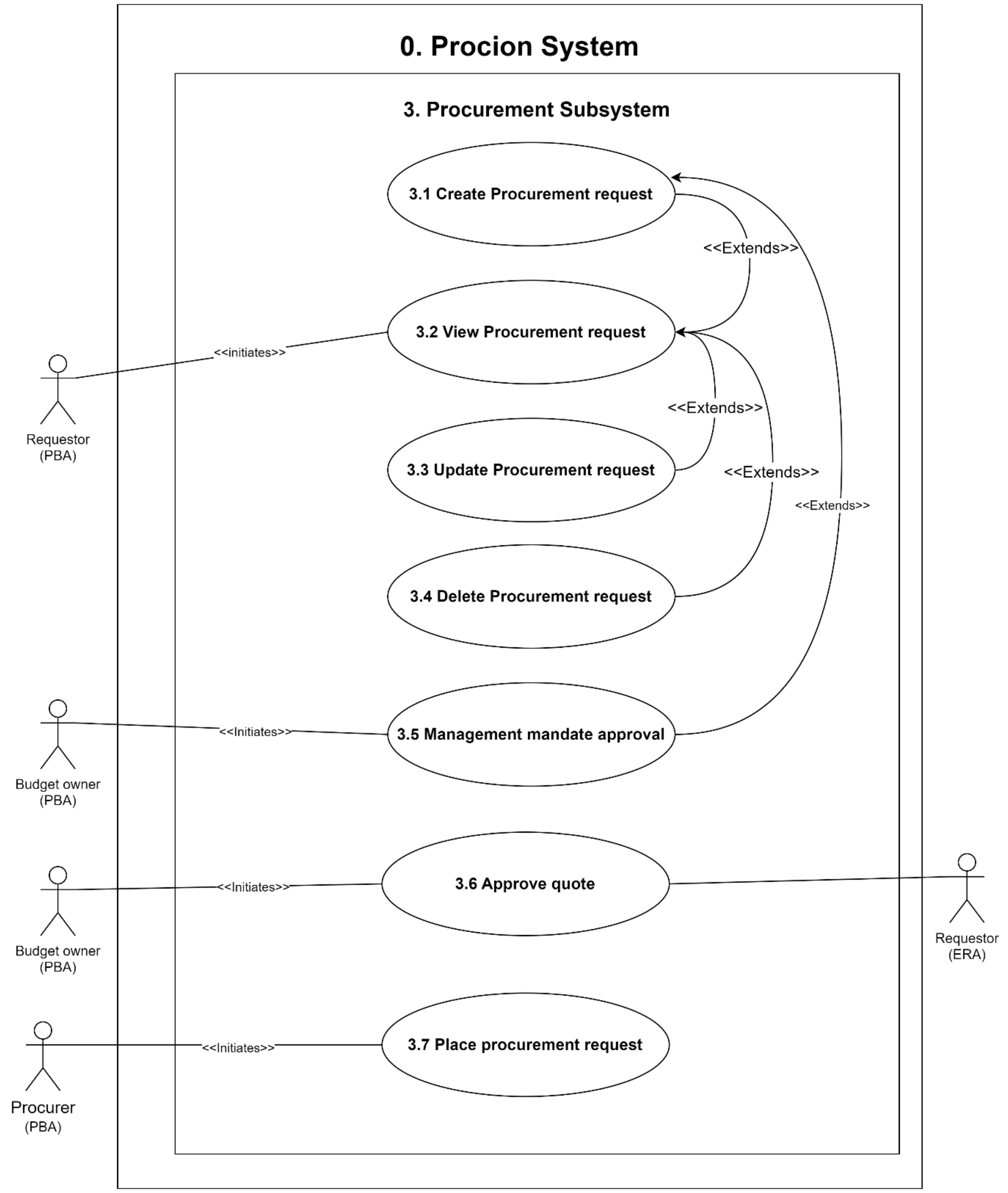


Figure 5 Use Case Diagram - Subsystem 3

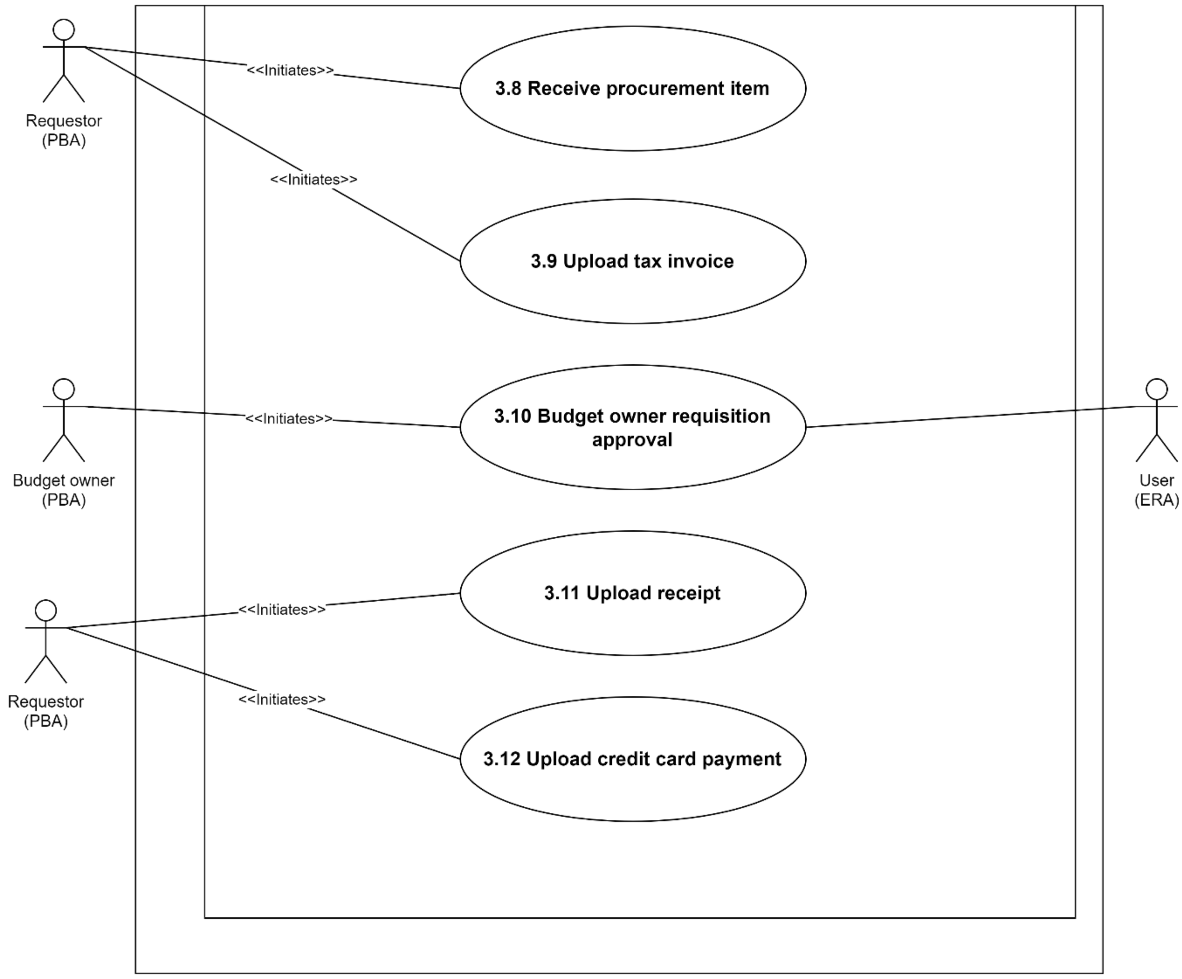


Figure 6 Use Case Diagram - Subsystem 3 Continuation

1.2.4 SUBSYSTEM 4 - FINANCE SUBSYSTEM

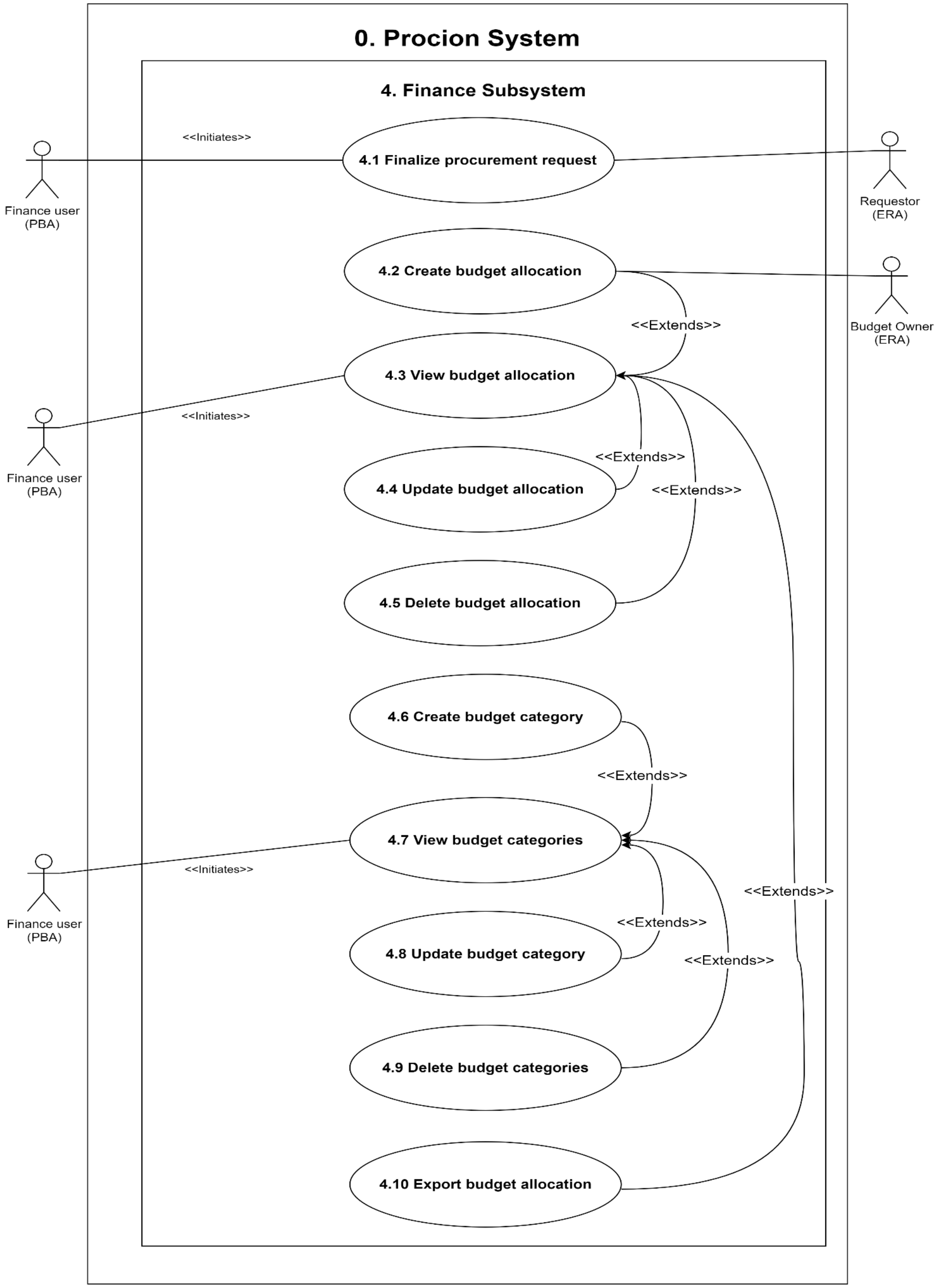


Figure 7 Use Case Diagram - Subsystem 4

1.2.5 SUBSYSTEM 5 - INVENTORY SUBSYSTEM

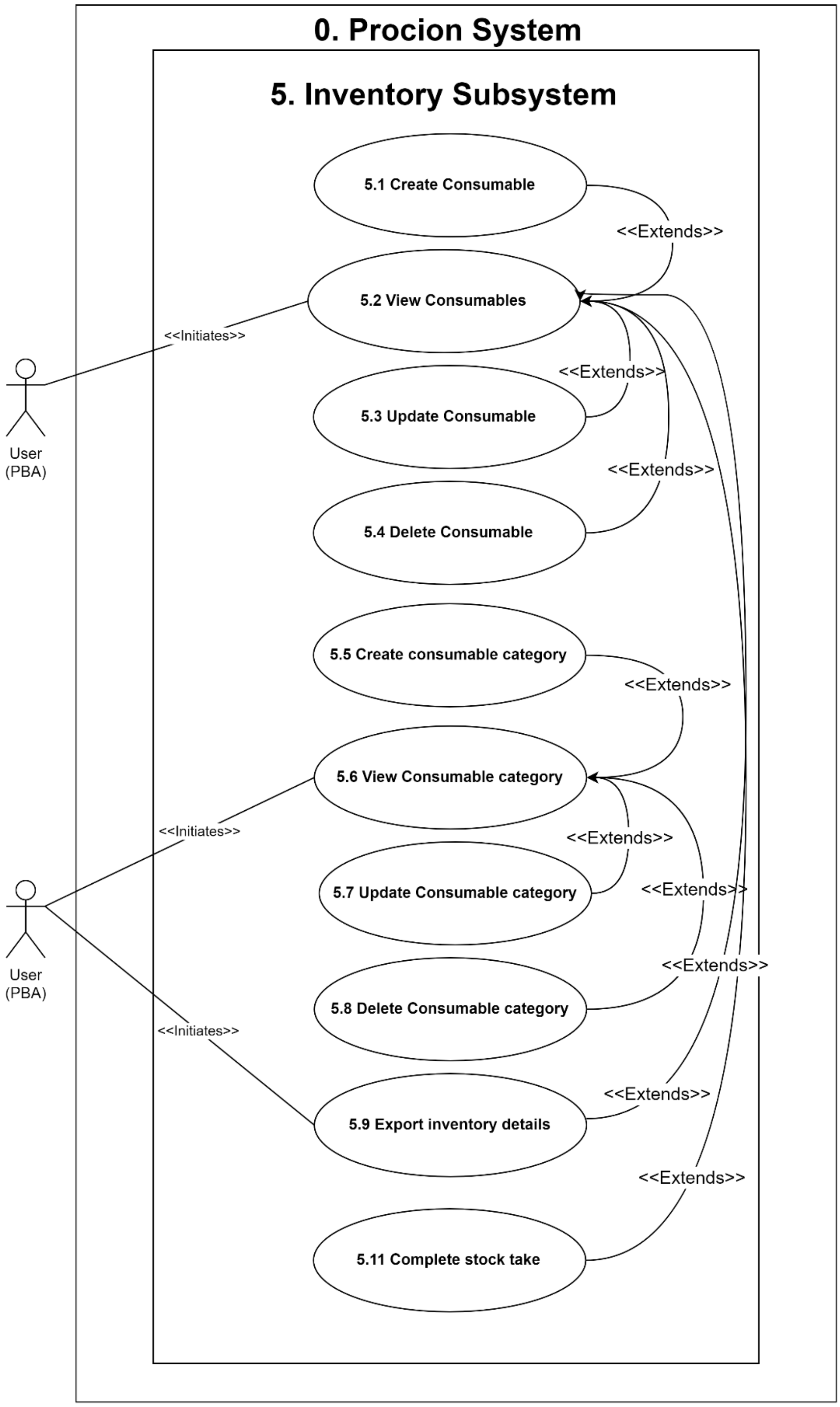


Figure 8 Use Case Diagram - Subsystem 5

1.2.6 SUBSYSTEM 6 - VENDOR SUBSYSTEM

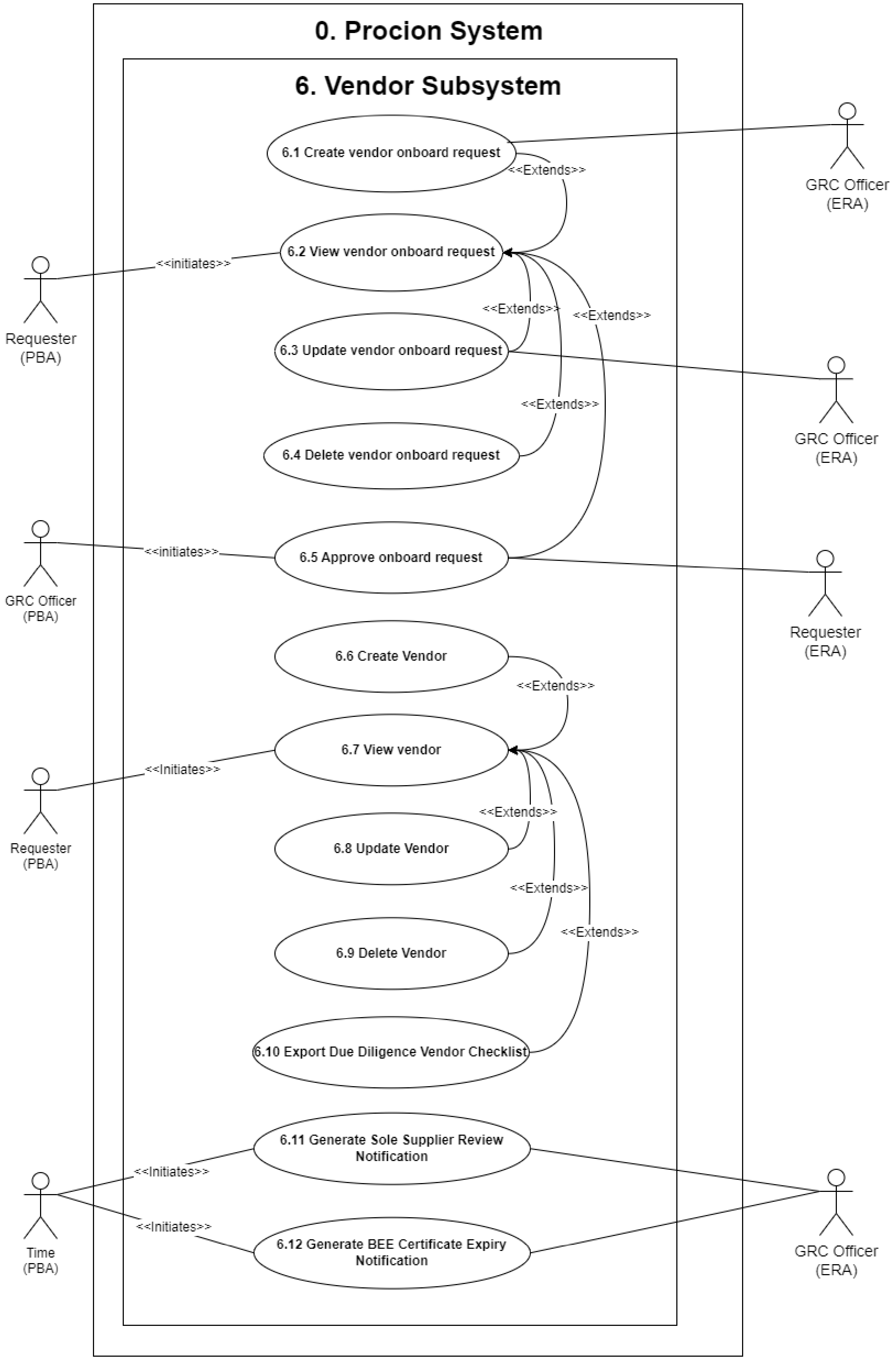


Figure 9 Use Case Diagram - Subsystem 6

1.2.7 SUBSYSTEM 7 - REPORTING SUBSYSTEM

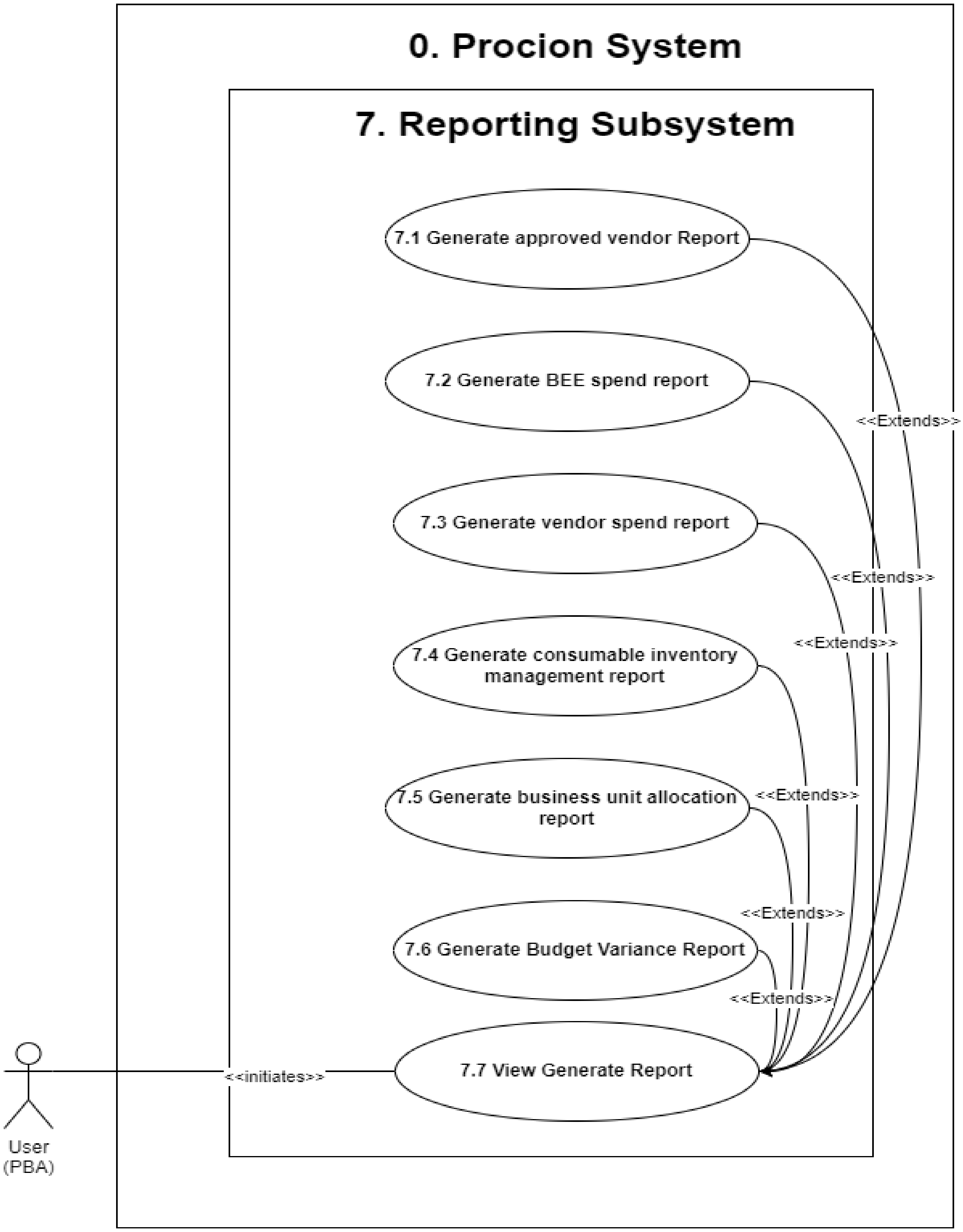


Figure 10 Use Case Diagram - Subsystem 7

1.3 CONCLUSION

This concludes the Use Case Diagrams of the Procion system. This will help give a better understanding of how the Procion system will operate.

2. LOGICAL USE CASE NARRATIVES

2.1 INTRODUCTION

In this section we will discuss the logical narrative for the following use cases: 2.26,2.37,4.10, 5.9, 7.1 , 7.2 , 7.3, 7.4, 7.5, 7.6, 7.7 of the Procion system.

2.2 LOGICAL USE CASE NARRATIVES

USE CASE 2.26

System Name: Procion System Sub-System: 2 Administration		
Author: Leon Combrinck	Date: 2023/07/10	Version: 1.0

Use case name:	Restore data to the system	**USE CASE TYPE**
Use case id:	2.26	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	Moyo	System Design: ☐
Primary business actor:	Admin (PBA)	
Primary system actor:	None	
Other participating actors:	None	
Other interested stakeholders:	None	
Description:		

<table>
<tr><td>Pre-condition:</td><td colspan="2">The admin must be logged into the system.</td></tr>
<tr><td>Trigger:</td><td colspan="2">The admin requests to restore data to the system.</td></tr>
<tr><td rowspan="5">Typical course Of events:</td><td>Actor Action</td><td>System Response</td></tr>
<tr><td>Step 1: The admin requests to restore data to the system.</td><td>Step 2: The system requests the admin to provide the backed-up file the Admin wishes to restore the system with.</td></tr>
<tr><td>Step 3: The admin provides the backed-up file.

[ALT]</td><td>Step 4: the system request Confirmation for Restoring the database.</td></tr>
<tr><td>Step 5: The Admin provides the Confirmation for restoring the database.

[ALT]</td><td>Step 6: The system then restores the data in the database back to the data of the backed-up file thus restoring the data on the system.</td></tr>
<tr><td></td><td>Step 7: The system will display a success notification regarding the successful restoring of data to the system, to the admin.

[ALT]</td></tr>
<tr><td rowspan="3">Alternate courses:</td><td colspan="2">ALT-Step 3: The Admin does not provide any backup files. The use case ends.</td></tr>
<tr><td colspan="2">ALT-Step 4: The Admin does not provide confirmation for restoring the database. The use case ends.</td></tr>
<tr><td colspan="2">ALT-Step 5: The system displays an unsuccessful notification regarding the unsuccessful restoring of data to the admin. The use case ends.</td></tr>
</table>

Conclusion:	The data form the chosen backup is restored to the system and a success notification is displayed.
Post-condition:	The system data is restored.
Business rules	Only the admin can restore data to the system.
Implementation constraints and specifications	• None
Assumptions:	• None
Open issues:	• None

Table 100: 2.26 Restore System Data

USE CASE 2.37

System Name: Procion System Sub-System: 2		
Author: Leon Combrinck	Date: 05 August 2023	Version: 1.0

Use case name:	View Audit Log	**USE CASE TYPE**
Use case id:	2.37	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	Moyo	System Design: ☐
Primary business actor	Admin	
Primary system actor	None	

<table>
<tr><td>Other participating actors:</td><td colspan="2">None</td></tr>
<tr><td>Other interested stakeholders:</td><td colspan="2">None</td></tr>
<tr><td>Description:</td><td colspan="2">The use case describes the event where a Admin would like to view Audit log on the system. The user will then request the system to view the Audit Log on the system. The system will then load all Audit log that are currently on the system and display them with the following information:
• Time Completed
• User
• Action Taken
The system promts the Admin to enter sort by criteria.The Admin will then request the system to be able to Sort by for the audit log. The system sorts the audit log in order of the options that the admin chose and load the sorted audit log to the Admin.</td></tr>
<tr><td>Pre-condition:</td><td colspan="2">The Admin has to be logged in to the system.</td></tr>
<tr><td>Trigger:</td><td colspan="2">The user requests to view Audit Log details</td></tr>
<tr><td rowspan="2">Typical course Of events:</td><td>Actors Action</td><td>Systems response</td></tr>
<tr><td>Step 1: The Admin request to view Audit Log on the system</td><td>Step 2: The system navigates to and retrieve the Audit Log details in the AuditLog table with the following infromation:
• Time Completed
• User
• Action Taken
The system will displays all the retrieved Audit log details to the Admin.
The system then promts the admin to enter a Sort By criteria.</td></tr>
</table>

	Step3: The Admin enters a Sort By criterea. • User or Action	**Step 4:** The system then Sort By through the retrieved Audit log list to match the Audit Log list with the Sort By criteria of the admin by comparing the following attributes: • User or Action The system then displays the matching fields.
Alternate courses:	**None**	
Conclusion:	The Admin is displayed the view Audit Log details	
Post-condition:	The Admin is displayed the view Audit Log details	
Business rules	• Only The Admin can view Audit Log	
Implementation constraints and specifications	• The Admin requires internet access	
Assumptions:	• None	
Open issues:	• None	

Table 101: 2.37 View Audit log

USE CASE 4.10

System Name: Procion System Sub-System: 4		
Author: Leon Combrinck	Date: 05 August 2023	Version: 1.0

Use case name:	Export Budget Allocation	**USE CASE TYPE**
Use case id:	4.10	Business Requirements: ☐

Priority:	High	System Analysis: ☒	
Source:	Moyo	System Design: ☐	
Primary business actor	Finance User		
Primary system actor	None		
Other participating actors:	None		
Other interested stakeholders:	None		
Description:	The use case describes the event where a Finance User would like to Export Budget Allocation on the system. The Finance User requests the system to Export the Budget allocation into a Excel File. The system will include all the budget allocation and budget line and budget Category details. The system download the excel file generated onto the Finance Users device.		
Pre-condition:	The Finance User has to be logged in to the system.		
Trigger:	The Finance User requests to Export Budget allocation details		
Typical course Of events:	**Actors Action**	**Systems response**	
	Step 1: The Finance user requests to export the Budget Allocation details	**Step 2:** The system retrieves all of the records contained in the **Budget_Allocation** Entity and the **Budegt_Lines** entity and **Budget_Category** entity	
		Step 3: The system generates and downloads the Export Budget allocation details based on the data retrieved from the system on the Finance Users device	
Alternate courses:	**None**		

Conclusion:	The Export Budegt allocation excel File is downloaded on the Finance User Device
Post-condition:	The Budget allocation and budget lines is Downloaded on the Finance Users device.
Business rules	• Only The Finance User Budget Owner can Export Budget Alloaction
Implementation constraints and specifications	• The Finance User requires internet access
Assumptions:	• None
Open issues:	• None

Table 102: 4.10 Export budget Allocation

USE CASE 5.9

System Name: Procion System Sub-System: 5		
Author: Jason van der Merwe	Date: 25 July 2023	Version: 1.0

Use case name:	Export Inventory Details	**USE CASE TYPE**
Use case id:	5.9	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	Moyo	System Design: ☐
Primary business actor	User	
Primary system actor	None	

Other participating actors:	None	
Other interested stakeholders:	None	
Description:	The use case describes the event where a user would like to export the inventory details of all consumables of the system The user Requests to export the inventory details. The system retrieves all the data stored for the consumables on the system along with their respective categories. The system generates the Inventory details report and displays it to the user	
Pre-condition:	The user has to be logged in to the system.	
Trigger:	The user requests to export the inventory details	
Typical course Of events:	**Actors Action**	**Systems response**
	Step 1: The user requests to export the inventory details	**Step 2:** The system retrieves all of the records contained in the **Consumables** Entity and the **Consumable_Category** entity
	Step 3: The user selects the inventory item to be updated	**Step 4**: The system generates inventory details report based on the data retrieved from the system and displays it to the user
Alternate courses:	**None**	
Conclusion:	The user is displayed the Inventory details report	
Post-condition:	The user is displayed the Inventory details report	
Business rules	None	
Implementation constraints and	• The user requires internet access	

specifications	
Assumptions:	• None
Open issues:	• None

Table 103: 5.9 Export Inventory Details

USE CASE 7.1

System Name: Procion System Sub-System: 7		
Author: Leon Combrinck	Date: 25 July 2023	Version: 1.0

Use case name:	Generate Approved Vendor Report	**USE CASE TYPE**
Use case id:	7.1	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	Moyo	System Design: ☐
Primary business actor	User	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	

Description:	The use case describes the event where a user would like to generate the list of all the Approved Vendors on the system The user Requests to generate approved vendors list. The system retrieves the data stored for the Vendors, Vendor_Status on the system. The system generates the approved vendor report and displays it to the user	
Pre-condition:	The user must be logged in to the system. Only GRC can generate this report.	
Trigger:	The user requests to generate approved vendor report.	
Typical course Of events:	**Actors Action**	**Systems response**
	Step 1: The user requests to generate approved vendor report details **[ALT]**	**Step 2:** The system retrieves all of the records contained in the **Vendor** Entity and the **Vendor_Status** Entity with the condition that the status is approved on the system.
		Step 3: The system generates approved vendor details report based on the data retrieved from the system and displays it to the user **[ALT]**
Alternate courses:	**ALT Step 1:** The User requests to download approved vendor report details. Use case Continues to step 2. **ALT Step 3:** The system downloads the approved vendor details based on the data retrieved from the system and downloads the report on the user's device. **The use Case Ends.**	
Conclusion:	The user is displayed the approved vendor details report	
Post-condition:	The user is displayed the approved vendor details report	
Business rules	Only the GRC can generate this report	

Implementation constraints and specifications	• The user requires internet access
Assumptions:	• • None
Open issues:	• None

Table 104: 7.1 Generate Approved Vendor Report

USE CASE 7.2

System Name: Procion System Sub-System: 7		
Author: Leon Combrinck	Date: 08 August 2023	Version: 1.0

Use case name:	Generate BEE Spend Report	**USE CASE TYPE**
Use case id:	7.2	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	Moyo	System Design: ☐
Primary business actor	User	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	

Description:	The use case describes the event where a user would like to generate the BEE Spend Report on the system The user Requests to generate the BEE spend details. The system retrieves all the data stored for the Procurement_Details, Procurement_Request, Vendor, Employee, Branch, Budget_Lines, Budget_Category, Budget Allocation on the system. The system request the user to enter start to end date criteria. The user provides date period criteria to the system. The system generates the BEE spend report filter according to the date criteria provided and displays it to the user.	
Pre-condition:	The user has to be logged in to the system. Only GRC can generate this report.	
Trigger:	The user requests to generate the BEE spend report details	
Typical course Of events:	**Actors Action**	**Systems response**
	Step 1: The user requests to generate the BEE Spend details **[ALT]**	**Step 2:** The system retrieves all of the records contained in the **Procurement_Details** Entity and the **Procurement_Request** entity and the **Vendor** entity and the **Employee** entity and the **Branch** entity and the **Budget_Lines** entity and the **Budget_Category** entity
		Step 3: The System requests the user to provide the following details from the **Procurement_Details** entity: • Full_Payment_Due_Date
	Step 4: The User Provides the requested details and generates the report. **[ALT]**	**Step 5:** The system validate to details provided to ensure the information provided is correct. The system generates BEE Spend details report based on the data retrieved from the system and the

		date condition given by the user then displays it to the user **[ALT]**
Alternate courses:	**ALT Step 1:** The User requests to download BEE Spend report details. Use case Continues to step 2. **ALT Step 4:** The user does not request to generate the report. **The usecase Ends** **ALT Step 5a:** The Validation of the details Failed. The system displays a validation error message. **Return to step 4** **ALT Step 5b:** The system downloads the BEE Spend details based on the data retrieved from the system and downloads the report on the user's device. **The use Case Ends.**	
Conclusion:	The user is displayed the BEE Spend details report	
Post-condition:	The user is displayed the BEE Spend details report	
Business rules	Only The GRC can generate this report	
Implementation constraints and specifications	• The user requires internet access	
Assumptions:	• • None	
Open issues:	• None	

Table 105: 7.2 Generate BEE Spend Report

USE CASE 7.3

System Name: Procion System Sub-System: 7		
Author: Leon Combrinck	Date: 08 August 2023	Version: 1.0

Use case name:	Generate Vendor Spend Report	**USE CASE TYPE**

Use case id:	7.3	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	Moyo	System Design: ☐
Primary business actor	User	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	
Description:	The use case describes the event where a user would like to generate the Vendor Spend Report on the system The user Requests to generate the Vendor spend details. The system retrieves all the data stored for the Procurement_Details, Procurement_Request, Vendor, Employee, department, Branch, Budget_Lines, Budget_Category, Budget Allocation on the system. The system generates the Vendor spend report and displays it to the user	
Pre-condition:	The user has to be logged in to the system. Only GRC can generate this report.	
Trigger:	The user requests to generate the Vendor spend report details	
Typical course Of events:	**Actors Action**	**Systems response**
	Step 1: The user requests to generate the Vendor Spend details **[ALT]**	**Step 2:** The system retrieves all of the records contained in the **Procurement_Details** Entity and the **Procurement_Request** entity and the **Vendor** entity and the **Employee** entity and the **Department** entity and the

		Branch entity and the **Budget_Lines** entity and the **Budget_Category** entity and the **Budget_Allocation** entity
		Step 3: The system generates Vendor Spend details report based on the data retrieved from the system and displays it to the user **[ALT]**
Alternate courses:	**ALT Step 1:** The User requests to download vendor spend report details. Use case Continues to step 2. **ALT Step 3:** The system downloads the vendor spend details based on the data retrieved from the system and downloads the report on the user's device. **The use Case Ends.**	
Conclusion:	The user is displayed the Vendor Spend details report	
Post-condition:	The user is displayed the Vendor Spend details report	
Business rules	Only The GRC can generate this report	
Implementation constraints and specifications	• The user requires internet access	
Assumptions:	• • None	
Open issues:	• None	

Table 106: 7.3 Generate Vendor Spend Report

USE CASE 7.4

System Name: Procion System Sub-System: 7		
Author: Leon Combrinck	Date: 05 August 2023	Version: 1.0

Use case name:	Generate Consumable Inventory Management Report	**USE CASE TYPE**
Use case id:	7.4	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	Moyo	System Design: ☐
Primary business actor	User	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	
Description:	The use case describes the event where a user would like to generate the Consumable Inventory Management Report on the system The user Requests to generate the Consumable Inventory Management details. The system retrieves all the data stored for the Consumable, Consumable_History on the system. The system generates the Consumable Inventory Management report and displays it to the user	
Pre-condition:	The user has to be logged in to the system. Only Business Enablement can generate this report.	
Trigger:	The user requests to generate the Consumable Inventory Management report details	
Typical course Of events:	**Actors Action**	**Systems response**
	Step 1: The user requests to generate the	**Step 2:** The system retrieves all of the records contained in the

	Consumable Inventory Management details **[ALT]**	**Consumable** Entity and the **Consumable_History** entity
		Step 3: The System requests the user to provide the following details from the **Consumable_History** entity: • DateCaptured
	Step 4: The User Provides the requested details and generates the report. **[ALT]**	**Step 5:** The system validate to details provided to ensure the information provided is correct. The system generates Consumable Inventory Management details report based on the data retrieved from the system and the date condition given by the user then displays it to the user **[ALT]**
Alternate courses:	**ALT Step 1:** The User requests to download Consumable Inventory Management report details. Use case Continues to step 2. **ALT Step 4:** The user does not request to generate the report. **The usecase Ends** **ALT Step 5a:** The Validation of the details Failed. The system displays a validation error message. **Return to step 4** **ALT Step 5b:** The system downloads the Consumable Inventory Management details based on the data retrieved from the system and downloads the report on the user's device. **The use Case Ends.**	
Conclusion:	The user is displayed the Consumable Inventory Management details report	

Post-condition:	The user is displayed the Consumable Inventory Management details report
Business rules	Only The Business Enablement can generate this report
Implementation constraints and specifications	• The user requires internet access
Assumptions:	• • None
Open issues:	• None

Table 107: 7.4 Generate Consumable Inventory Management Report

USE CASE 7.5

System Name: Procion System Sub-System: 7		
Author: Leon Combrinck	Date: 05 August 2023	Version: 1.0

Use case name:	Generate Business Unit Allocation Report	**USE CASE TYPE**
Use case id:	7.5	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	Moyo	System Design: ☐
Primary business actor	User	
Primary system actor	None	
Other participating actors:	None	

<table>
<tr><td>Other interested stakeholders:</td><td colspan="2">None</td></tr>
<tr><td>Description:</td><td colspan="2">The use case describes the event where a user would like to generate the Business Unit Allocation Report on the system

The user Requests to generate the Business Unit Allocation details. The system retrieves all the data stored for the Budget_Lines, Budget_Category, Budget_Allocation, Department on the system. The system generates the Business Unit Allocation report and displays it to the user</td></tr>
<tr><td>Pre-condition:</td><td colspan="2">The user has to be logged in to the system.

Only Finance can generate this report.</td></tr>
<tr><td>Trigger:</td><td colspan="2">The user requests to generate the Business Unit Allocation report details</td></tr>
<tr><td rowspan="3">Typical course Of events:</td><td>Actors Action</td><td>Systems response</td></tr>
<tr><td>Step 1: The user requests to generate the Business Unit Allocation details

[ALT]</td><td>Step 2: The system retrieves all of the records contained in the Budget_Lines Entity and the Budget_Category entity and the Budget_Allocation entity and the Department entity</td></tr>
<tr><td></td><td>Step 3: The System requests the user to provide the following details from the Budget_Allocation entity:

• DateCreated

And the Department Entity:

• Name</td></tr>
</table>

	Step 4: The User Provides the requested details and generates the report. **[ALT]**	**Step 5:** The system validate to details provided to ensure the information provided is correct. The system generates Business Unit Allocation details report based on the data retrieved from the system and the date condition given by the user then displays it to the user **[ALT]**
Alternate courses:	**ALT Step 1:** The User requests to download Business Unit Allocation report details. Use case Continues to step 2. **ALT Step 4:** The user does not request to generate the report. **The usecase Ends** **ALT Step 5a:** The Validation of the details Failed. The system displays a validation error message. **Return to step 4** **ALT Step 5b:** The system downloads the Business Unit Allocation details based on the data retrieved from the system and downloads the report on the user's device. **The use Case Ends.**	
Conclusion:	The user is displayed the Business Unit Allocation details report	
Post-condition:	The user is displayed the Business Unit Allocation details report	
Business rules	Only The Finance can generate this report	
Implementation constraints and specifications	• The user requires internet access	
Assumptions:	• • None	
Open issues:	• None	

Table 108: 7.5 Generate Business Unit Allocation Report

USE CASE 7.6

System Name: Procion System Sub-System: 7

Author: Leon Combrinck	Date: 05 August 2023	Version: 1.0

Use case name:	Generate Budget Variance Report		USE CASE TYPE
Use case id:	7.6		Business Requirements: ☐
Priority:	High		System Analysis: ☒
Source:	Moyo		System Design: ☐
Primary business actor	User		
Primary system actor	None		
Other participating actors:	None		
Other interested stakeholders:	None		
Description:	The use case describes the event where a user would like to generate the Budget Variance Report on the system The user Requests to generate the Budget Variance details. The system retrieves all the data stored for the Budget_Lines, Budget_Category, Budget_Allocation on the system. The system generates the Budget Variance report and displays it to the user		
Pre-condition:	The user has to be logged in to the system. Only Business Owner can generate this report.		
Trigger:	The user requests to generate the Budget Variance report details		
	Actors Action	**Systems response**	

<table>
<tr><td rowspan="3">Typical course Of events:</td><td>Step 1: The user requests to generate the Budget Variance details

[ALT]</td><td>Step 2: The system retrieves all of the records contained in the Budget_Lines Entity and the Budget_Category entity and the Budget_Allocation entity</td></tr>
<tr><td></td><td>Step 3: The System requests the user to provide the following details from the Budget_Allocation entity:

• DateCreated</td></tr>
<tr><td>Step 4: The User Provides the requested details and generates the report.

[ALT]</td><td>Step 5: The system validate to details provided to ensure the information provided is correct.

The system generates Budget Variance details report based on the data retrieved from the system and the date condition given by the user then displays it to the user

[ALT]</td></tr>
<tr><td>Alternate courses:</td><td colspan="2">ALT Step 1: The User requests to download Budget Variance report details. Use case Continues to step 2.

ALT Step 4: The user does not request to generate the report. The usecase Ends

ALT Step 5a: The Validation of the details Failed. The system displays a validation error message. Return to step 4

ALT Step 5b: The system downloads the Budget Variance details based on the data retrieved from the system and downloads the report on the user’s device. The use Case Ends.</td></tr>
<tr><td>Conclusion:</td><td colspan="2">The user is displayed the Budget Variance details report</td></tr>
<tr><td>Post-condition:</td><td colspan="2">The user is displayed the Budget Variance details report</td></tr>
<tr><td>Business rules</td><td colspan="2">Only The Business Owner can generate this report</td></tr>
</table>

Implementation constraints and specifications	• The user requires internet access
Assumptions:	• • None
Open issues:	• None

Table 109: 7.6 Generate Budget Variance Report

USE CASE 7.7

System Name: Procion System Sub-System: 7		
Author: Leon Combrinck	Date: 05 August 2023	Version: 1.0

Use case name:	View Generate Report	**USE CASE TYPE**
Use case id:	7.7	Business Requirements: ☐
Priority:	High	System Analysis: ☒
Source:	Moyo	System Design: ☐
Primary business actor	User	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	

Description:	The use case describes the event where a user would like to view generate Report on the system. The user will then request the system to view the Generate report on the system. The system will then load all Generate Reports that are currently on the system and them display all the generate report options to the user.	
Pre-condition:	The user has to be logged in to the system.	
Trigger:	The user requests to view generate report details	
Typical course Of events:	**Actors Action**	**Systems response**
	Step 1: The user request to view generate report on the system	**Step 2:** The system navigates to and displays the view generate report on the system to the user
Alternate courses:	**None**	
Conclusion:	The user is displayed the view generate report details	
Post-condition:	The user is displayed the view generate report details	
Business rules	• none	
Implementation constraints and specifications	• The user requires internet access	
Assumptions:	• None	
Open issues:	• None	

Table 110: 7.7 View Generate Report

2.3 CONCLUSION

This concludes the Logical Use case Narratives section of the Procion system.

3. LOGICAL CONTEXT DIAGRAM

3.1 INTRODUCTION

In this section we will display the visual representation of the logical Context diagrams for the following use cases: 2.26, 2.37, 4.10, 5.9, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7 of the Procion system.

3.2 LOGICAL CONTEXT DIAGRAMS

USE CASE 2.26

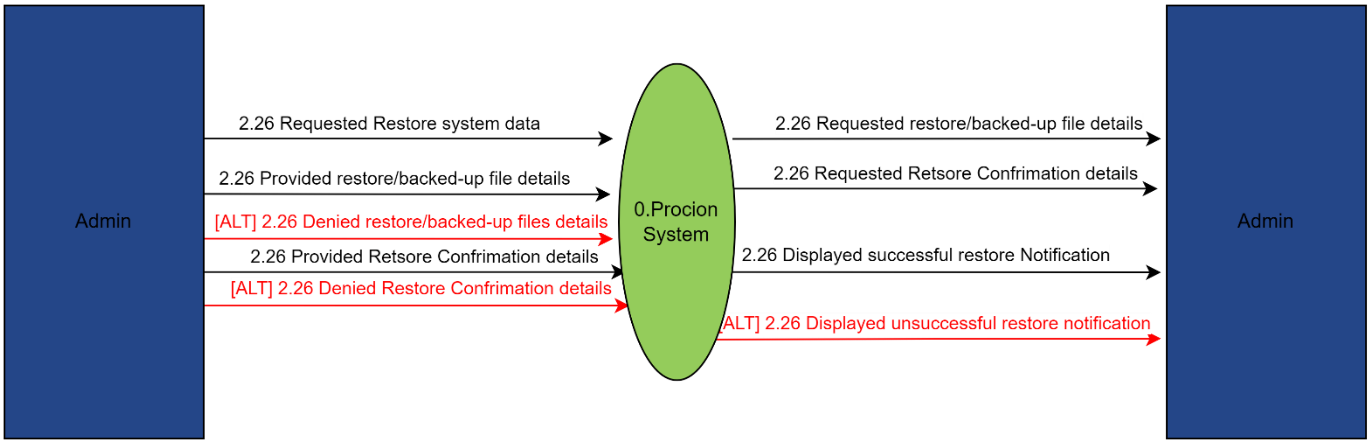


Figure 11: Use Case 2.26 Logical Context

USE CASE 2.37

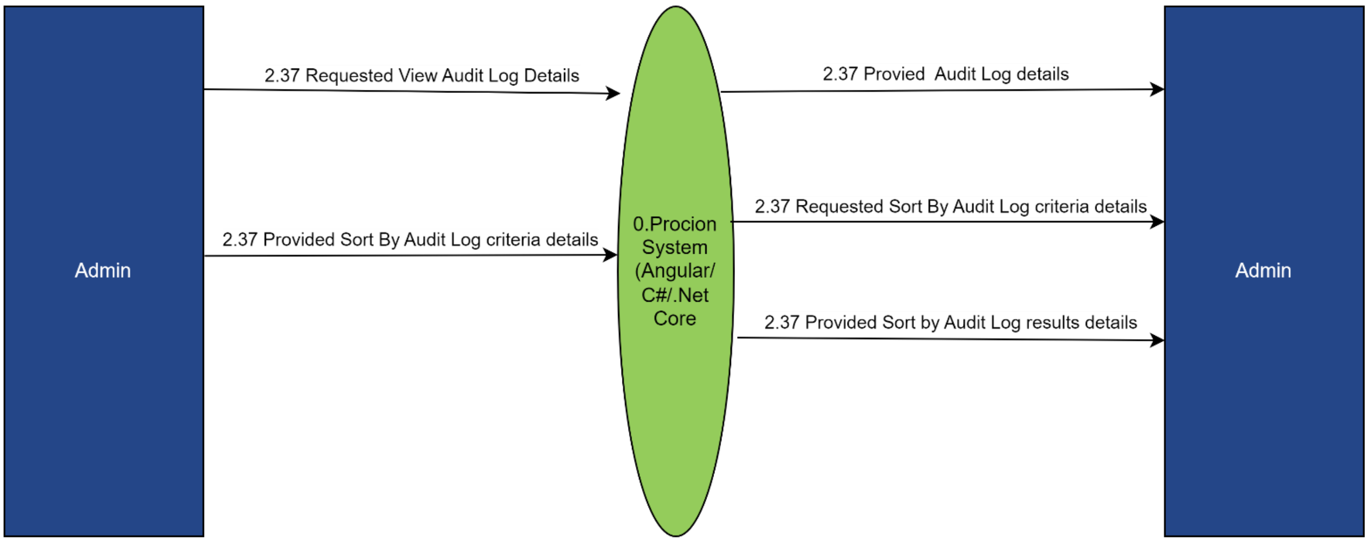


Figure 12: Use Case 2.37 Logical Context

USE CASE 4.10

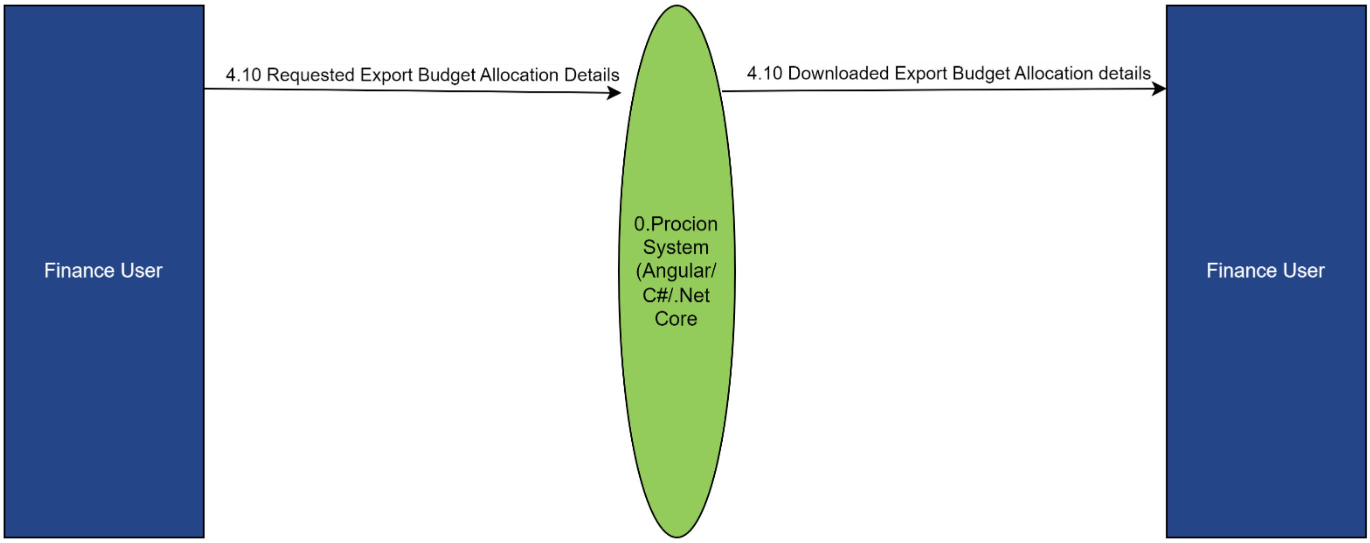


Figure 13: Use Case 4.10 Logical Context

USE CASE 5.9

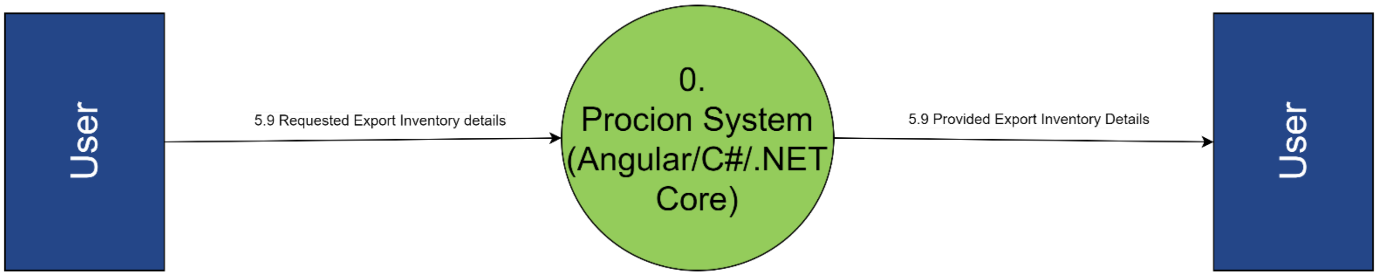


Figure 14: Use Case 5.9 Logical Context

USE CASE 7.1-7.7

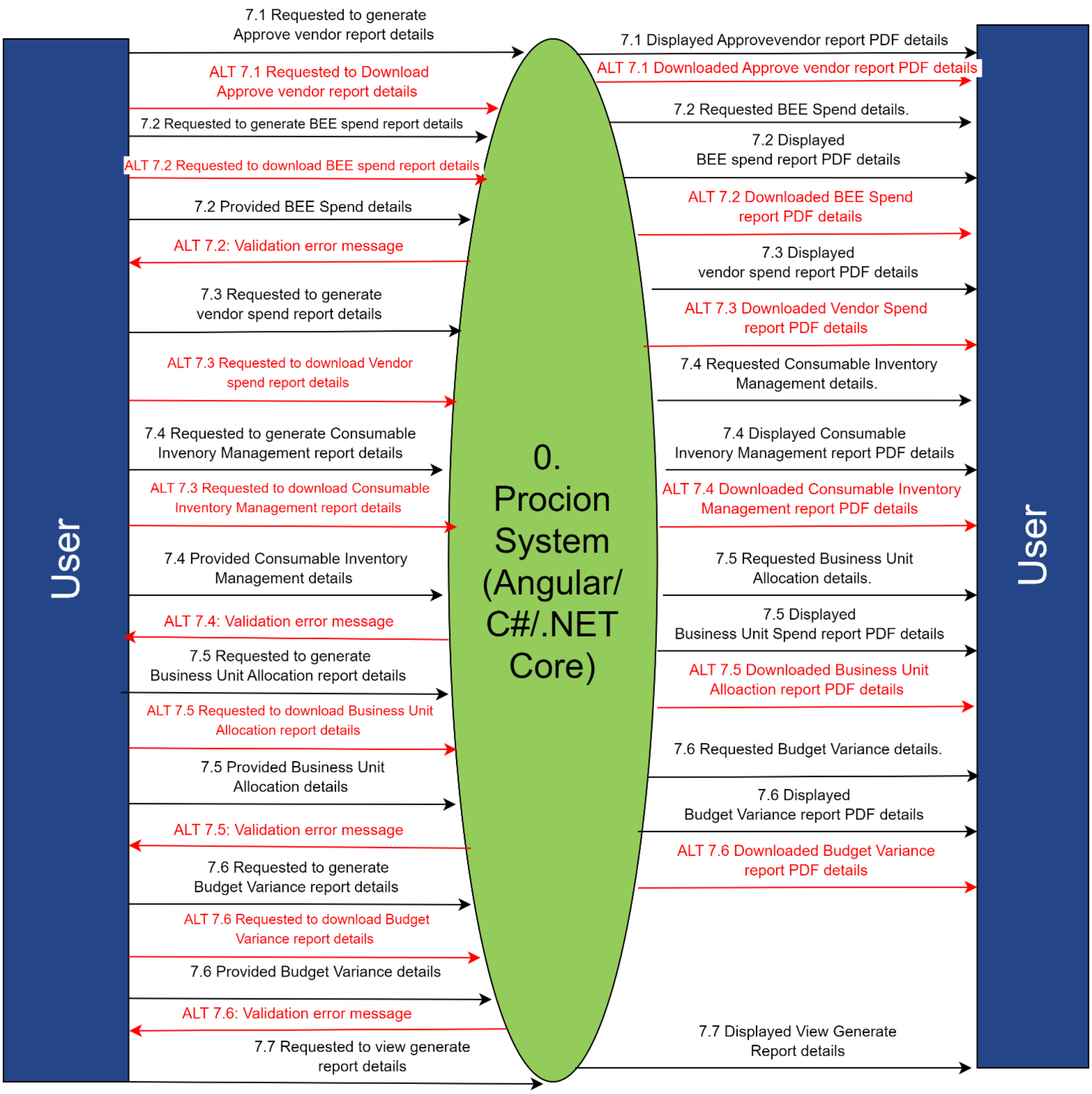


Figure 15: Use Case 7.1-7.7 Logical Context

3.3 CONCLUSION

This Concludes the visual representation of the Logical Context diagrams for the Procion system.

4. SCREEN DESIGN

4.1 INTRODUCTION

This section contains the design principles for the example input and output designs including Cruds from the following use cases: 2.26,2.37, 4.10, 5.9, 7.1-7.7 the navigation bar to depict the principles that will be used for all the screens in the Procion System.

4.2 DESIGN PRINCIPLES & EXAMPLE WIREFRAMES

USE CASE 2.26

2.25-2.26 Settings side-nav to Backup and Restore

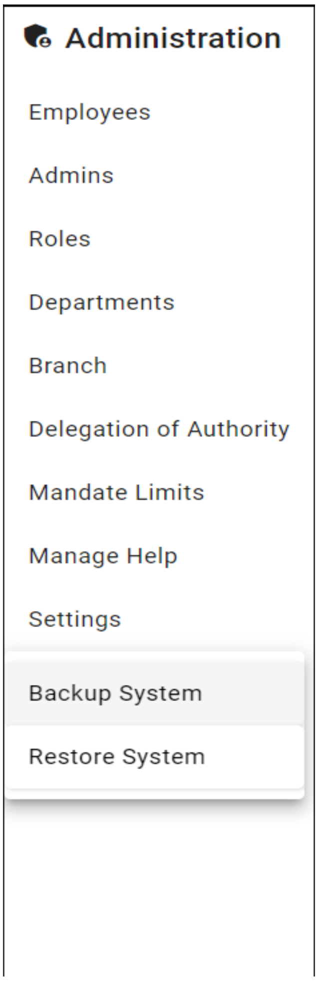


Figure 16: Settings dropdown menu for Backup and Restore

2.26 When the “Restore” button is clicked on Settings dropdown:

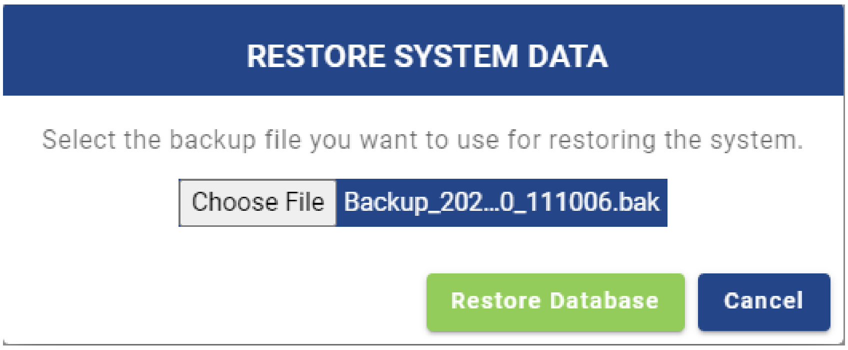


Figure 17: Restore Screen

2.26 When the Restore is Successful:

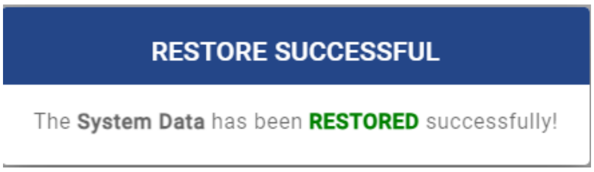


Figure 18: Restore Successful message

2.26 When the Restore is Successful loader:

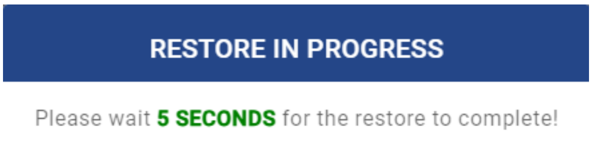


Figure 19: Restore load Successful message

2.26 When the Restore is Unsuccessful:

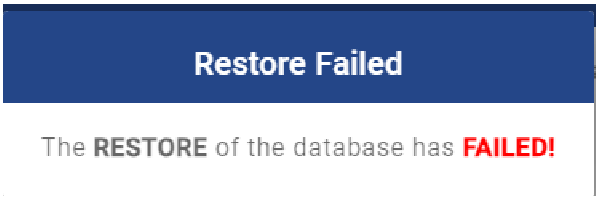
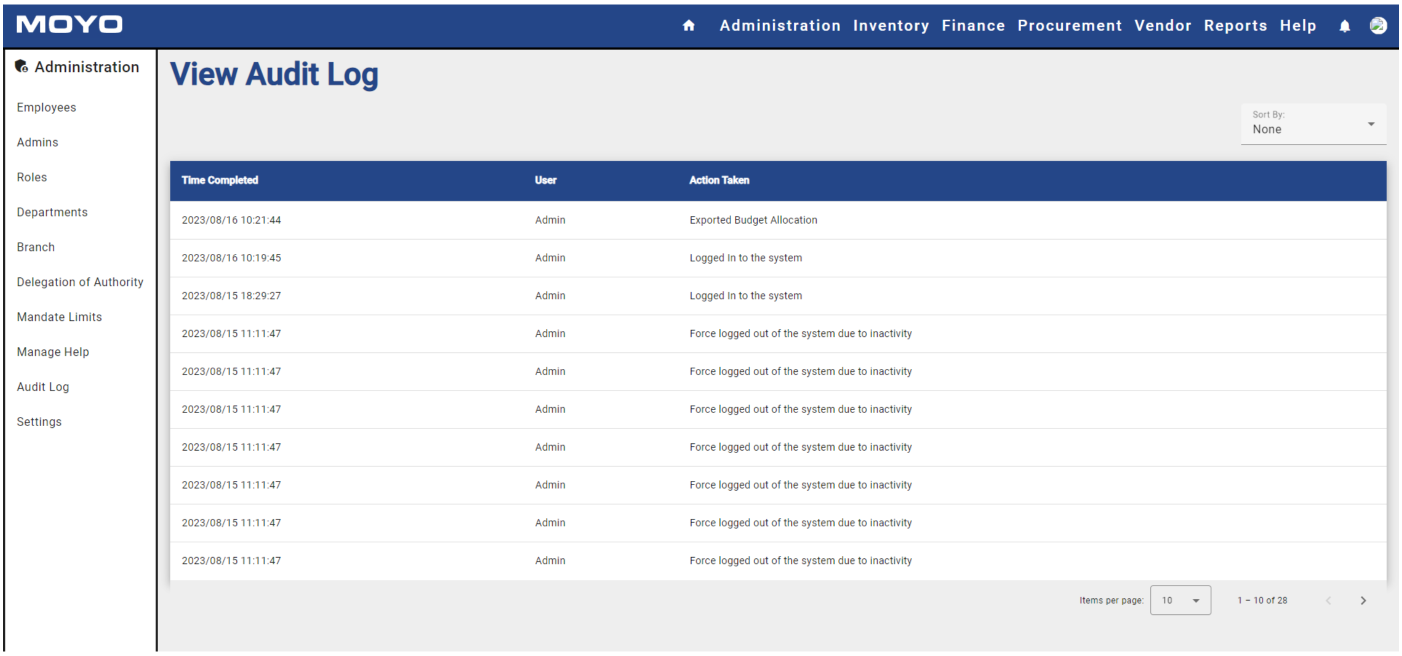


Figure 20: Restore Unsuccessful message

USE CASE 2.37

2.37 View Audit log

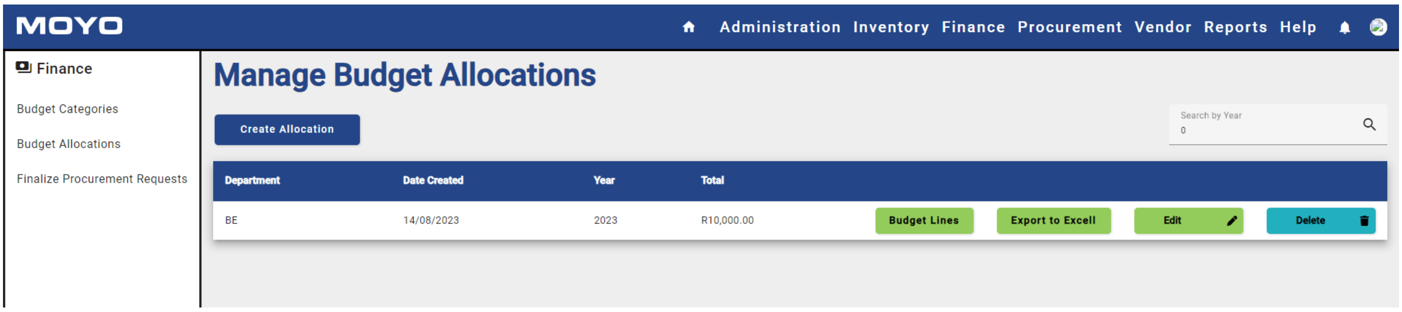


Time Completed	User	Action Taken
2023/08/16 10:21:44	Admin	Exported Budget Allocation
2023/08/16 10:19:45	Admin	Logged In to the system
2023/08/15 18:29:27	Admin	Logged In to the system
2023/08/15 11:11:47	Admin	Force logged out of the system due to inactivity
2023/08/15 11:11:47	Admin	Force logged out of the system due to inactivity
2023/08/15 11:11:47	Admin	Force logged out of the system due to inactivity
2023/08/15 11:11:47	Admin	Force logged out of the system due to inactivity
2023/08/15 11:11:47	Admin	Force logged out of the system due to inactivity
2023/08/15 11:11:47	Admin	Force logged out of the system due to inactivity
2023/08/15 11:11:47	Admin	Force logged out of the system due to inactivity

Figure 21: View Audit Log

USE CASE 4.10

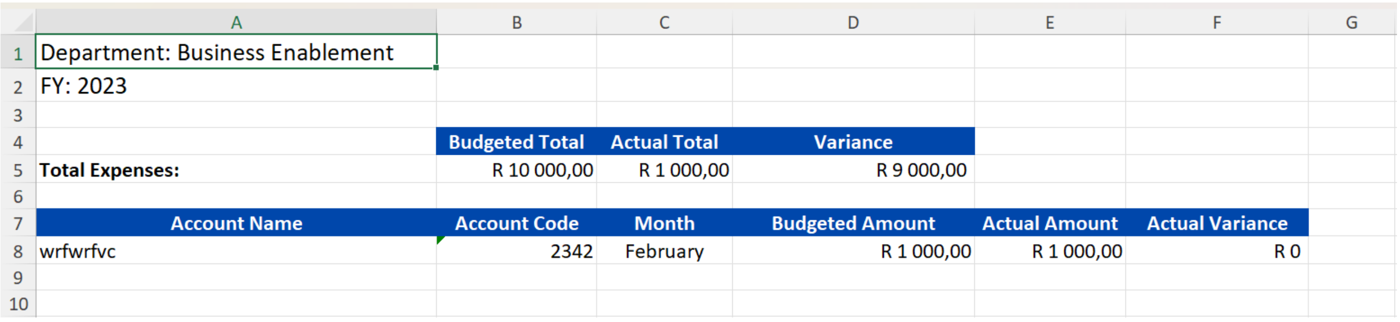
4.10 Export Budegt Alloaction



Department	Date Created	Year	Total
BE	14/08/2023	2023	R10,000.00

Figure 22: Export budget Allocation

4.10 Export Budegt Alloaction Excel File



	A	B	C	D	E	F	G
1	Department: Business Enablement						
2	FY: 2023						
3							
4		Budgeted Total	Actual Total	Variance			
5	**Total Expenses:**	R 10 000,00	R 1 000,00	R 9 000,00			
6							
7	Account Name	Account Code	Month	Budgeted Amount	Actual Amount	Actual Variance	
8	wrfwrfvc	2342	February	R 1 000,00	R 1 000,00	R 0	
9							
10							

Figure 23: Export budget Allocation Excel File

USE CASE 5.9

5.9 View Consumable Screen with Export Details Button

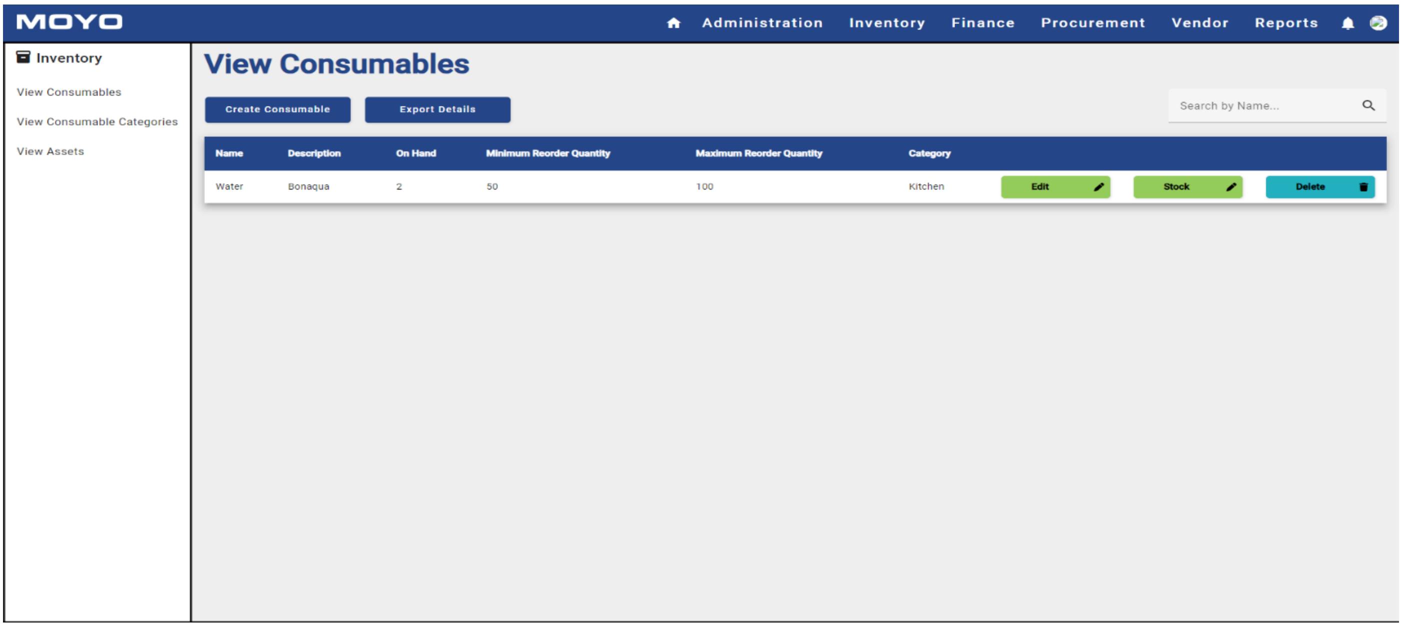
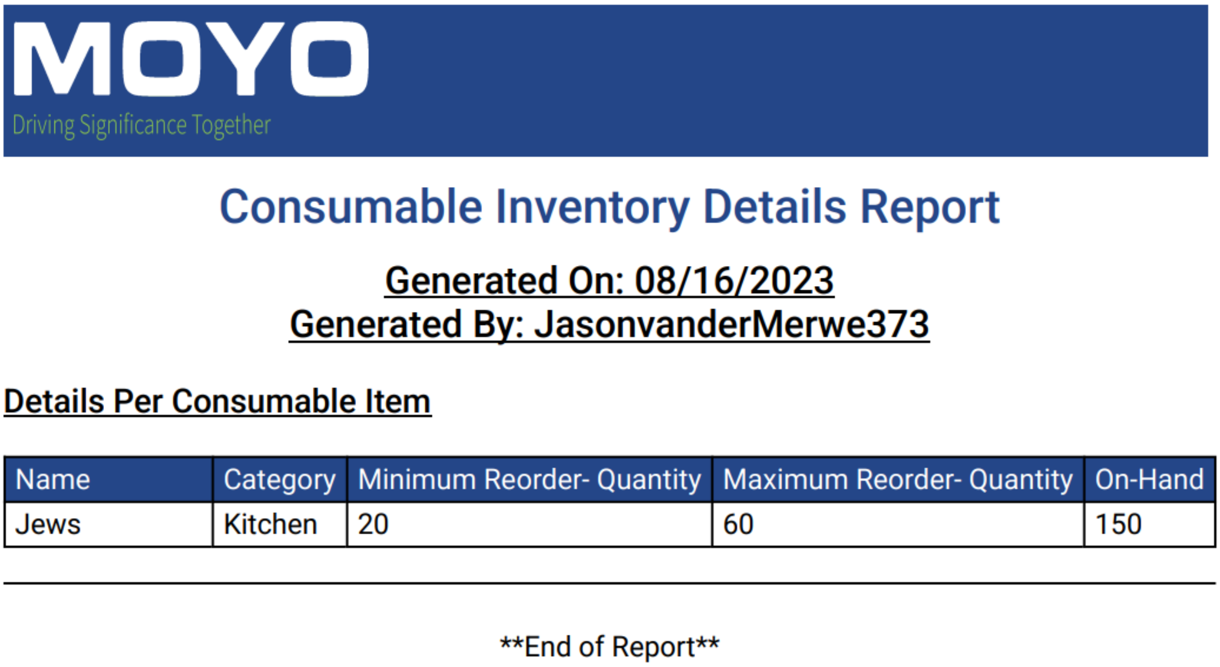


Figure 24 View Consumable Screen with Export Details Button

5.9 Export Details PDF



Driving Significance Together

Consumable Inventory Details Report

Generated On: 08/16/2023
Generated By: JasonvanderMerwe373

Details Per Consumable Item

Name	Category	Minimum Reorder- Quantity	Maximum Reorder- Quantity	On-Hand
Jews	Kitchen	20	60	150

End of Report

Figure 25 Export Details PDF

USE CASE 7.1-7.7

7.7 View Generate Report Screen with 'Generate?" Buttons:

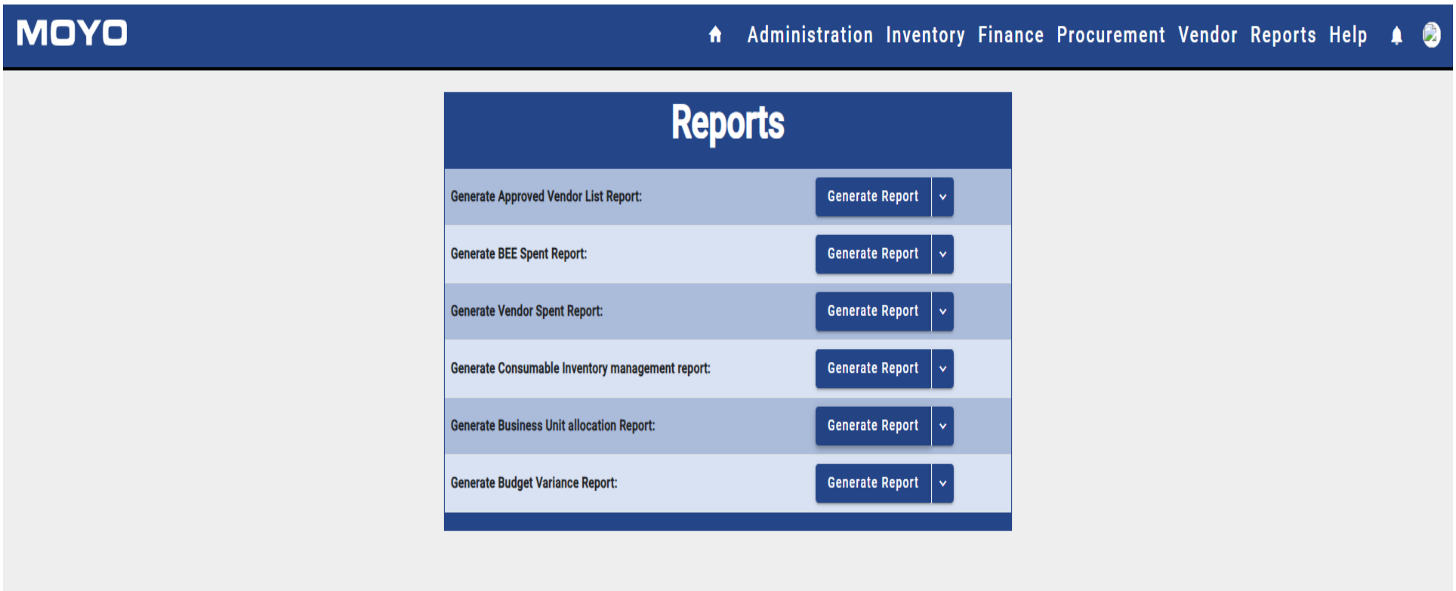


Figure 26 View Generate Report Screen with "Generate?" Buttons

7.1 Generate Approved Vendor Report PDF

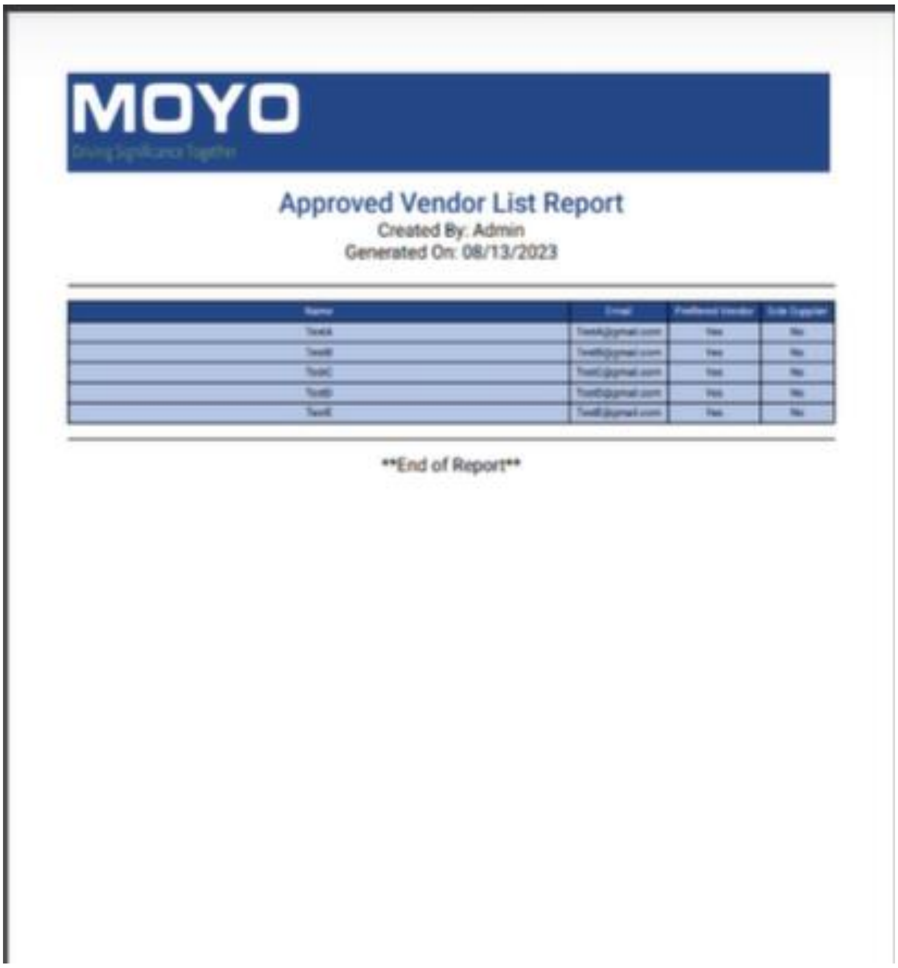


Figure 27 Generate Approved Vendor Report PDF

7.2 BEE Spend Screen

Figure 28 BEE Spend Screen

7.2 Generate BEE Spend Report PDF

MOYO
Driving Significance Together

BEE Spent Report

Created By: Admin
Generated On: 08/13/2023
Period: 08/01/2023 - 08/13/2023

Overall BEE Spend Summary

BEE Level	1	2	3	4	5	6	7	8	Non-Compliant	Total
Procurement Spend	6000.00	5000.00	4000.00	3000.00	0.00	0.00	0.00	0.00	2000.00	20000.00
Procurement %	135%	125%	110%	100%	80%	60%	50%	10%		
Procurement Entitlement	8100.00	6250.00	4400.00	3000.00	0.00	0.00	0.00	0.00	2000.00	23750.00

Pretoria BEE Spend Summary

BEE Level	1	2	3	4	5	6	7	8	Non-Compliant	Total
Procurement Spend	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2000.00	2000.00
Procurement %	135%	125%	110%	100%	80%	60%	50%	10%		
Procurement Entitlement	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2000.00	2000.00

MBA BEE Spend Summary

BEE Level	1	2	3	4	5	6	7	8	Non-Compliant	Total
Procurement Spend	6000.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6000.00
Procurement %	135%	125%	110%	100%	80%	60%	50%	10%		
Procurement Entitlement	8100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8100.00

CPT BEE Spend Summary

BEE Level	1	2	3	4	5	6	7	8	Non-Compliant	Total
Procurement Spend	0.00	5000.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5000.00
Procurement %	135%	125%	110%	100%	80%	60%	50%	10%		
Procurement Entitlement	0.00	6250.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6250.00

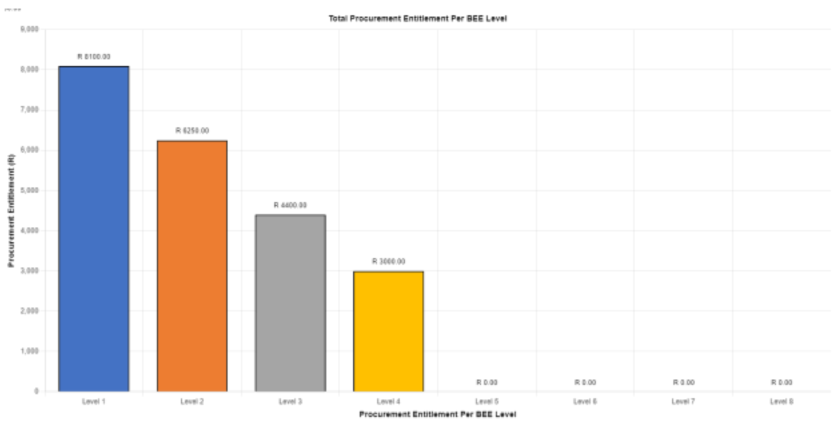
VEN BEE Spend Summary

BEE Level	1	2	3	4	5	6	7	8	Non-Compliant	Total
Procurement Spend	0.00	0.00	4000.00	0.00	0.00	0.00	0.00	0.00	0.00	4000.00
Procurement %	135%	125%	110%	100%	80%	60%	50%	10%		
Procurement Entitlement	0.00	0.00	4400.00	0.00	0.00	0.00	0.00	0.00	0.00	4400.00

ENG BEE Spend Summary

BEE Level	1	2	3	4	5	6	7	8	Non-Compliant	Total
Procurement Spend	0.00	0.00	0.00	3000.00	0.00	0.00	0.00	0.00	0.00	3000.00
Procurement %	135%	125%	110%	100%	80%	60%	50%	10%		
Procurement Entitlement	0.00	0.00	0.00	3000.00	0.00	0.00	0.00	0.00	0.00	3000.00

Column Chart for Total Procurement Entitlement Per BEE Level



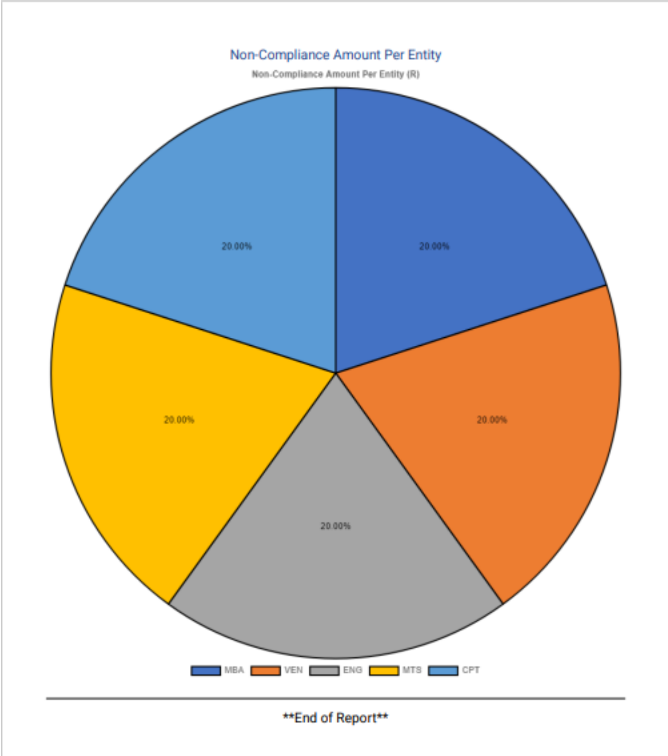


Figure 29 Generate BEE Spend Report PDF

7.3 Generate Vendor Spend Report PDF

MOYO
Driving Significance Together

Vendor Spent Report

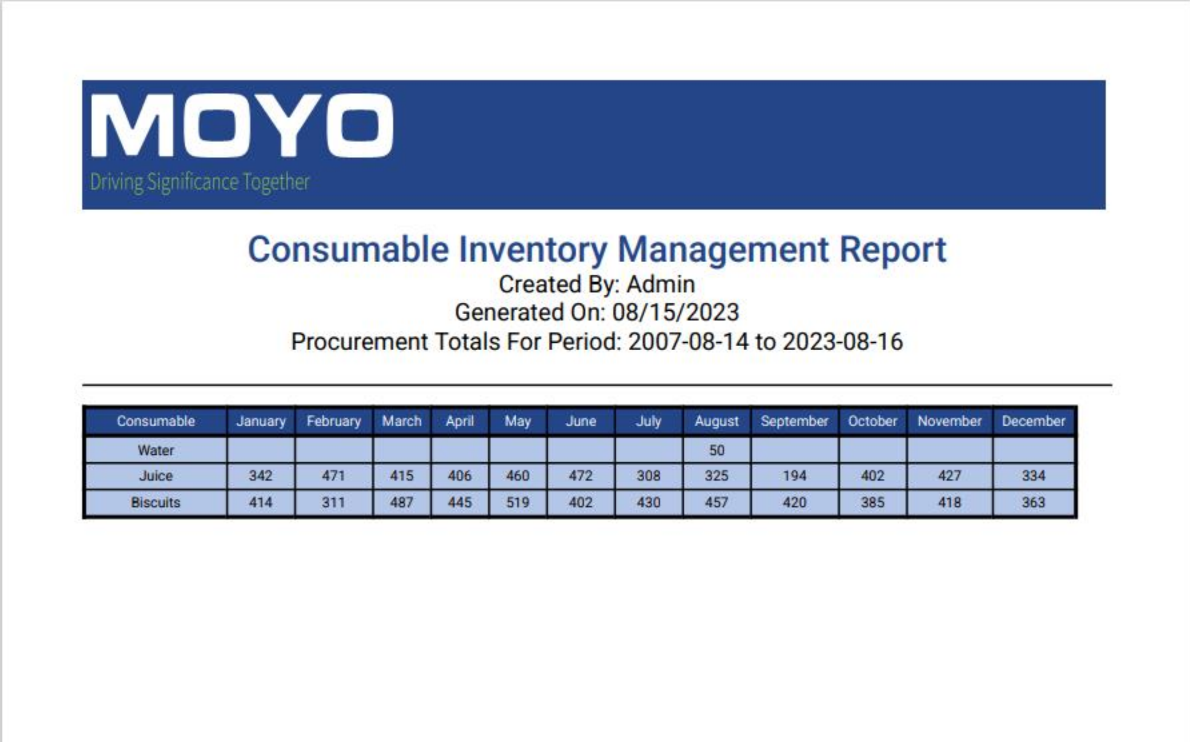
Created By: Admin
Generated On: 08/13/2023

Supplier	Account Code	Account Name (Budget Line)	Budget (Department)	Company (Branch)	BEE Level	Amount Spend
vaf	4000	Test	BE	Pretoria	0	R2,000.00
Total Amount for the Pretoria						R2,000.00
TestA	4000	Test	BE	MBA	1	R6,000.00
Total Amount for the MBA						R6,000.00
TestB	4000	Test	BE	CPT	2	R5,000.00
Total Amount for the CPT						R5,000.00
TestC	4000	Test	BE	VEN	3	R4,000.00
Total Amount for the VEN						R4,000.00
TestD	4000	Test	BE	ENG	4	R3,000.00
Total Amount for the ENG						R3,000.00

End of Report

Figure 30 Generate Vendor Spend Report PDF

7.4 Generate Consumable Inventory Management Report PDF



MOYO
Driving Significance Together

Consumable Inventory Management Report

Created By: Admin
Generated On: 08/15/2023
Procurement Totals For Period: 2007-08-14 to 2023-08-16

Consumable	January	February	March	April	May	June	July	August	September	October	November	December
Water								50				
Juice	342	471	415	406	460	472	308	325	194	402	427	334
Biscuits	414	311	487	445	519	402	430	457	420	385	418	363

Figure 31 Generate Consumable Inventory Management Report PDF

7.5 Generate Business Unit Allocation Report PDF

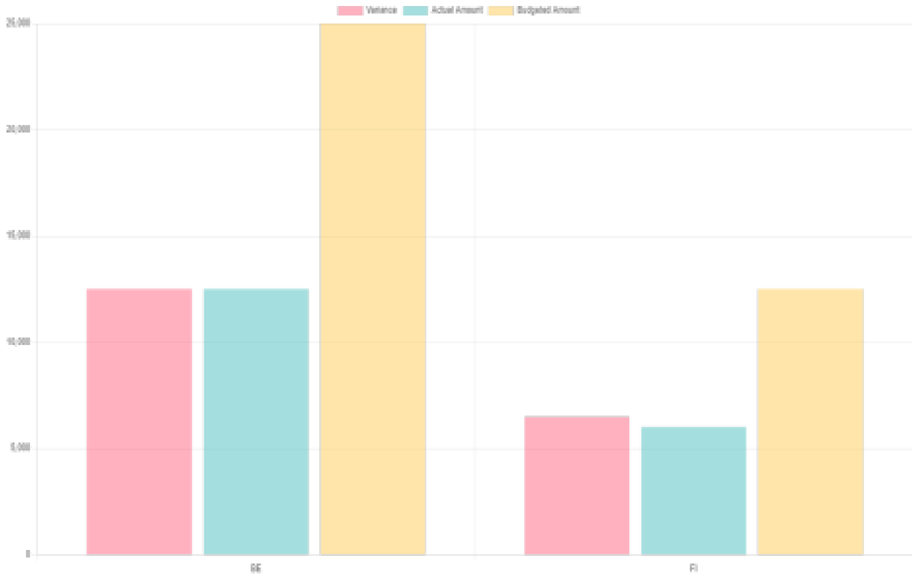
MOYO
Driving Significance Together

Business Unit Allocation Report

Created By: Admin
Generated On: 08/15/2023

Department	Account Code	Actual Amount	Budgeted Amount	Variance
BE	0909	R5,000.00	R10,000.00	R5,000.00
BE	2134	R5,000.00	R10,000.00	R5,000.00
BE	4444	R2,500.00	R5,000.00	R2,500.00
Total Amount for BE		R12,500.00	R25,000.00	R12,500.00
FI	1011	R5,000.00	R10,000.00	R5,000.00
FI	1111	R1,000.00	R2,500.00	R1,500.00
Total Amount for FI		R6,000.00	R12,500.00	R6,500.00

Bar Chart Showing Business Unit Allocations



End of Report

Figure 32 Generate Business Unit Allocation Report PDF

7.6 Generate Budget Variance Report PDF

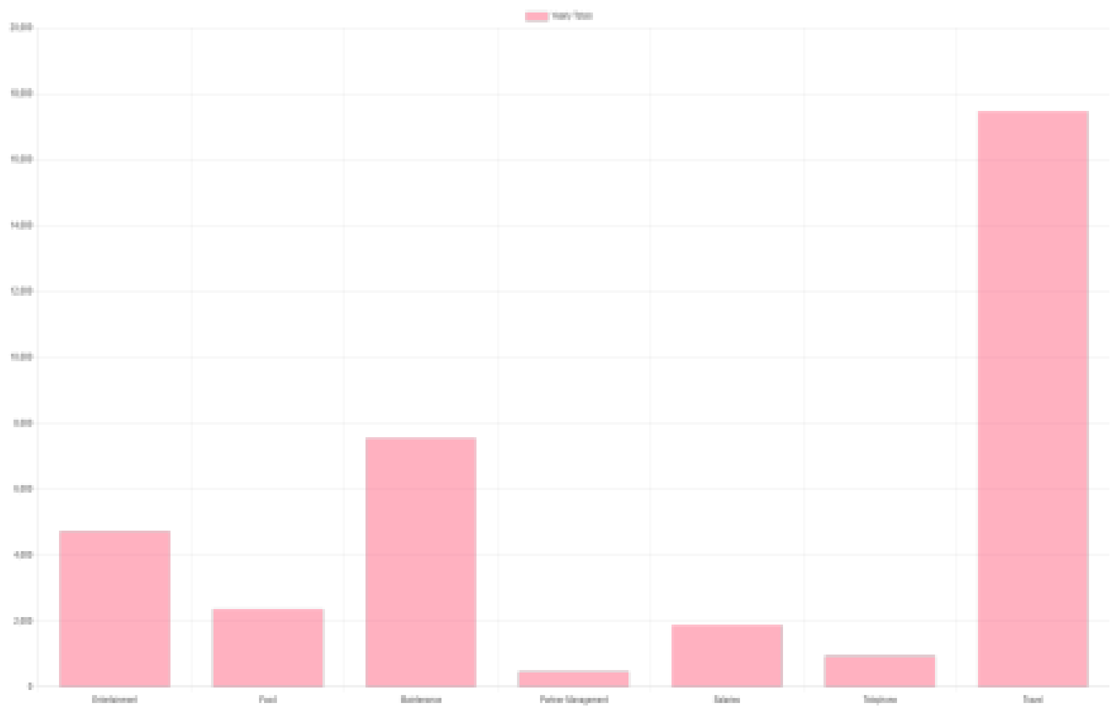
MOYO
Driving Significance Together

Budget Variance Report

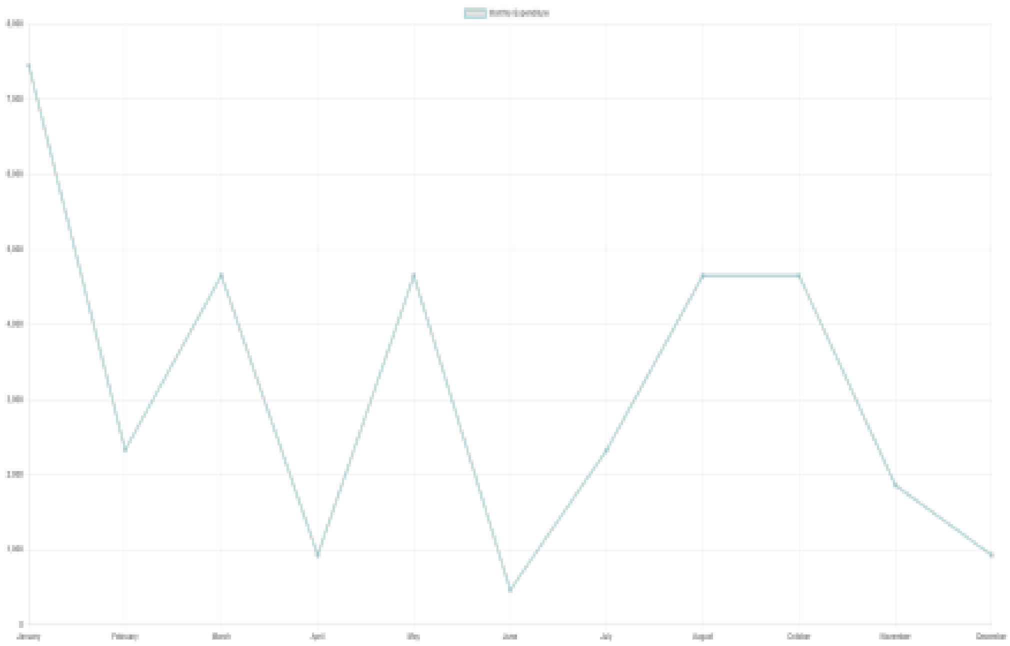
Created By: Admin
Generated On: 08/15/2023

Month	Budget Category	Budget Amount	Actual Amount	Variance	Variance (%)
March	Travel	R10,000.00	R5,000.00	R5,000.00	50.00%
May	Travel	R10,000.00	R5,000.00	R5,000.00	50.00%
July	Travel	R5,000.00	R2,500.00	R2,500.00	50.00%
October	Travel	R10,000.00	R5,000.00	R5,000.00	50.00%
December	Travel	R2,500.00	R1,000.00	R1,500.00	60.00%
Total Amount for Travel		R18,500.00	R37,500.00	R19,000.00	
January	Maintenance	R12,000.00	R8,000.00	R4,000.00	33.33%
Total Amount for Maintenance		R12,000.00	R8,000.00	R4,000.00	
February	Food	R5,000.00	R2,500.00	R2,500.00	50.00%
Total Amount for Food		R5,000.00	R2,500.00	R2,500.00	
April	Telephone	R2,500.00	R1,000.00	R1,500.00	60.00%
Total Amount for Telephone		R2,500.00	R1,000.00	R1,500.00	
June	Partner Management	R1,000.00	R500.00	R500.00	50.00%
Total Amount for Partner Management		R1,000.00	R500.00	R500.00	
November	Salaries	R4,000.00	R2,000.00	R2,000.00	50.00%
Total Amount for Salaries		R4,000.00	R2,000.00	R2,000.00	
August	Entertainment	R15,000.00	R5,000.00	R10,000.00	66.67%
Total Amount for Entertainment		R15,000.00	R5,000.00	R10,000.00	

Bar Chart Showing Total Expenses For Different Categories



Line Chart Showing Monthly Expenditure



End of Report

Figure 33 Generate Budget Variance Report PDF

7.7 View Generate Report Screen

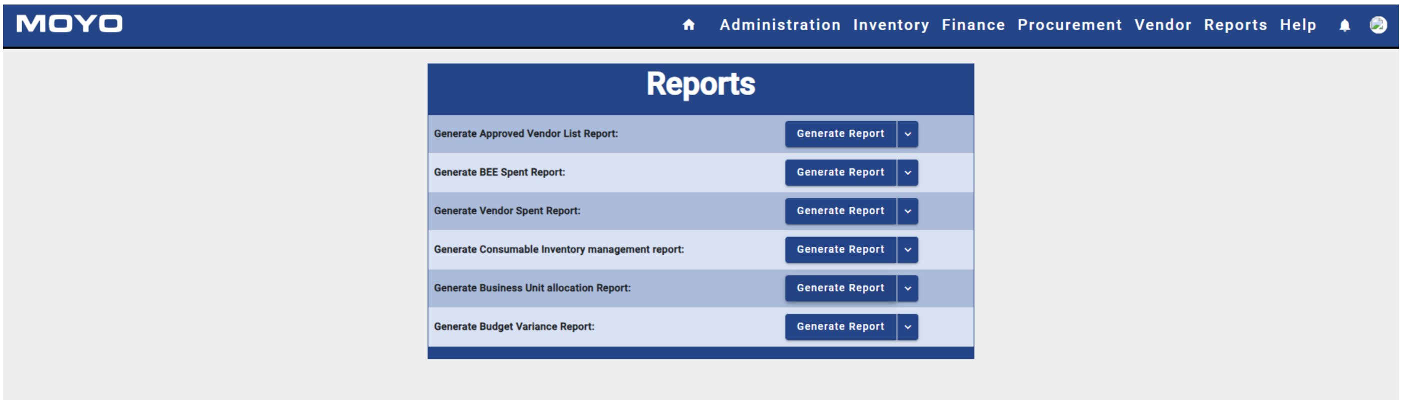


Figure 34 View Generate Report Screen

7.6 View Generate Report screen with Download and generate options:

Figure 35 View Generate Report screen with download and generate options

4.3 CONCLUSION

In conclusion this section provided an overview of how the inputs and outputs screen designs look and what principles need to be followed to achieve the desired screen designs of the Procion system.

5. TECHNICAL USE CASE NARRATIVES

5.1 INTRODUCTION

In this section we will discuss the Technical Use Case narratives for the following use cases: 2.26, 2.37, 4.10, 5.9, 7.1-7.7 of the Procion system.

5.2 TECHNICAL USE CASE NARRATIVES

USE CASE 2.26

System Name: Procion System Sub-System: 2 Administration		
Author: Leon Combrinck	Date: 2023/07/10	Version: 1.0

Use case name:	Restore data to system		**USE CASE TYPE**
Use case id:	2.26		Business Requirements: ☐
Priority:	high		System Analysis: ☐
Source:	Moyo		System Design: ☒
Primary business actor	Admin (PBA)		
Primary system actor	None		
Other participating actors:	None		
Other interested stakeholders:	None		
Description:			
Pre-condition:	The admin must be logged into the system.		
Typical course	**Actor Action**	**System Response**	

Of events:		Manual Action	Automated Action
	Step 1: The Admin navigates from the home screen to the Help screen by clicking on the **Administration** button on the top user navbar and then in the side Administration navbar clicking on the **Settings** button and then a Dropdown with two options: Backup, Restore. The Admin Clicks on the **Restore** Button.		**Step 2**: The system will generate the "*Restore System Data*" screen with the angular frontend that contains an angular material dialog with the following details: • Header: Restore System data • Label with text: "Select the backup file you want to use for restoring the system data." • input of type file with the following text (the default text of "no file chosen" called Restore file • Buttons: Restore Database, Cancel the system will make use of the API and angular frontend to make use of an uploadRestoreFile() function that will allow the Admin to upload the backed-up file form the previously created backups onto the system. The system will request the Admin to upload the backed-up file.
	Step 3: The admin select the restore/backed-up file		**Step 4:** The system will validate the input fields that was entered. - That the "Restore:" must not be null and

	they which to restore the system with.		files may not contain the same name. The system will prompt the Admin to Restore the Database **[ALT]**
	Step 5: The Admin Clicks on the “Restore Database” Button **[ALT]**		**Step 6:** The angular frontend will then call the Restore() function which will communicate with the appropriate header in the API by means of an HTTPPUT request and it will send it the Restore Object. The API will then receive the Restore object and Restore the backed-up the system data to the state of the backed-up file that is used.
			Step 7: The system will generate a model by making use of the .ASP NET core API Help object and the angular for the MatDialog with the following details: • Header: “Restore System Successful” • Label with text: “The System has been Restored successfully!” The dialog will pop up for 1.75 seconds.

<table>
<tr><td rowspan="2"></td><td></td><td></td><td>Thereafter the user will be routed to "Home" page.

[ALT]</td></tr>
<tr><td></td><td></td><td>Step 8: The system will navigate to the Settings tab on the Administration side navbar with the two dropdowns: backup, restore by making use of Angular routes with system restore data restore to the backed up file chosen by the Admin.</td></tr>
<tr><td rowspan="3">Alternate courses:</td><td colspan="3">[ALT] Step 4: The validation failed since no file was found. The system generates a Alert to the Admin with the following details: "Please select a backup file to restore the database." Return to step 3.</td></tr>
<tr><td colspan="3">[ALT] Step 5: The admin clicks on the "Cancel" Button. The admin will be redirected to the screen he/she was on previously. The use case ends.</td></tr>
<tr><td colspan="3">ALT-Step 7: The angular frontend receives the object instead of the null value. A Error notification will be generated which has the following Details:
• Header: ERROR: Restore Failed
• Body: Label: The Restore FAILED and the system data could not be restored!

Return to Step 1.</td></tr>
<tr><td>Conclusion:</td><td colspan="3">The system will navigate to the Home screen with the system data retore to chosen backed up state.</td></tr>
<tr><td>Post-condition:</td><td colspan="3">The system data is restored.</td></tr>
<tr><td>Business rules</td><td colspan="3">Only the admin can restore data to the system.</td></tr>
</table>

Implementation constraints and specifications	• None
Assumptions:	• None
Open issues:	• None

Table 111: 2.26 Restore System Data

USE CASE 2.37

System Name: Procion System Sub-System: 2 Administration		
Author: Leon Combrinck	Date: 2023/08/08	Version: 1.0

Use case name:	View Audit Log	**USE CASE TYPE**
Use case id:	2.37	Business Requirements: ☐
Priority:	high	System Analysis: ☐
Source:	Moyo	System Design: ☒
Primary business actor	Admin (PBA)	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	
Description:	In this use case the admin would like to view Audit Log on the system. The admin will request the system to view the Audit log on the system. The system will then load the Audit Log that are currently on the system and display them on the **AuditLog** table with the following details: • Time Completed • User	

<table>
<tr><td></td><td colspan="3">

- Action Taken

The system will then prompt the admin to enter the Sort By Criteria and at the Sort By Dropdwon. The admin will then enter the Sort By critria at the Sort By Dowpdown. The system will then Sort the list by Audit Log based on the admins Sort By criteria. The system will then display the list of the Sort By criteria.</td></tr>
<tr><td>Pre-condition:</td><td colspan="3">The admin must be logged into the system.

The admin has the authority to view the “Audit log” screen.

The system loaded the Administration side navbar.</td></tr>
<tr><td>Trigger:</td><td colspan="3">Admin requests to view Audit log on the system.</td></tr>
<tr><td rowspan="4">Typical course

Of events:</td><td rowspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The admin navigates from the home screen to the Audit log screen by clicking on the Administration button on the top <u>Admin</u> navbar and then in the side <u>Administration</u> navbar clicking on the Audit Log button.</td><td></td><td>Step 2: The system will call the GetLogs() function then make use of the Angular frontend of the AuditLog data by making a [HTTPGET] request to the .net core API backend. The backend uses a SQL read query to retrieve the Audit Log data from the database.

The system will thus retrieve from that Audit Log table in the database the following information:
Time Completed
User
Action Taken</td></tr>
<tr><td></td><td></td><td>Step 3: The system will display the “View Audit Log” screen with the following details:
Header: Label containing text: “View Audit Log”</td></tr>
</table>

			Sort By Dropwdown List: Label Containing “Sort By” **Table Containing:** - Column headers: -Time Completed, -User, -Action taken. **Main Navbar:** (From left to right) - Moyo Logo (Just a logo) - Home Icon **Administration** - Inventory - Finance - Procurement - Vendor - Reports - Help - Notification Bell Icon - Profile Icon Dropdown **Side Navbar:** “Employee” tab “Admin” tab “Roles” tab “Department” tab “Branch” tab “Delegation of authority” tab “Mandate limits” tab “Manage Help” tab **“Audit Log” tab** “Settings” tab
			Step 4: The system will then populate the SortAuditLog array and

			the AuditLog array with the data retrieved from the database by means of the GetLogs() Function.. The angular material table will be populated with the data contained in the SortAuditLog array. The system will prompts the admin to enter Sort By criteria.
	Step 5: The admin provides Sort criteria with the following details. • User or Action **[ALT]**		**Step 6:** The system makes use of the LINQ LAMBDA function to populate the Audit Log array with the **AuditLog** table information. The system then populates the model array by using the API to the Angular frontend that populated the Audit Log service. This model array then corresponds with the Audit log typescripts in the Angular frontend.
Alternate courses:	**ALT Step 5:** The Admin does not enter any Sort By Criteria. **The Use Case Ends**		
Conclusion:	The system displays the Audit log list that matches with the admins Sort By criteria.		
Post-condition:	The matching **Audit Log** table fields are displayed.		
Business rules	Only the admin can View Audit Log to the system.		

Implementation constraints and specifications	• None
Assumptions:	• None
Open issues:	• None

Table 112: 2.37 View Audit log

USE CASE 4.10

System Name: Procion System Sub-System: 4		
Author: Leon Combrinck	Date: 8 August 2023	Version: 1.0.0

Use case name:	Export Budegt Allocaton	**USE CASE TYPE**
Use case id:	4.10	Business Requirements: ☐
Priority:	High	System Analysis: ☐
Source:	Moyo	System Design: ☒
Primary business actor	Finance User	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	
Description:	The use case describes the event where a user will want to Export The Budget Alloaction which contains all of the details for Budegt Allocation and Budegt lines items stored on the system. The user will click on the “Export To Excel” button located on the “View Budegt Allocation” screen. The system will get all of the Budget Allocations and Budegt Lines and Budget Category in the database. The Excel File will be generated and Downloaded to a Finance users Device. The use case ends	

<table>
<tr><td>Pre-condition:</td><td colspan="3">The Finance user has to be logged in to the system.</td></tr>
<tr><td>Trigger:</td><td colspan="3">The Finance user requests to view Budget Allocation on the system

The Finance User Then Clicks on the “Export To Excel” Button on the View Budget Allocation page.</td></tr>
<tr><td rowspan="3">Typical course

Of events:</td><td rowspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td></td><td></td><td>Step 2: The system calls the function GetBudgetAllocationExport() in the angular frontend that communicates to the appropriate header in the ASP.NET core API by means of an HTTPGET request. This function will get the following data from the <u>Budget Lines</u> entity by means of an <u>SQL Read Query:</u>

• Budget_ID
• Category_ID
• Account Code
• Month
• BudgetedAmt
• ActualAmt
• Variance

The system will also INCLUDE the following details from the <u>Budget_Category</u> Entity where the Category_ID in the <u>Budget_Lines</u> entity is equal to the Category_ID in the <u>Budget_Category</u> Entity:</td></tr>
</table>

			• Category_ID • Account Name • Description The system will also INCLUDE the following details from the **Budget_Allocation** Entity where the Budget_ID in the **Budget_Lines** entity is equal to the Budget_ID in the **Budget_Allocation** Entity: • Budget_ID • Department_ID • Date_Created • Year • Total The Angular frontend will receive this data in the following array • ExportToExcel[]

			Step 3: The angular frontend will generate a pdf using ClosedXML as follows: • **Header Cell 1**: Department: [Department] • **Cell 2**: FY [Year] • **Generated by**: [User] • **Cell 4:** Table with columns: -Budgeted Amount, -Actual Amount, -Variance

			• **Last Cell in table:** Total Expenses. • **Next Table Cell:**Table with columns: -Account Name, -Account Code, -month, -Budegted Amount, -Actual Amount, **-**Actual Variance. • This continues until all budegt lines for the Budget Allocation is covered. The Rows of the table will be populated using the data contained in the ExportToExcel[] array.
			Step 4: The Excel File is Downloaded to the Finance user Device.
Alternate courses:	**None**		
Conclusion:	The system Downloads the Export Budegt Allocation details Excel file to the Finance users device		
Post-condition:	The Excel File is generated and downloaded to the Finance Users Device		
Business rules	• None		
Implementation constraints and specifications	• None		
Assumptions:	• None		
Open issues:	• None		

Table 113: 4.10 Export Budget Alloaction

USE CASE 5.9

System Name: Procion System Sub-System:		
Author: Jason van der Merwe	Date: 13 May 2023	Version: 1.0.0

Use case name:	Export Inventory Details	**USE CASE TYPE**	
Use case id:	5.9	Business Requirements: ☐	
Priority:	High	System Analysis: ☐	
Source:	Moyo	System Design: ☒	
Primary business actor	User		
Primary system actor	None		
Other participating actors:	None		
Other interested stakeholders:	None		
Description:	The use case describes the event where a user will want to generate a report which contains all of the details for consumable items stored on the system. The user will click on the Export Details button located on the "View Consumables" screen. The system will get all of the consumables and consumable categories in the database. The pdf will be generated and displayed to a user in a new tab. The use case ends		
Pre-condition:	The user has to be logged in to the system.		
Trigger:	The user requests to view all of the Consumable items stored on the system in a report		
Typical course Of events:	**Actor Action**	**System Response**	
		Manual Action	**Automated Action**

	Step 1: The user requests to view all of the Consumable items stored on the system in a report. The User clicks on the Export Details button	**None**	**Step 2:** The system calls the function GetConsumables() in the angular frontend that communicates to the appropriate header in the ASP.NET core API by means of an HTTPGET request. This function will get the following data from the **Consumable** entity by means of an SQL Read Query: • Name • Description • On_Hand • Minimum_Reorder_Quantity • Maximum_Reorder_Quantity • Consumable_Category_ID The system will also INCLUDE the following details from the **CONSUMABLE_CATEGORY** Entity where the Consumable_Category_ID in the **Consumable** entity is equal to the Consumable_Category_ID in the **CONSUMABLE_CATEGORY** Entity: • Name • Description The Angular frontend will receive this data in the following array • Consumables[]

		None	**Step 3:** The angular frontend will generate a pdf using PDFMAKE as follows: • **Moyo Logo** • **Header**: Consumable Inventory details Report • **Generated on**: [Date] • **Generated by**: [User] • **Label:** “Details per consumable item” • **Table** with headers: Name, category. Minimum reorder quantity Maximum reorder quantity On hand • **Text:** **End of Report** The Rows of the table will be populated using the data contained in the Consumables[] array.
			Step 4: The PDF is displayed to the user in a new tab
Alternate courses:	**None**		
Conclusion:	The system displays the Inventory details report to the user		
Post-condition:	The report is generated containing the inventory details		
Business rules	• None		

Implementation constraints and specifications	• None
Assumptions:	• None
Open issues:	• None

Table 114: 5.9 Export Inventory Details

USE CASE 7.1

System Name: Procion System **Sub-System: 7**		
Author: Leon Combrinck	Date: 8 August 2023	Version: 1.0.0

Use case name:	Generate Approved Vendor Report	**USE CASE TYPE**
Use case id:	7.1	Business Requirements: ☐
Priority:	High	System Analysis: ☐
Source:	Moyo	System Design: ☒
Primary business actor	User	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	
Description:	The use case describes the event where a user will want to generate a report which contains all of the approved vendors stored on the system. The user will click on the Generate Approve Vendor Report button located on the “View Generate Report” screen. The system will get all of the Vendors and Vendor_Status in the database. The pdf will be generated and displayed to a user in a new tab. The use case ends	

Pre-condition:	The user has to be logged in to the system.		
Trigger:	The user requests to view Generate Report screen on the system in a report The user clicks on the approved vendor generate button. OR The User clicks on the approved vendor download button.		
Typical course **Of events:**	**Actor Action**	**System Response**	
		Manual Action	**Automated Action**
			Step 1: The system will send a request form the Angular frontend to the Report service where the service will make a http get request to the .NET Core backend which makes use of a Lambda Linq Query which creates a SQL Read query to retrieve all the items from the **<u>Vendor</u>** Entity, the **<u>Vendor_Status</u>** Entity The system calls the function <u>GetApprovedVendor()</u> in the angular frontend that communicates to the appropriate header in the ASP.NET core API by means of an HTTPGET request. This function will get the following data from the **<u>Vendor</u>** entity by means of an <u>SQL Read Query:</u> • Vendor_ID • Vendor_Status_ID

			• Name • Email • Number_Of_Times_Used • SoleSupplierProvided The system will also INCLUDE the following details from the **Vendor_Status** Entity where the Vendor_Status_ID in the **Vendor** entity is equal to the Vendor_Status_ID in the **Vendor_Status** Entity as a foreign key: • Vendor_Status_ID • Name • Description The Angular frontend will receive this data in the following array • ApprovedVendor[]

			Step 2: The angular frontend will generate a pdf using PDFMAKE as follows: • **Logo Header**: Moyo Logo • **Heading:** Approved Vendor List Report

			Label Generated By: [Role] Person we generated the report. **Label Generated on**:[Date] Date when the report was generated on **Table Tabel Column**: name, email, prefered vendor, sole supplier **Text**: **Report End** The Rows of the table will be populated using the data contained in the ApprovedVendor[] array.
			Step 3: The PDF is displayed to the user in a new tab **[ALT]**
Alternate courses:	**ALT Step 3:** The PDF is downloaded to the user's device. **Use Case Ends.**		
Conclusion:	The system displays the Approved Vendor report to the user		
Post-condition:	The report is generated containing the Approved Vendors details		
Business rules	• Only the GRC are allowed to generate this report		
Implementation constraints and specifications	• None		
Assumptions:	• None		

Open issues:	• None

Table 115: 7.1 Generate Approved Vendor Report

USE CASE 7.2

System Name: Procion System **Sub-System:**		
Author: Leon Combrinck	Date: 8 August 2023	Version: 1.0.0

Use case name:	Generate BEE Spend Report	**USE CASE TYPE**
Use case id:	7.2	Business Requirements: ☐
Priority:	High	System Analysis: ☐
Source:	Moyo	System Design: ☒
Primary business actor	User	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	
Description:	The use case describes the event where a user will want to generate a report which contains all of the BEE Spending stored on the system. The user will click on the Generate BEE Spend Report button located on the “View Generate Report” screen. The system will get all of the Procurement_Details Procurement_Request, Vendor, Employee, Branch, Budget_Lines, Budget_Category in the database. The system will Then display a screen modal popup where user will be able to enter Full_Payment_Due_Date[start and end date] filter criteria. The system will then prompt the user to generate the report. The User the clicks on the “Generate” button on the popup modal screen. The pdf will be generated and displayed to a user in a new tab. The use case ends	
Pre-condition:	The user has to be logged in to the system.	

<table>
<tr><td rowspan="1">Trigger:</td><td colspan="3">The user requests to view Generate Report screen on the system in a report

The user clicks on the BEE Spend Generate button.

OR The User clicks on the BEE Spend download button.</td></tr>
<tr><td rowspan="3">Typical course

Of events:</td><td rowspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td></td><td></td><td>Step 1: The system will send a request form the Angular frontend to the Report service where the service will make a http get request to the .NET Core backend which makes use of a Lambda Linq Query which creates a SQL Read query to retrieve all the items from the <u>Procurement_Details</u> Entity, the <u>Procurement_Request</u> Entity, the <u>Vendor</u> Entity, , the <u>Employee</u> Entity, , the <u>Branch</u> Entity, the <u>Budget_Lines</u> Entity, the <u>Budget_Category</u> Entity

The system calls the function <u>GetBEESpendReport()</u> in the angular frontend that communicates to the appropriate header in the ASP.NET core API by means of an HTTPGET request. This function will get the following data from the</td></tr>
</table>

			Procurement_Details entity by means of an SQL Read Query: • ProcurementDetail ID • Employee ID • ProcurementRequest ID • Sign Off Status • ProcurementPayment Status ID • Account Code • Procurement Status ID • PaymentMethod ID • ItemType • BuyerName • BuyerEmail • DepositRequest • FullPaymentDueDate • TotalAmount • PaymentMade • Comment • ProofOfPaymentRequired The system will also INCLUDE the following details from the **Procurement_Request** Entity where the Procurement_Request_ID in the **Procurement_Details** entity is equal to the Procurement_Request_ID in the **Procurement_Request** Entity as a foreign key: • Procurement_ID • Vendor_ID • Requistion_ID • User_ID

			• Name • Description The system will also THENINCLUDE the following details from the **<u>Vendor</u>** Entity where the Vendor _ID in the **<u>Procurement_Request</u>** entity is equal to the Vendor _ID in the **<u>Vendor</u>** Entity as a foreign key: • Vendor_ID • Vendor_Status_ID • Name • Email • NumberOfTimesUsed • SoleSupplierProvided The system will also INCLUDE the following details from the **<u>Employee</u>** Entity where the Employee_ID in the **<u>Procurement_Details</u>** entity is equal to the Employee_ID in the **<u>Employee</u>** Entity as a foreign key: • Employee_ID • User_ID • Department_ID • Branch_ID • Mandate_ID • EmployeeName • EmployeeSurname • CellPhone_Nr • Email

			The system will also THENINCLUDE the following details from the **Branch** Entity where the Branch _ID in the **Employee** entity is equal to the Branch _ID in the **Branch** Entity as a foreign key: • Branch_ID • Name • Street • City • Postal_Code • Province The system will also INCLUDE the following details from the **Budget_Line** Entity where the Account_Code in the **Procurement_Details** entity is equal to the Account_Code in the **Budget_Line** Entity as a foreign key: • Budget_ID • Category_ID • Account_Code • Month • BudgetAmt • ActualAmt • Variance The system will also THENINCLUDE the following details from the **Budget_Category** Entity where the Category _ID in the **Budget_Line** entity is equal to the Category _ID in the **Budget_Category**

			Entity as a primary key foreign key: • Category_ID • Account_Name • Description The system will also include a condition WHERE the following details: • Procurement_Status_ID = 2 • Full_Payment_Due_Date >= StartDate • Full_Payment_Due_Date <= EndDate The Angular frontend will receive this data in the following array • BEESpend[]

			Step 2: The system will display the BEE Spend popup model with the following information: • **Title:** Select filter Criteria. • **Text:** Please provide the following details. • **Input Field:** Calendar date picker[start-end] • **Button:** Generate, Cancel

			The System prompt the user to enter the required field details: • Full_Payment_Due_Date
	Step 3: The user Provides the following details: • Full_Payment_Due_Date		**Step 4:** The system validates the details that was enetered. -That the “Date Range” input box is not null, start date is not invalid, end date is not invalid. **[ALT]** The system prompts the user the click on the generate button.
	Step 5: The User clicks on the “Generate” Button **[ALT]**		**Step 6:** The angular frontend will generate a pdf using PDFMAKE as follows: • **Logo Header:** Moyo Logo • **Heading:** BEE Spend Report • **Labels:** Generated By-[User], Generated on: [Date], Period:[StartDate]-[EndDate] • **Tabel Heading:** [Branch] BEE Spend Summary • **Tabel Columns:** Bee Level, 1-8, Non-Complaint, Total • **Tabel Rows:** Procurement

			Spend, Procurement %, Procurement Entitlement. • This table is repeated until all Branch are covered • **Header:** Column Chart for Total Procurement Entitlement Per BEE Level. • **Bar Chart Header:** Total Procurement entitlement per BEE level. • **Bar Chart y-axis:** Procurement Entitlement. • **Bar Chart x-axis:** BEE Level. • **Header:** Non-Complaint amount per Entity. • **Pie Chart Header:** Non-Complaint Amount Per Entity (R) • **Pie Chart** • **Legend:** Color, Branch. (This repeats for all the Branches) • **Text:** **Report End** The Rows of the table will be populated using the data contained in the BEESpend[] array.

			Step 7: The PDF is displayed to the user in a new tab **[ALT]**

Alternate courses:	**ALT Step 4:** The field do not correspond to validation criteria. A red text under the input box will display explaining the validation error. Return to step 3. **ALT Step 5:** The user clicks on the “Cancel” Button. **The Use Case Ends.** **ALT Step 7:** The PDF is downloaded on the user’s device. **The Use Case Ends.**
Conclusion:	The system displays the BEE Spend report to the user
Post-condition:	The report is generated containing the BEE Spend details
Business rules	• Only the GRC are allowed to generate this report
Implementation constraints and specifications	• None
Assumptions:	• None
Open issues:	• None

Table 116: 7.2 Generate BEE Spend Report

USE CASE 7.3

System Name: Procion System **Sub-System:**		
Author: Leon Combrinck	Date: 8 August 2023	Version: 1.0.0

Use case name:	Generate Vendor Spend Report	**USE CASE TYPE**
Use case id:	7.3	Business Requirements: ☐
Priority:	High	System Analysis: ☐

<table>
<tr><td>Source:</td><td>Moyo</td><td colspan="2">System Design: ☒</td></tr>
<tr><td>Primary business actor</td><td colspan="3">User</td></tr>
<tr><td>Primary system actor</td><td colspan="3">None</td></tr>
<tr><td>Other participating actors:</td><td colspan="3">None</td></tr>
<tr><td>Other interested stakeholders:</td><td colspan="3">None</td></tr>
<tr><td>Description:</td><td colspan="3">The use case describes the event where a user will want to generate a report which contains all of the Vendor Spending stored on the system. The user will click on the Generate Vendor Spend Report button located on the “View Generate Report” screen. The system will get all of the Procurement_Details, Procurement_Request, Vendor, Employee, Department, Branch, Budget_Lines, Budget_Category Budget_Allocation in the database. The pdf will be generated and displayed to a user in a new tab. The use case ends</td></tr>
<tr><td>Pre-condition:</td><td colspan="3">The user has to be logged in to the system.</td></tr>
<tr><td>Trigger:</td><td colspan="3">The user requests to view Generate Report screen on the system in a report

The user clicks on the Vendor Spend Generate button.

OR The User clicks on the Vendor Spend download button.</td></tr>
<tr><td rowspan="3">Typical course

Of events:</td><td rowspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td></td><td></td><td>Step 1: The system will send a request form the Angular frontend to the Report service where the service will make a http get request to the .NET Core backend which makes use of a Lambda Linq Query which creates a SQL Read query to retrieve all the items from the</td></tr>
</table>

			<u>Procurement_Details</u> Entity, the **<u>Procurement_Request</u>** Entity, the **<u>Vendor</u>** Entity, the **<u>Employee</u>** Entity, the **<u>Department</u>** Entity, the **<u>Branch</u>** Entity, the **<u>Budget_Lines</u>** Entity, the **<u>Budget_Category</u>** Entity, the **<u>Budget_Allocation</u>** Entity The system calls the function <u>GetVendorSpendReport()</u> in the angular frontend that communicates to the appropriate header in the ASP.NET core API by means of an HTTPGET request. This function will get the following data from the **<u>Procurement_Details</u>** entity by means of an <u>SQL Read Query:</u> • ProcurementDetail ID • Employee ID • ProcurementRequest ID • Sign Off Status • ProcurementPayment Status ID • Account Code • Procurement Status ID • PaymentMethod ID • ItemType • BuyerName • BuyerEmail

			- DepositRequest - FullPaymentDue Date - TotalAmount - PaymentMade - Comment - ProofOfPayment Required The system will also INCLUDE the following details from the **<u>Procurement_Request</u>** Entity where the Procurement_Request _ID in the **<u>Procurement_Details</u>** entity is equal to the Procurement_Request _ID in the **<u>Procurement_Request</u>** Entity as a foreign key: - Procurement_ID - Vendor_ID - Requistion_ID - User_ID - Name - Description The system will also THENINCLUDE the following details from the **<u>Vendor</u>** Entity where the Vendor _ID in the **<u>Procurement_Request</u>** entity is equal to the Vendor _ID in the **<u>Vendor</u>** Entity as a foreign key: - Vendor_ID - Vendor_Status_ID - Name - Email

			- NumberOfTimes Used - SoleSupplierProvided The system will also INCLUDE the following details from the **Employee** Entity where the Employee_ID in the **Procurement_Details** entity is equal to the Employee_ID in the **Employee** Entity as a foreign key: - Employee_ID - User_ID - Department_ID - Branch_ID - Mandate_ID - EmployeeName - EmployeeSurname - CellPhone_Nr - Email The system will also THENINCLUDE the following details from the **Department** Entity where the Department _ID in the **Employee** entity is equal to the Department _ID in the **Department** Entity as a foreign key: - Department_ID - Name - Description The system will also THENINCLUDE the following details from the **Branch** Entity where the Branch _ID in the **Employee** entity

			is equal to the Branch _ID in the **Branch** Entity as a foreign key: • Branch_ID • Name • Street • City • Postal_Code • Province The system will also INCLUDE the following details from the **Budget_Line** Entity where the Account_Code in the **Procurement_Details** entity is equal to the Account_Code in the **Budget_Line** Entity as a foreign key: • Budget_ID • Category_ID • Account_Code • Month • BudgetAmt • ActualAmt • Variance The system will also THENINCLUDE the following details from the **Budget_Category** Entity where the Category _ID in the **Budget_Line** entity is equal to the Category _ID in the **Budget_Category** Entity as a primary key foreign key: • Category_ID • Account_Name • Description

			The system will also THENINCLUDE the following details from the **Budget_Allocation** Entity where the Budget _ID in the **Budget_Line** entity is equal to the Budget _ID in the **Budget_Allocation** Entity as a primary key foreign key: • Budget_ID • Department_ID • Date_Created • Year • Total The system will also include a condition WHERE the following details: • Procurement_Status_ID = 2 The Angular frontend will receive this data in the following array • VendorSpend[]

			Step 2: The angular frontend will generate a pdf using PDFMAKE as follows: • **Logo Header:** Moyo Logo • **Heading:** Vendor Spend Report • **Labels:** Generated By-

			[User], Generated on: [Date] • **Tabel: Tabel Columns:** Supplier, Account Code, Account Name(BudgetLine), Budget(Department), Company(Branch), BEE Level, Amount Spent. • **Tabel Rows:** Corresponding values, Total Amount for [Branch](reportbreak), Blank seperator. This continues until all branches are covered. • **Text:** **Report End** The Rows of the table will be populated using the data contained in the VendorSpend[] array.
			Step 3: The PDF is displayed to the user in a new tab **[ALT]**
Alternate courses:	**ALT Step 3:** The PDF is downloaded on the user's device. **The Use Case Ends.**		
Conclusion:	The system displays the Vendor Spend report to the user		
Post-condition:	The report is generated containing the Vendor Spend details		

Business rules	• Only the GRC are allowed to generate this report
Implementation constraints and specifications	• None
Assumptions:	• None
Open issues:	• None

Table 117: 7.3 Generate Vendor Spend Report

USE CASE 7.4

System Name: Procion System **Sub-System:**		
Author: Leon Combrinck	Date: 8 August 2023	Version: 1.0.0

Use case name:	Generate Consumable Inventory Management Report	**USE CASE TYPE**
Use case id:	7.4	Business Requirements: ☐
Priority:	High	System Analysis: ☐
Source:	Moyo	System Design: ☒
Primary business actor	User	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	
Description:	The use case describes the event where a user will want to generate a report which contains all of the Consumable Inventory Management stored on the system. The user will click on the Generate Consumable Inventory Management Report button located on the “View Generate Report” screen. The system will get all of the Consumable, Consumable_History in the	

	database. The system will Then display a screen modal popup where user will be able to enter DateCaptured[start and end date] filter criteria. The system will then prompt the user to generate the report. The User clicks on the "Generate" button on the popup modal screen. The pdf will be generated and displayed to a user in a new tab. The use case ends		
Pre-condition:	The user has to be logged in to the system.		
Trigger:	The user requests to view Generate Report screen on the system in a report The user clicks on the Consumable Inventory Management Generate button. OR The User clicks on the Consumable Inventory Management download button.		
Typical course **Of events:**	**Actor Action**	**System Response**	
		Manual Action	**Automated Action**
			Step 1: The system will send a request form the Angular frontend to the Report service where the service will make a http get request to the .NET Core backend which makes use of a Lambda Linq Query which creates a SQL Read query to retrieve all the items from the **Consumable** Entity, the **Consumable_History** Entity The system calls the function GetCIMReport() in the angular frontend that communicates to the appropriate header in the ASP.NET core API by means of an HTTPGET request. This function will get

			the following data from the **Consumable_History** entity by means of an SQL Read Query: • History_ID • Consumable_ID • StockAmt • DateCaptured The system will also INCLUDE the following details from the **Consumable** Entity where the Consumable_ID in the **Consumable_History** entity is equal to the Consumable_ID in the **Consumable** Entity as a foreign key: • Consumable_ID • ConsumableCategory_ID • Branch_ID • Name • Description • On_Hand • Minimum_Reorder_Quantity • Maximim_Reorder_Quantity The Angular frontend will receive this data in the following array • ConsumableInventoryManagement[]

			Step 2: The system will display the BEE Spend

			popup model with the following information: • **Title:** Select filter Criteria. • **Text:** Please provide the following details. • **Input Field:** Calendar date picker[start-end] • **Button:** Generate, Cancel The System prompt the user to enter the required field details: • DateCaptured
	Step 3: The user Provides the following details: • DateCaptured		**Step 4:** The system validates the details that was entered. -That the “Date Range” input box is not null, start date is not invalid, end date is not invalid. **[ALT]** The system prompts the user the click on the generate button.
	Step 5: The User clicks on the “Generate” Button **[ALT]**		**Step 6:** The angular frontend will generate a pdf using PDFMAKE as follows: • **Logo Header:** Moyo Logo • **Heading:** Consumable Inventory Management Report

			• **Labels:** Created By: [User], Generated on: [Date] Procurement totals For Period[StartDate -EndDate] • **Tabel Columns:** Consumable, January-December. • **Tabel Rows:** All the consumables. • This is repeated until all Consumables are covered • **Text:** **End of Report** The Rows of the table will be populated using the data contained in the ConsumableInventory Management[] array.
			Step 7: The PDF is displayed to the user in a new tab **[ALT]**
Alternate courses:	**ALT Step 4:** The field do not correspond to validation criteria. A red text under the input box will display explaining the validation error. Return to step 3. **ALT Step 5:** The user clicks on the “Cancel” Button. **The Use Case Ends.** **ALT Step 7:** The PDF is downloaded on the user’s device. **The Use Case Ends.**		
Conclusion:	The system displays the Consumable Inventory Management report to the user		

Post-condition:	The report is generated containing the Consumable Inventory Management details
Business rules	• Only the Business Enablement are allowed to generate this report
Implementation constraints and specifications	• None
Assumptions:	• None
Open issues:	• None

Table 118: 7.4 Generate Consumable inventory Management Report

USE 7.5

System Name: Procion System **Sub-System:**		
Author: Leon Combrinck	Date: 8 August 2023	Version: 1.0.0

Use case name:	Generate Business Unit Allocation Report	**USE CASE TYPE**
Use case id:	7.5	Business Requirements: ☐
Priority:	High	System Analysis: ☐
Source:	Moyo	System Design: ☒
Primary business actor	User	
Primary system actor	None	
Other participating actors:	None	
Other interested stakeholders:	None	

Description:	The use case describes the event where a user will want to generate a report which contains all of the Business Unit Allocation stored on the system. The user will click on the Generate Business Unit Allocation Report button located on the "View Generate Report" screen. The system will get all of the Budget_Lines, Budget_Category, Budget_Allocation, Department in the database. The system will Then display a screen modal popup where user will be able to enter DateCreated[start and end date] and Department filter criteria. The system will then prompt the user to generate the report. The User clicks on the "Generate" button on the popup modal screen. The pdf will be generated and displayed to a user in a new tab. The use case ends		
Pre-condition:	The user has to be logged in to the system.		
Trigger:	The user requests to view Generate Report screen on the system in a report The user clicks on the Business Unit Allocation Generate button. OR The User clicks on the Business Unit Allocation download button.		
Typical course **Of events:**	**Actor Action**	**System Response**	
		Manual Action	**Automated Action**
			Step 1: The system will send a request form the Angular frontend to the Report service where the service will make a http get request to the .NET Core backend which makes use of a Lambda Linq Query which creates a SQL Read query to retrieve all the items from the **Budget_Lines** Entity, the **Budget_Category** Entity, the **Budget_Allocation** Entity, , the **Department** Entity, The system calls the function GetBUAReport() in the angular frontend that communicates to the

			appropriate header in the ASP.NET core API by means of an HTTPGET request. This function will get the following data from the **Budget_Lines** entity by means of an SQL Read Query: • Budget_ID • Category_ID • Account_Code • Month • BudgetedAmt • ActualAmt • Variance The system will also INCLUDE the following details from the **Budget_Category** Entity where the Category_ID in the **Budget_Lines** entity is equal to the Category_ID in the **Budget_Category** Entity as a Primary key foreign key: • Category_ID • AccountName • Description • User_ID The system will also INCLUDE the following details from the **Budget_Allocation** Entity where the Budget_ID in the **Budget_Lines** entity is equal to the Budget_ID in the **Budget_Allocation**

			Entity as a primary key foreign key: • Budget_ID • Department_ID • DateCreated • Total • Year The system will also THENINCLUDE the following details from the **Department** Entity where the Department _ID in the **Budget_Allocation** entity is equal to the Department _ID in the **Department** Entity as a foreign key: • Department_ID • Name • Description The Angular frontend will receive this data in the following array • BusinessUnitAllocation[]

			Step 2: The system will display the BEE Spend popup model with the following information: • **Title:** Select filter Criteria. • **Text:** Please provide the following details. • **Input Field:** Calendar date picker[start-end]

			• **Dropdown:** Department • **Button:** Generate, Cancel The System prompt the user to enter the required field details: • DateCaptured • Department
	Step 3: The user Provides the following details: • DateCaptured • Department		**Step 4:** The system validates the details that was entered. -That the "Date Range" input box is not null, start date is not invalid, end date is not invalid. -That the "Department" input box is not null. **[ALT]** The system prompts the user the click on the generate button.
	Step 5: The User clicks on the "Generate" Button **[ALT]**		**Step 6:** The angular frontend will generate a pdf using PDFMAKE as follows: • **Logo Header:** Moyo Logo • **Heading:** Business Unit Allocation Report • **Labels:** Generated By-[User], Generated on: [Date] • **Tabel:**

			• **Tabel Columns:** Department, Account Code, Actual Amount, Budgeted Amount, Variance • **Tabel Rows:** Covering all Department based budget lines, control break for that budget department. • This continues till all departments and budget lines are covered **Header:** Column Barchart showing business unit allocation. • **Legend:** Variance, Actual Amount, Budgeted Amount • **Bar Chart y-axis:** Amounts. • **Bar Chart x-axis:** Departments per Budget lines. • **Text:** **Report End** The Rows of the table will be populated using the data contained in the BusinessUnitAllocation[ ] array.

			Step 7: The PDF is displayed to the user in a new tab **[ALT]**

Alternate courses:	**ALT Step 4:** The field do not correspond to validation criteria. A red text under the input box will display explaining the validation error. Return to step 3. **ALT Step 5:** The user clicks on the "Cancel" Button. **The Use Case Ends.** **ALT Step 7:** The PDF is downloaded on the user's device. **The Use Case Ends.**
Conclusion:	The system displays the Business Unit Allocation report to the user
Post-condition:	The report is generated containing the Business Unit Allocation details
Business rules	• Only the Finance are allowed to generate this report
Implementation constraints and specifications	• None
Assumptions:	• None
Open issues:	• None

Table 119: 7.5 Generate Business Unite Spend Report

USE CASE 7.6

System Name: Procion System **Sub-System:**		
Author: Leon Combrinck	Date: 8 August 2023	Version: 1.0.0

Use case name:	Generate Budget Variance Report	**USE CASE TYPE**
Use case id:	7.6	Business Requirements: ☐
Priority:	High	System Analysis: ☐

Source:	Moyo	System Design: ☒	
Primary business actor	User		
Primary system actor	None		
Other participating actors:	None		
Other interested stakeholders:	None		
Description:	The use case describes the event where a user will want to generate a report which contains all of the Budget Variance stored on the system. The user will click on the Generate Budget Variance Report button located on the "View Generate Report" screen. The system will get all of the Budget_Lines, Budget_Category, Budget_Allocation in the database. The system will Then display a screen modal popup where user will be able to enter DateCreated [start and end date] filter criteria. The system will then prompt the user to generate the report. The User clicks on the "Generate" button on the popup modal screen. The pdf will be generated and displayed to a user in a new tab. The use case ends		
Pre-condition:	The user has to be logged in to the system.		
Trigger:	The user requests to view Generate Report screen on the system in a report The user clicks on the Budget Variance Generate button. OR The User clicks on the Budget Variance download button.		
Typical course **Of events:**	**Actor Action**	**System Response**	
		Manual Action	**Automated Action**
			Step 1: The system will send a request form the Angular frontend to the Report service where the service will make a http get request to the .NET Core backend which makes use of a Lambda Linq Query which creates a

			SQL Read query to retrieve all the items from the **Budget_Lines** Entity, the **Budget_Category** Entity, the **Budget_Allocation** Entity. The system calls the function GetBVReport() in the angular frontend that communicates to the appropriate header in the ASP.NET core API by means of an HTTPGET request. This function will get the following data from the **Budget_Lines** entity by means of an SQL Read Query: • Budget_ID • Category_ID • Account_Code • Month • BudgetedAmt • ActualAmt • Variance The system will also INCLUDE the following details from the **Budget_Category** Entity where the Category_ID in the **Budget_Lines** entity is equal to the Category_ID in the **Budget_Category** Entity as a Primary key foreign key: • Category_ID • AccountName • Description

			• User_ID The system will also INCLUDE the following details from the **<u>Budget_Allocation</u>** Entity where the Budget_ID in the **<u>Budget_Lines</u>** entity is equal to the Budget_ID in the **<u>Budget_Allocation</u>** Entity as a primary key foreign key: • Budget_ID • Department_ID • DateCreated • Total • Year The Angular frontend will receive this data in the following array • BudgetVariance[ ]

			Step 2: The system will display the BEE Spend popup model with the following information: • **Title:** Select filter Criteria. • **Text:** Please provide the following details. • **Input Field:** Calendar date picker[start-end] • **Button:** Generate, Cancel

			The System prompt the user to enter the required field details: • DateCreated
	Step 3: The user Provides the following details: • DateCreated		**Step 4:** The system validates the details that was entered. -That the "Date Range" input box is not null, start date is not invalid, end date is not invalid. **[ALT]** The system prompts the user the click on the generate button.
	Step 5: The User clicks on the "Generate" Button **[ALT]**		**Step 6:** The angular frontend will generate a pdf using PDFMAKE as follows: • **Logo Header:** Moyo Logo • **Heading:** Budget Variance Report • **Labels:** Created By: [User], Generated on: [Date] • **Tabel:** • **Tabel Columns:** Month, Budget Category, Budgeted Amount, Actual Amount, Variance, Variance %. • **Tabel Rows:** Grouped by Category,

			Control Break per Category. • This is repeated until all Categories are covered • **Bar Chart Header:** Bar Chart showing total expenses, for different Categories • **Legend:** Yearly Totals. • **Bar Chart y-axis:** Amount. • **Bar Chart x-axis:** Category. • **Line Chart Header:** Line Chart Showing Monthly Expenditure • **Legend:** Monthly Expenditure • **Bar Chart y-axis:** Amount. • **Bar Chart x-axis:** Month. • **Text:** **Report End** The Rows of the table will be populated using the data contained in the BudgetVariance[] array.
			Step 7: The PDF is displayed to the user in a new tab **[ALT]**

Alternate courses:	**ALT Step 4:** The field do not correspond to validation criteria. A red text under the input box will display explaining the validation error. Return to step 3. **ALT Step 5:** The user clicks on the “Cancel” Button. **The Use Case Ends.** **ALT Step 7:** The PDF is downloaded on the user’s device. **The Use Case Ends.**
Conclusion:	The system displays the Budget Variance report to the user
Post-condition:	The report is generated containing the Budget Variance details
Business rules	• Only the Business Owner are allowed to generate this report
Implementation constraints and specifications	• None
Assumptions:	• None
Open issues:	• None

Table 120: 7.6 Generate Budget Variance Report

USE CASE 7.7

System Name: Procion System **Sub-System:**		
Author: Leon Combrinck	Date: 8 August 2023	Version: 1.0.0

Use case name:	View Generate Report	**USE CASE TYPE**
Use case id:	7.7	Business Requirements: ☐
Priority:	High	System Analysis: ☐

<table>
<tr><td>Source:</td><td>Moyo</td><td colspan="2">System Design: ☒</td></tr>
<tr><td>Primary business actor</td><td colspan="3">User</td></tr>
<tr><td>Primary system actor</td><td colspan="3">None</td></tr>
<tr><td>Other participating actors:</td><td colspan="3">None</td></tr>
<tr><td>Other interested stakeholders:</td><td colspan="3">None</td></tr>
<tr><td>Description:</td><td colspan="3">The use case describes the event where a user will Requests to view the generate report screen on the system. The user will click on the “Report” button located on the top navbar on the system. The system will Load the View Generate Report Screen with all its components and display the screen to the user. The system will then prompt the user to select the report they wish to generate. The user then selects no reports to generate and the use case ends.</td></tr>
<tr><td>Pre-condition:</td><td colspan="3">The user has to be logged in to the system.

The system loaded the top navbar or home screen.</td></tr>
<tr><td>Trigger:</td><td colspan="3">The user requests to view Generate Report screen on the system in a report

The user clicks on the “Reports” button on the top Navbar</td></tr>
<tr><td rowspan="3">Typical course

Of events:</td><td rowspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Manual Action</td><td>Automated Action</td></tr>
<tr><td>Step 1: The User navigates from the home screen to the Reports screen by clicking on the Reports button on the top <u>user</u> navbar.

[ALT]</td><td></td><td>Step 2: The system will make use of the angular frontend routing to navigate the user to the Reports screen.

The system will then display the reports screen with the following details:

Main Navbar: (From left to right)</td></tr>
</table>

			• Moyo Logo (Just a logo) • Home Icon • Administration • Inventory • Finance • Procurement • Vendor • Reports • **Help** • Notification Bell Icon • Profile Icon Dropdown **Reports Content:** • **Title:** Reports. • **Labels with corresponding “Generate” Buttons.** -Generate Approved Vendor List Report, -Generate BEE Spend Report, -Generate Vendor Spend Report, -Generate Consumable Inventory Management Report, -Generate Business Unit Allocation Report, -Generate Budget Variance Report. **Next to each “Generate” Button is a dropdown arrow with the following details:**

			• Generate Report • Download Report

			Step 3: The system Prompts the user to select the report they wish to generate by clicking on the Generate or dropdown arrow button.
	Step 4: The user does not click on any generate buttons. **[ALT]**		
Alternate courses:	**ALT Step 1:** The User Navigates from the home screen to the reports screen by clicking on the **Reports** button on the home screen. The use Case Continues to step 2		
	ALT Step 4ai: The user Clicks on the Approved Vendor "Generate" Button or the "Generate Report" or "Download Report" button under the dropdown located on the View Generate Report screen. The system will extend to the use case 7.1 "Generate Approve Vendor Report". **The Use Case Ends.** **ALT Step 4b:** The user Clicks on the BEE Spend "Generate" report Button or the "Generate Report" or "Download Report" button under the dropdown located on the View Generate Report screen. The system will extend to the use case 7.2 "Generate BEE Spend Report". **The Use Case Ends.** **ALT Step 4c:** The user Clicks on the Vendor Spend "Generate" report Button or the "Generate Report" or "Download Report" button under the dropdown located on the View Generate Report screen. The system will extend to the use case 7.3 "Generate Vendor Spend Report". **The Use Case Ends.**		

	ALT Step 4d: The user Clicks on the Consumable Inventory Management "Generate" report Button or the "Generate Report" or "Download Report" button under the dropdown, located on the View Generate Report screen. The system will extend to the use case 7.4 "Consumable Inventory Management Report". **The Use Case Ends.** **ALT Step 4e:** The user Clicks on the Business Unit Allocation "Generate" report Button or the "Generate Report" or "Download Report" button under the dropdown, located on the View Generate Report screen. The system will extend to the use case 7.4 "Generate Business Unit Allocation Report". **The Use Case Ends.** **ALT Step 4f:** The user Clicks on the Budget Variance "Generate" report Button or the "Generate Report" or "Download Report" button under the dropdown, located on the View Generate Report screen. The system will extend to the use case 7.6 "Generate Budget Variance Report". **The Use Case Ends.**
Conclusion:	The system displays the report screen to the user
Post-condition:	The report screen details is displayed.
Business rules	• Only the GRC, Business owner, Business Enablement, Finance is allowed to view the reports.
Implementation constraints and specifications	• None
Assumptions:	• None
Open issues:	• None

Table 121: 7.7 View Generate Report

5.3 CONCLUSION

This concludes the Logical Use case Narratives section of the Procion system.

6. TECHNICAL CONTEXT DIAGRAM

6.1 INTRODUCTION

In this section we will display the visual representation of the Technical Context diagrams for the following use cases: 2.26,2.37, 4.10,5.9, 7.1-7.7 of the Procion system.

6.2 TECHNICAL CONTEXT DIAGRAM

USE CASE 2.26

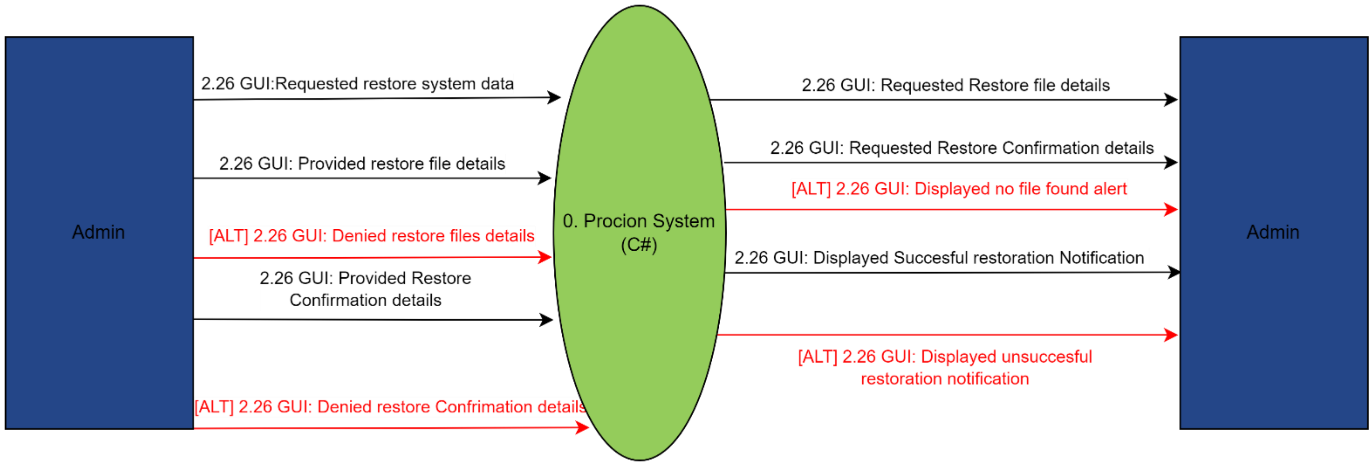


Figure 36 Use Case 2.26 Tech Context

USE CASE 2.37

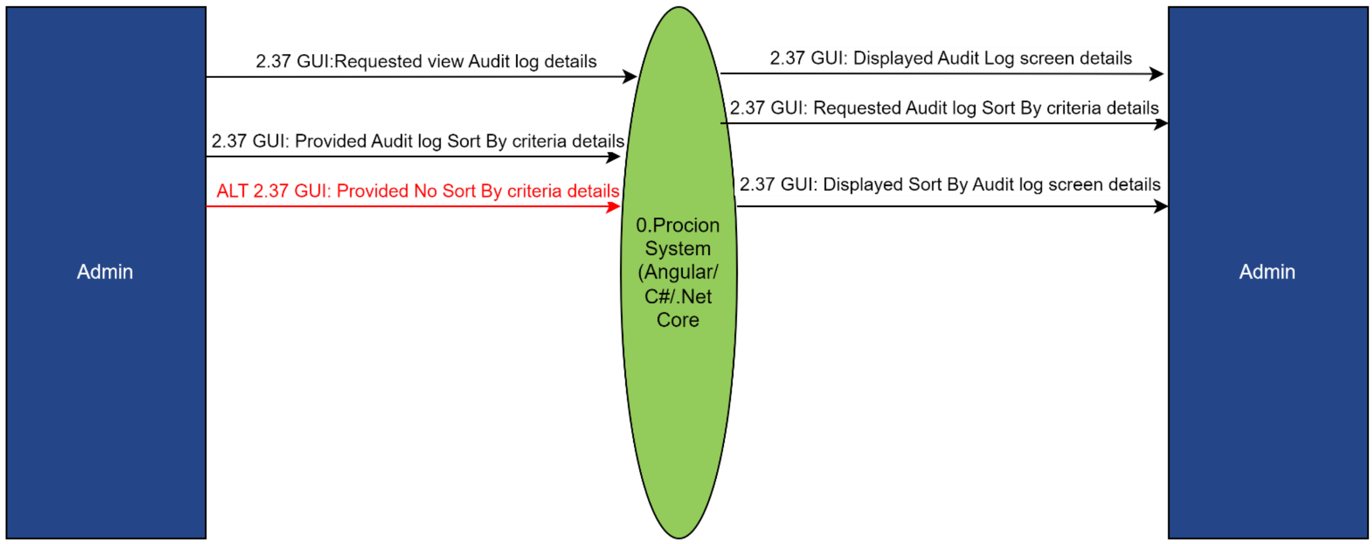


Figure 37 Use Case 2.37 Tech Context

USE CASE 4.10

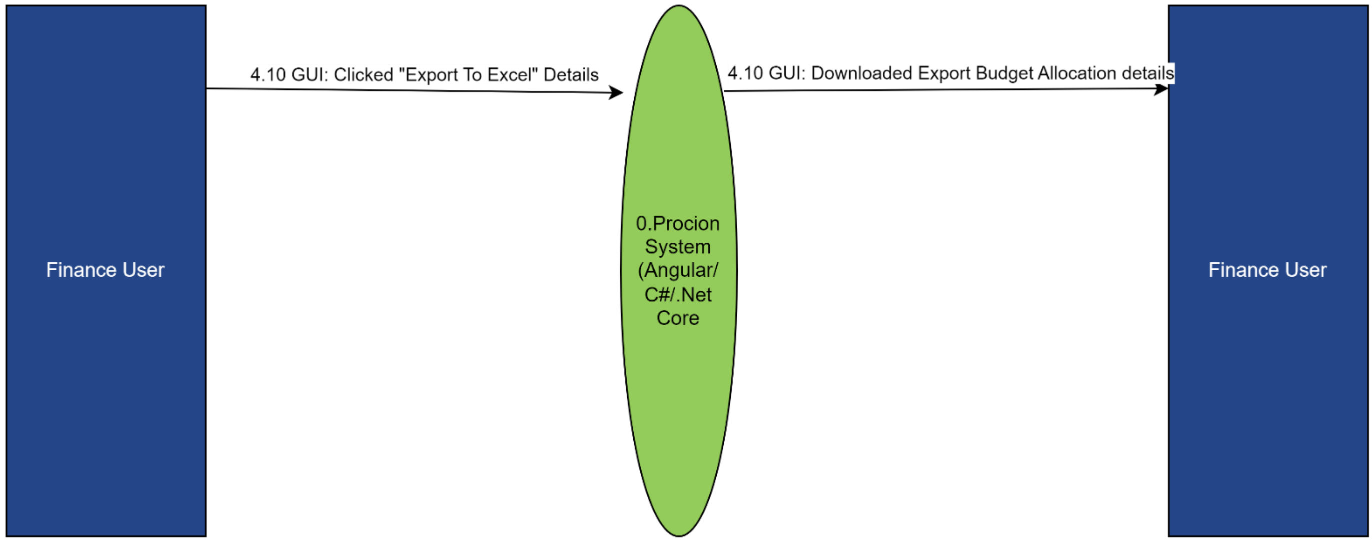


Figure 38 Use Case 4.10 Tech Context

USE CASE 5.9

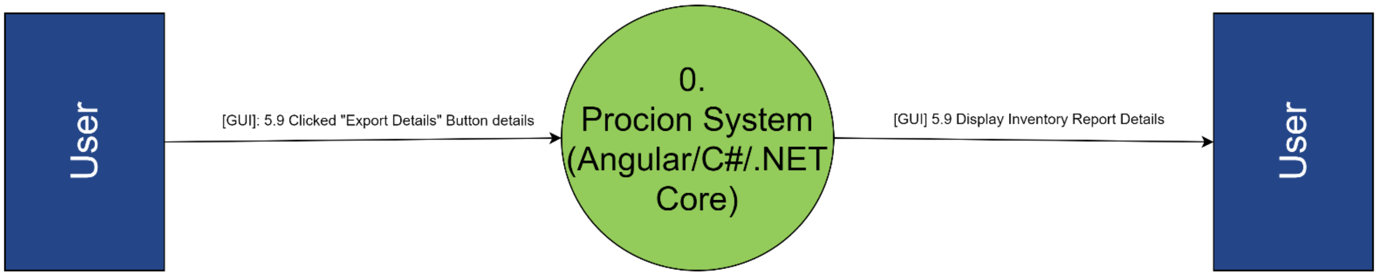


Figure 39 Use Case 5.9 Tech Context

USE CASE 7.1-7.7

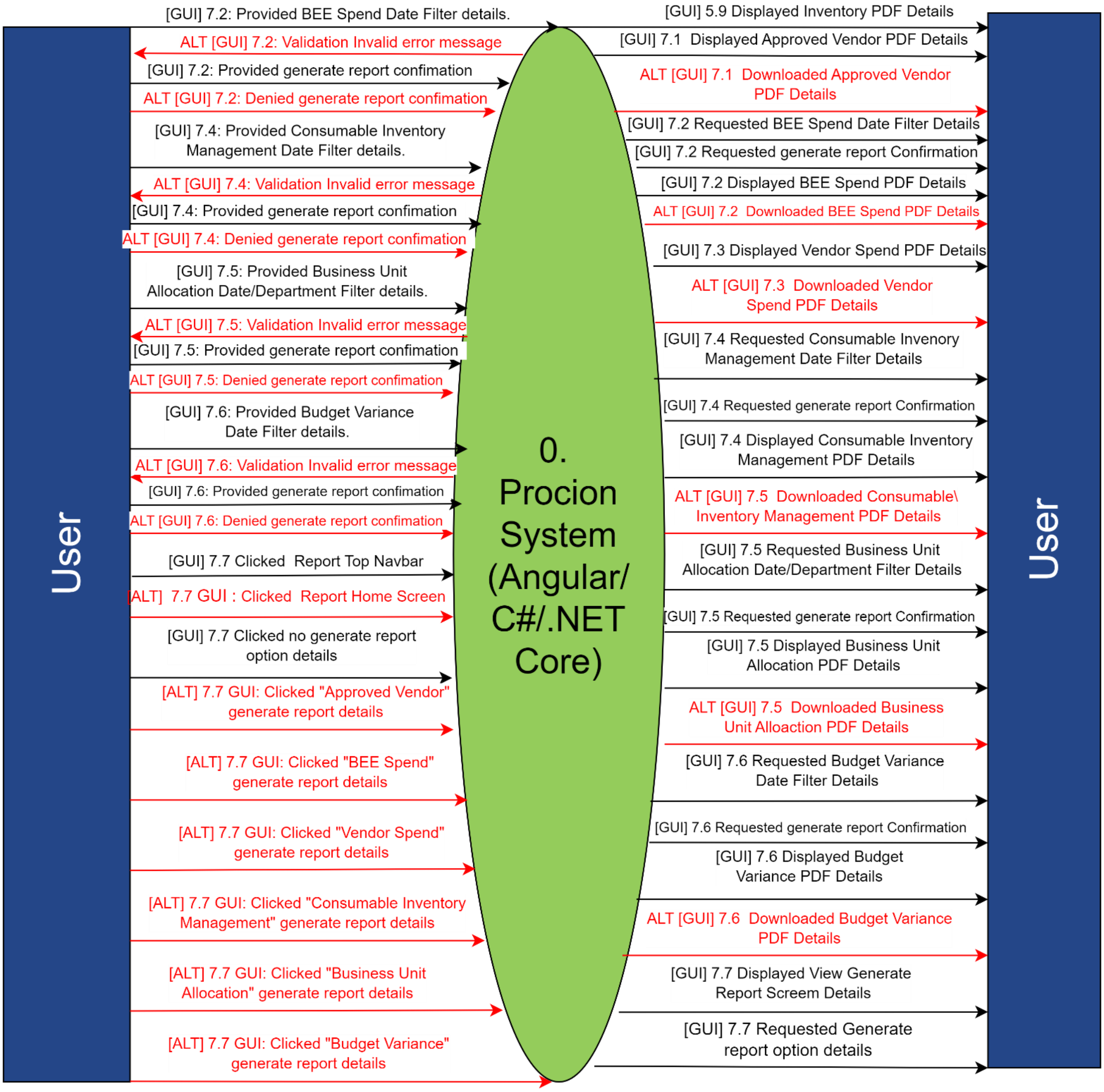


Figure 40 Use Case 7.1-7.7 Tech Context

6.3 CONCLUSION

This Concludes the visual representation of the Technical Context diagrams for the Procion system.

7. TECHNICAL PRIMITIVE LEVEL DIAGRAM

7.1 INTRODUCTION

In this section we will display the visual representation of the Technical Primitive level diagrams for the following use cases: 2.26, 2.37, 4.10 , 5.9, 7.1-7.7 respectively, of the Procion system.

7.2 TECHNICAL PRIMITIVE LEVEL DIAGRAM

USE CASE 2.26

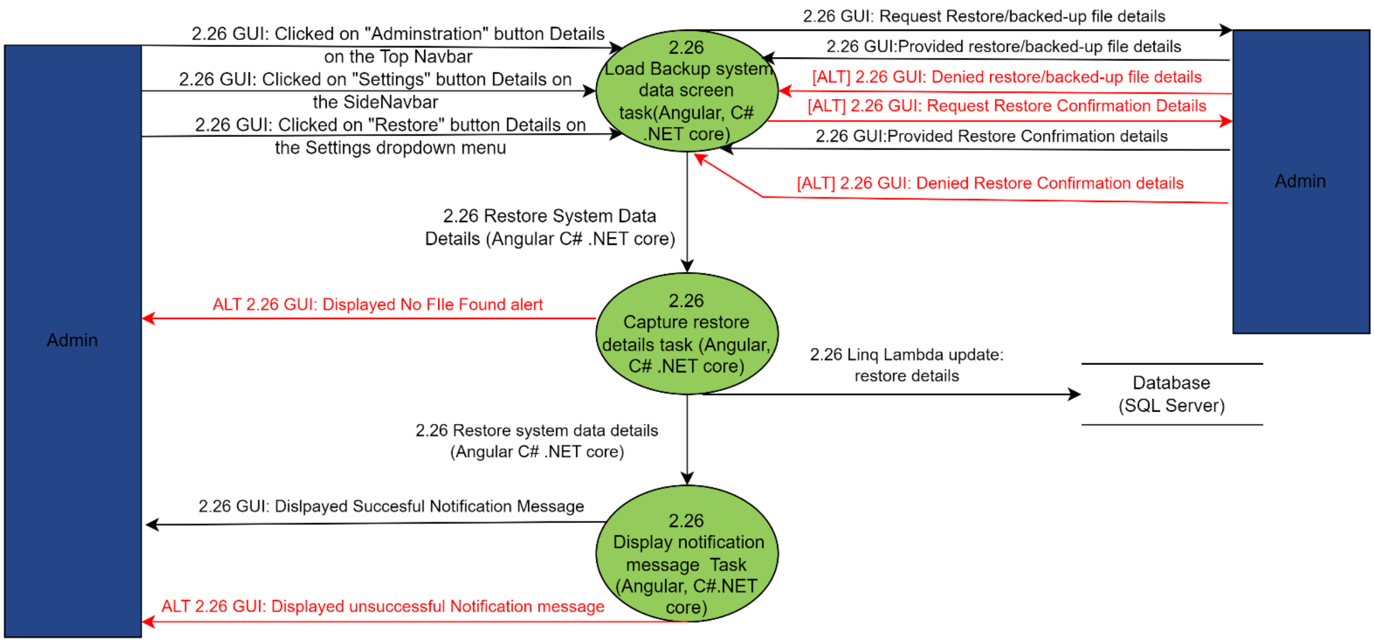


Figure 41 2.26 Restore System Data Tech Prim

USE CASE 2.37

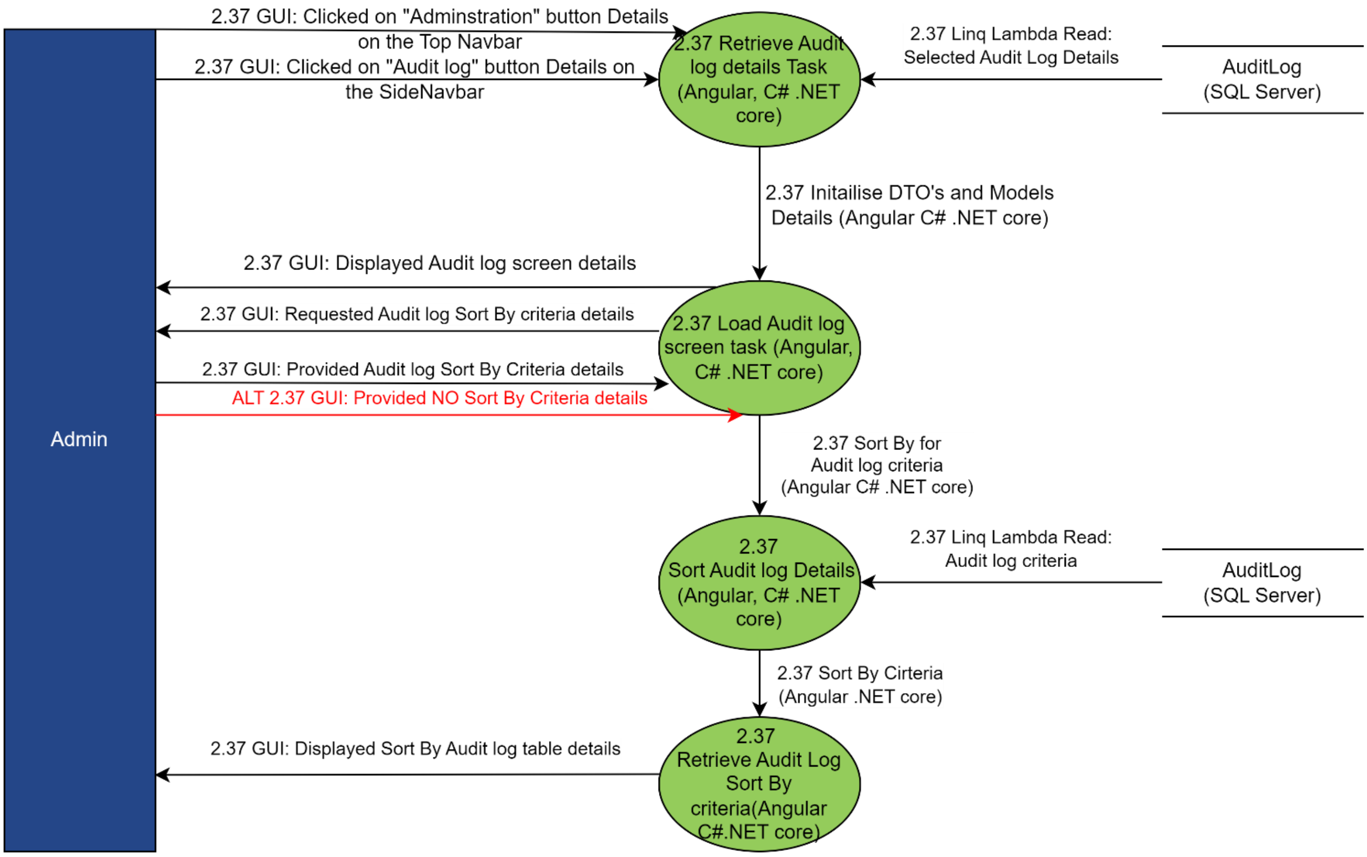


Figure 42 2.37 View Audit Log Tech Prim

USE CASE 4.10

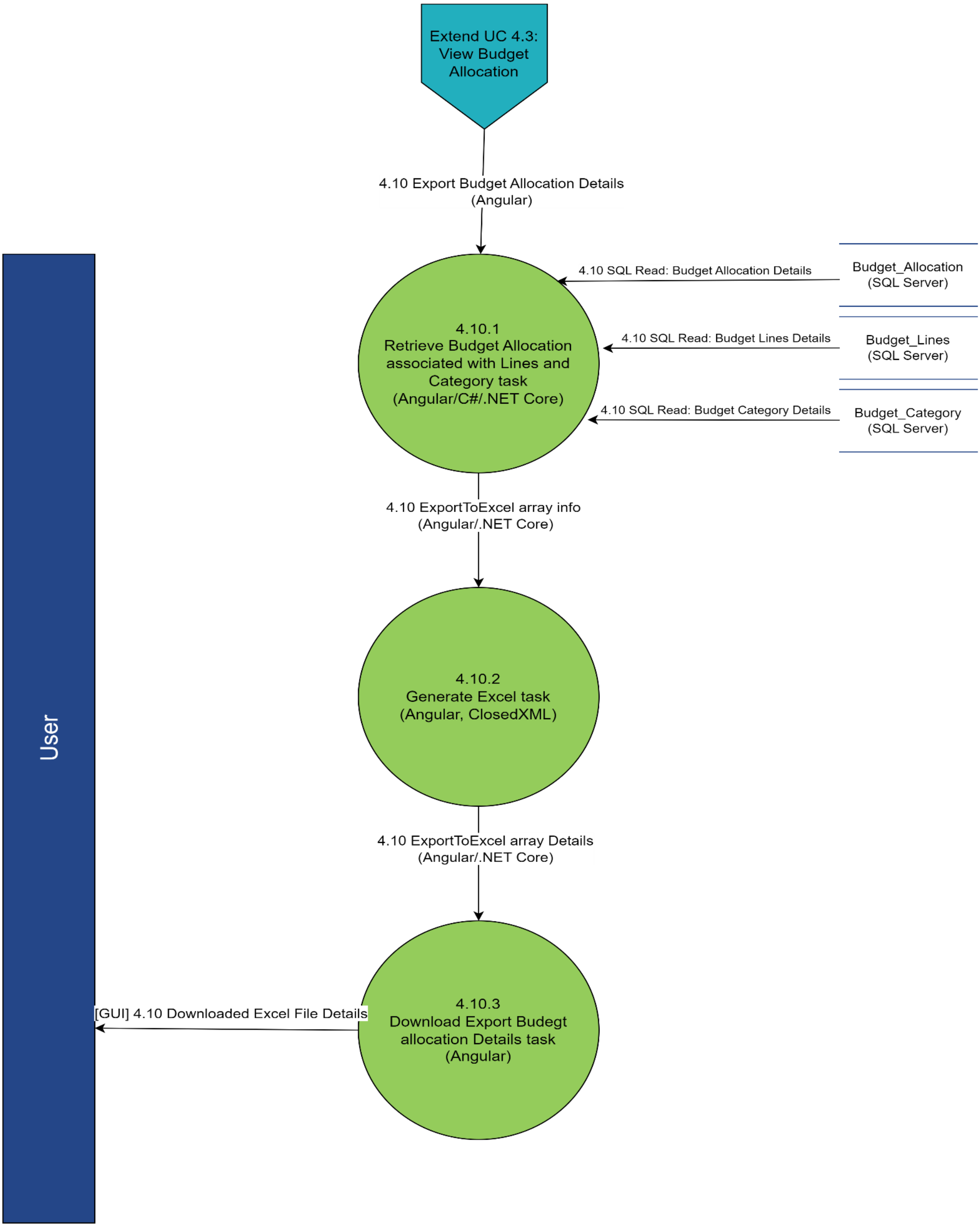


Figure 43 4.10 Export Budget Alloaction Tech Prim

USE CASE 5.9

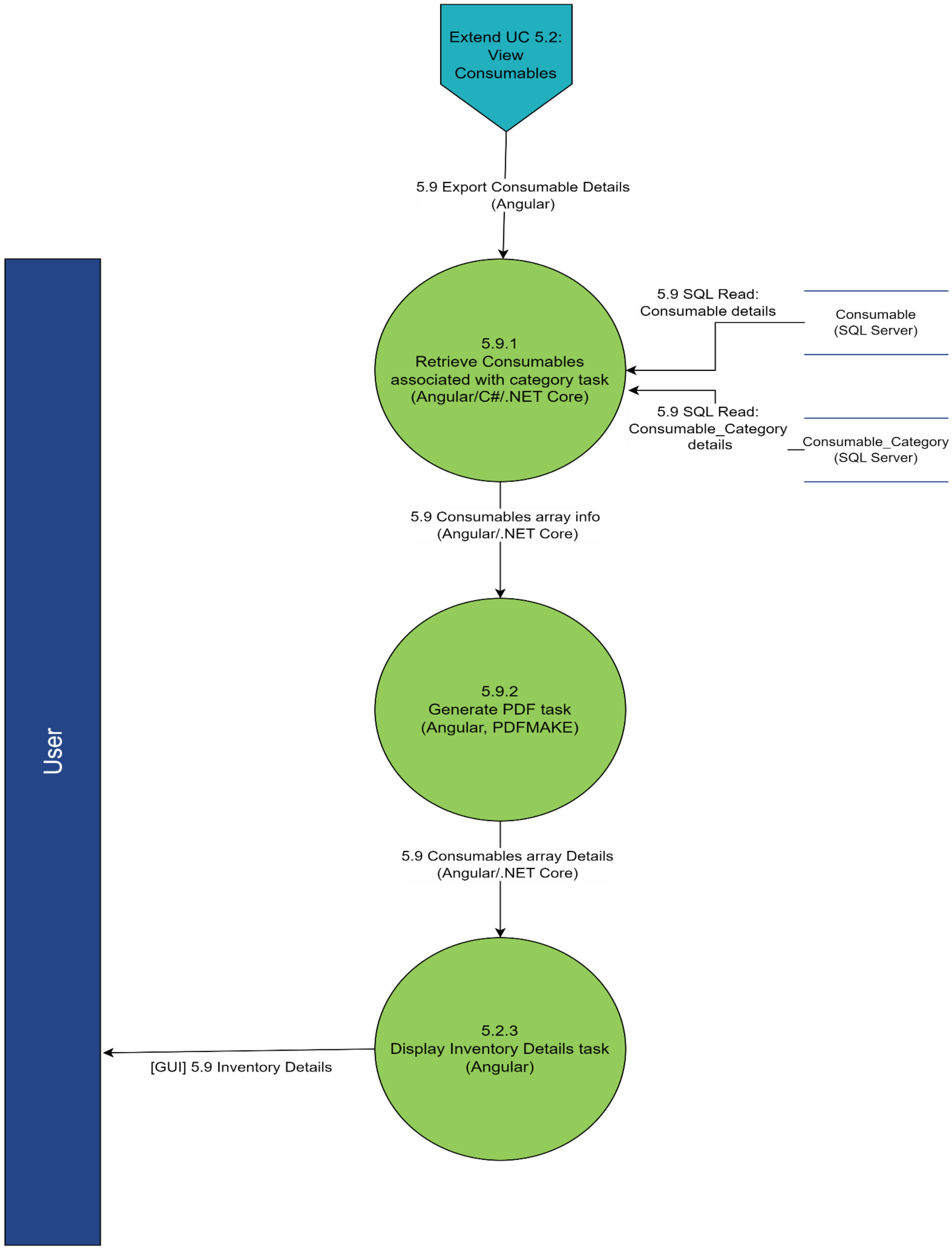


Figure 44 5.9 Export Inventory Details Tech Prim

USE CASE 7.1

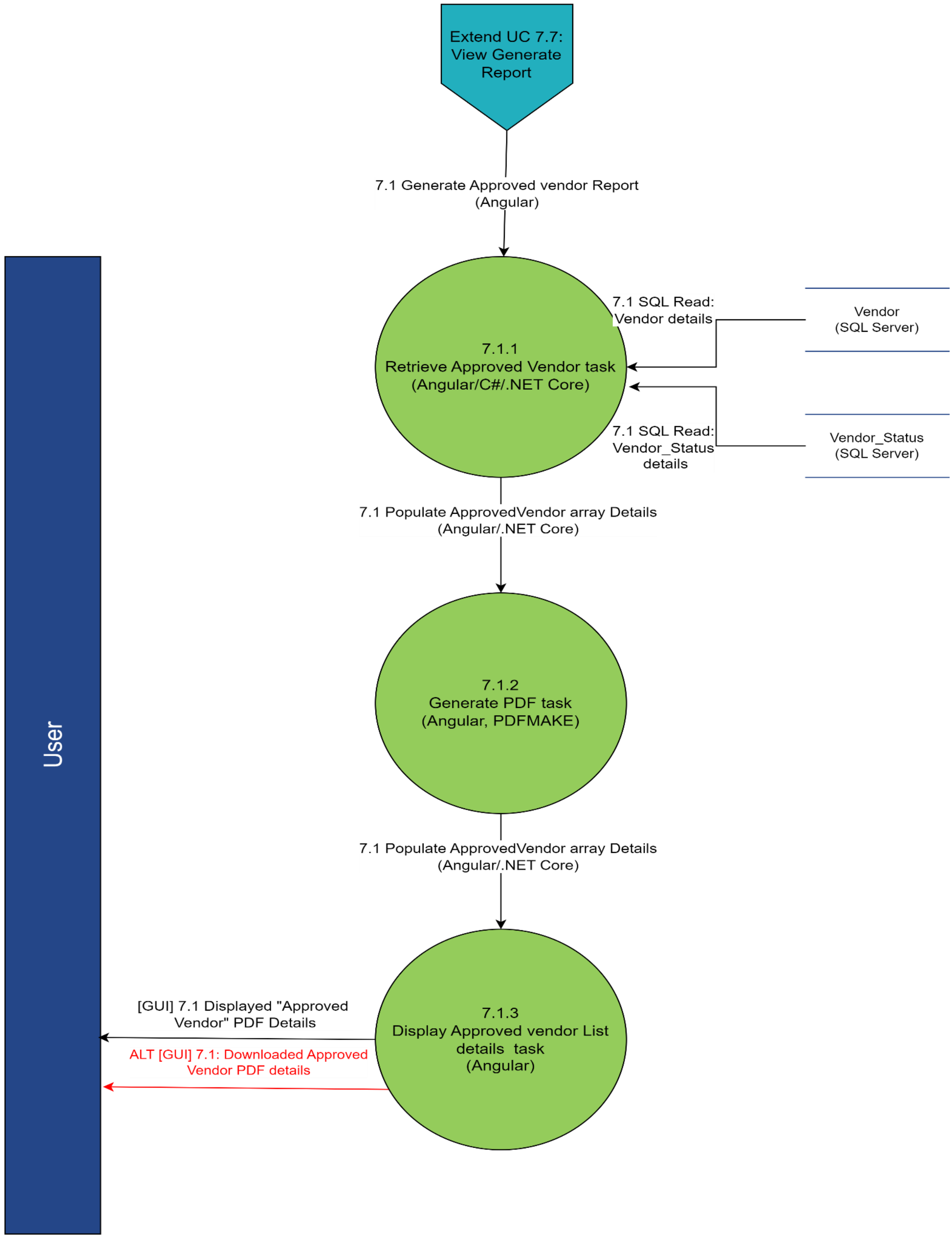


Figure 45 7.1 Generate Approved Vendor Report Tech Prim

USE CASE 7.2

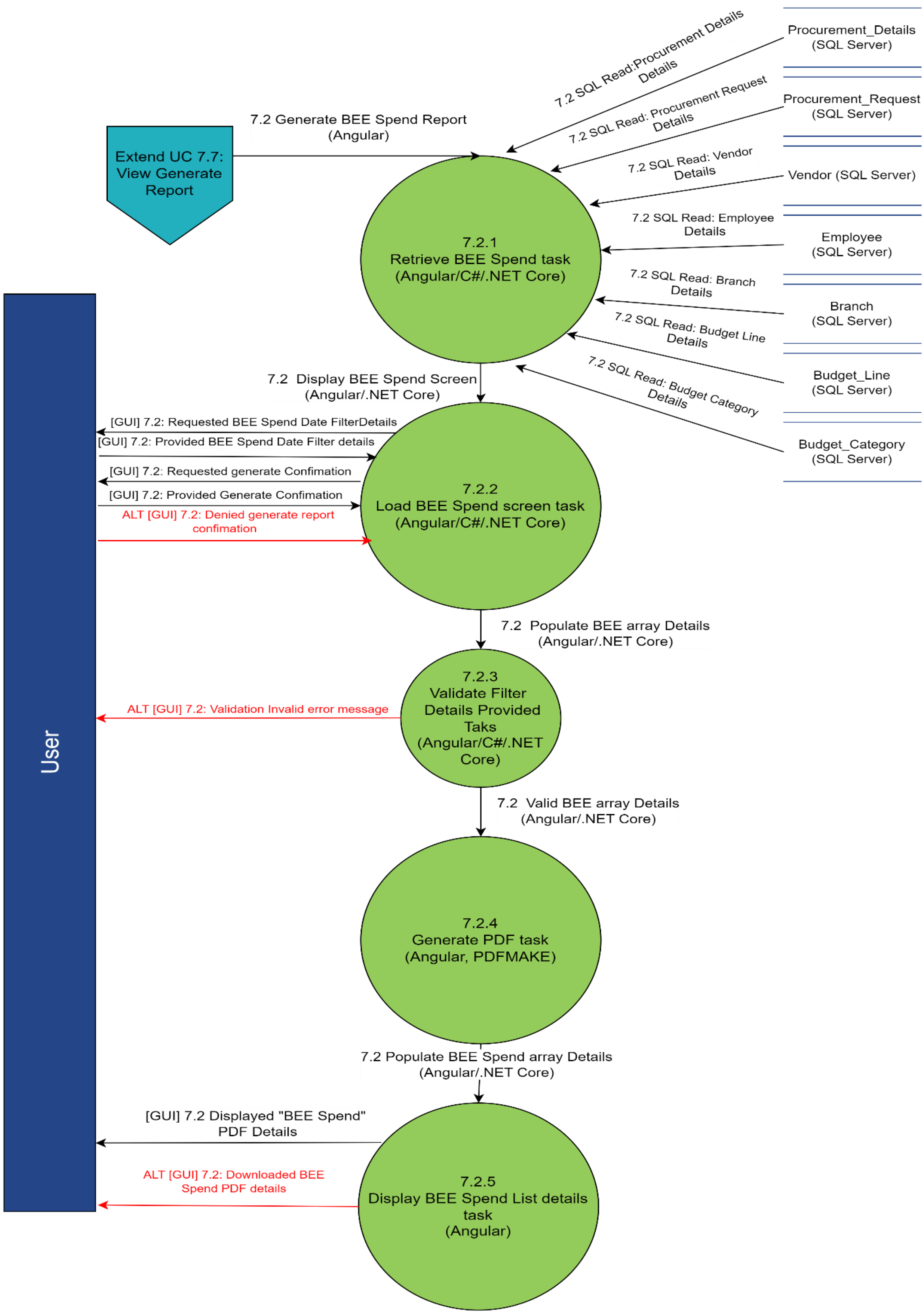


Figure 46 7.2 Generate BEE Spend Report Tech Prim

USE CASE 7.3

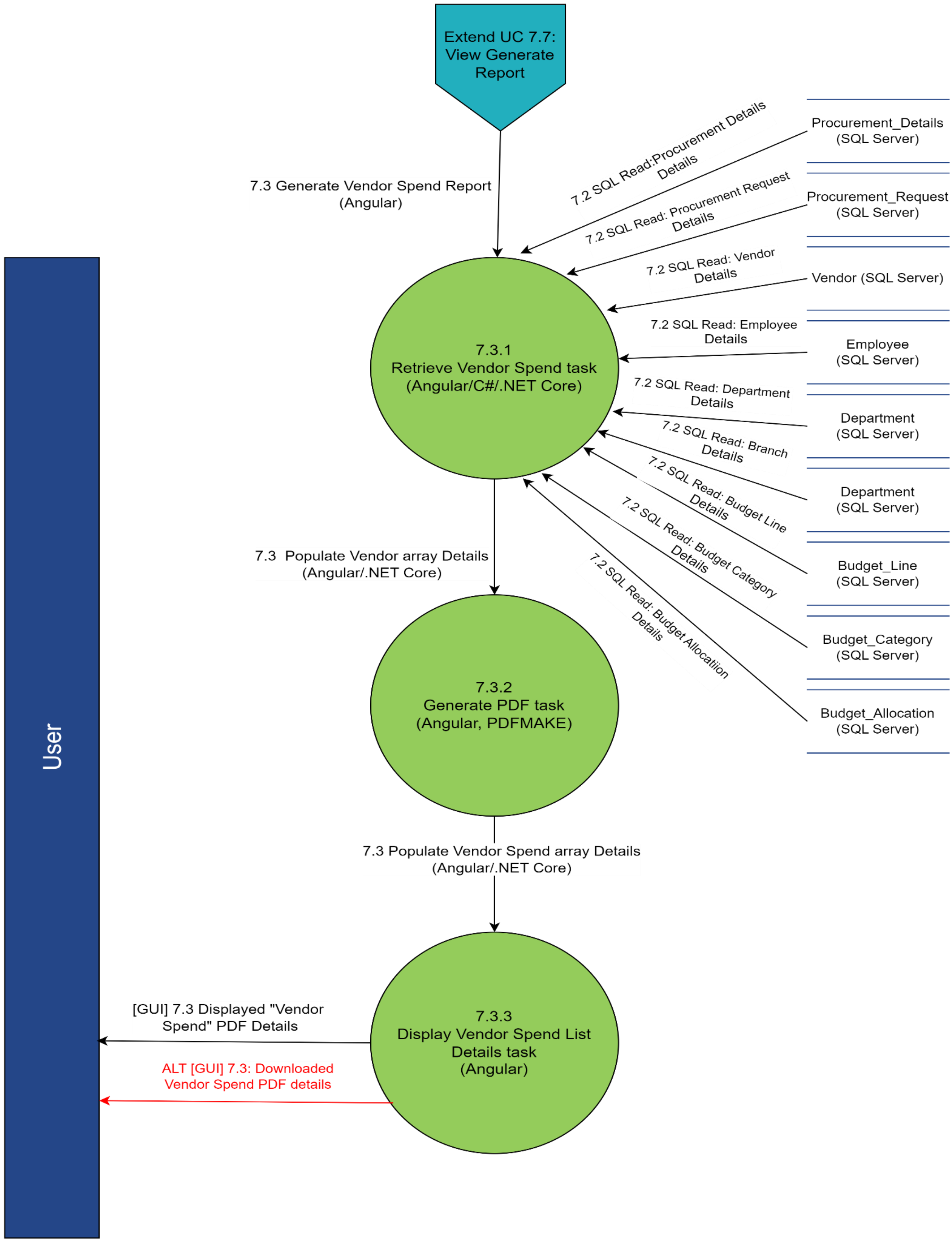


Figure 47 7.3 Generate Vendor Spend Report Tech Prim

USE CASE 7.4

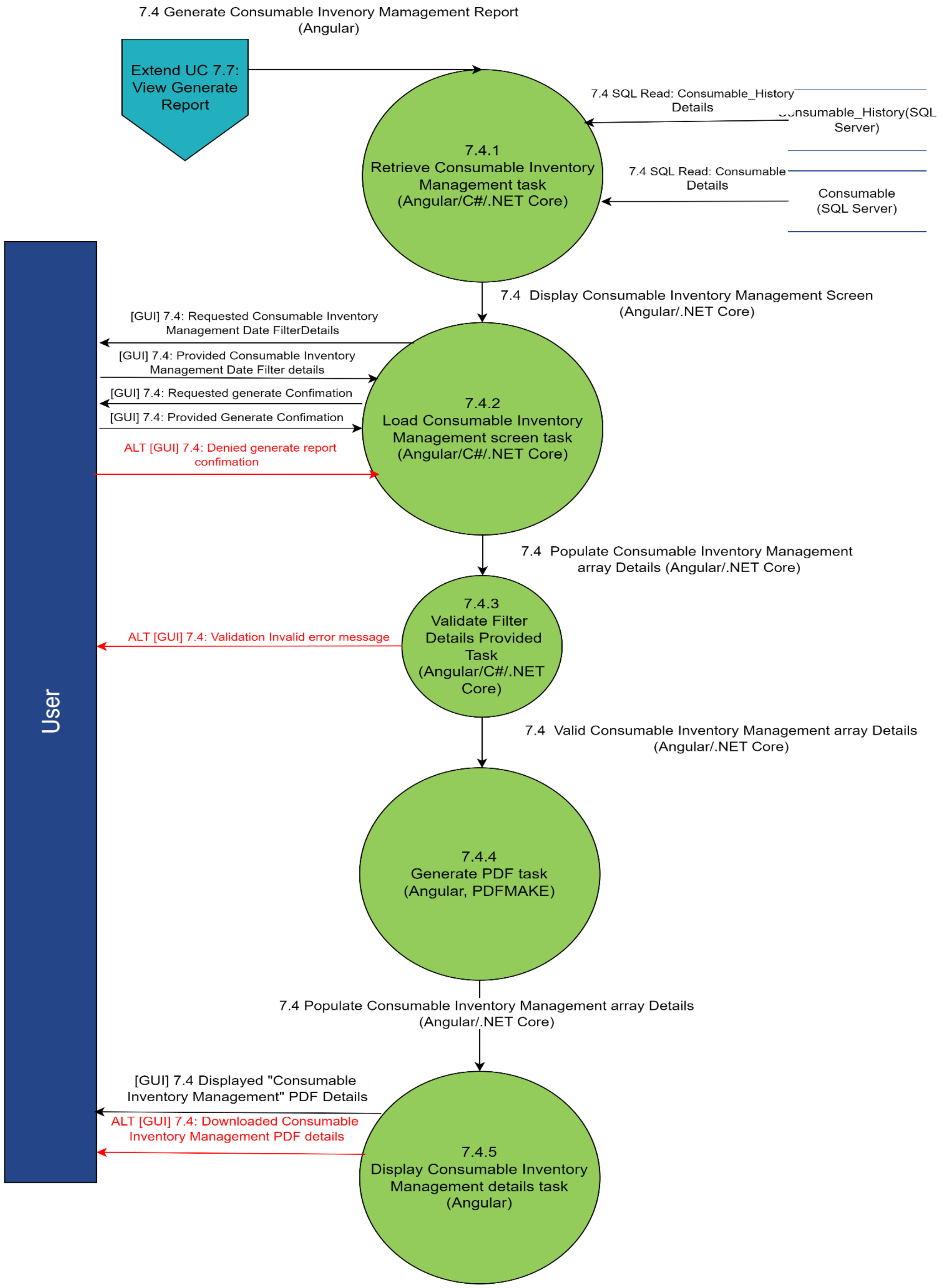


Figure 48 7.4 Generate Consumable Inventory Management Report Tech Prim

USE CASE 7.5

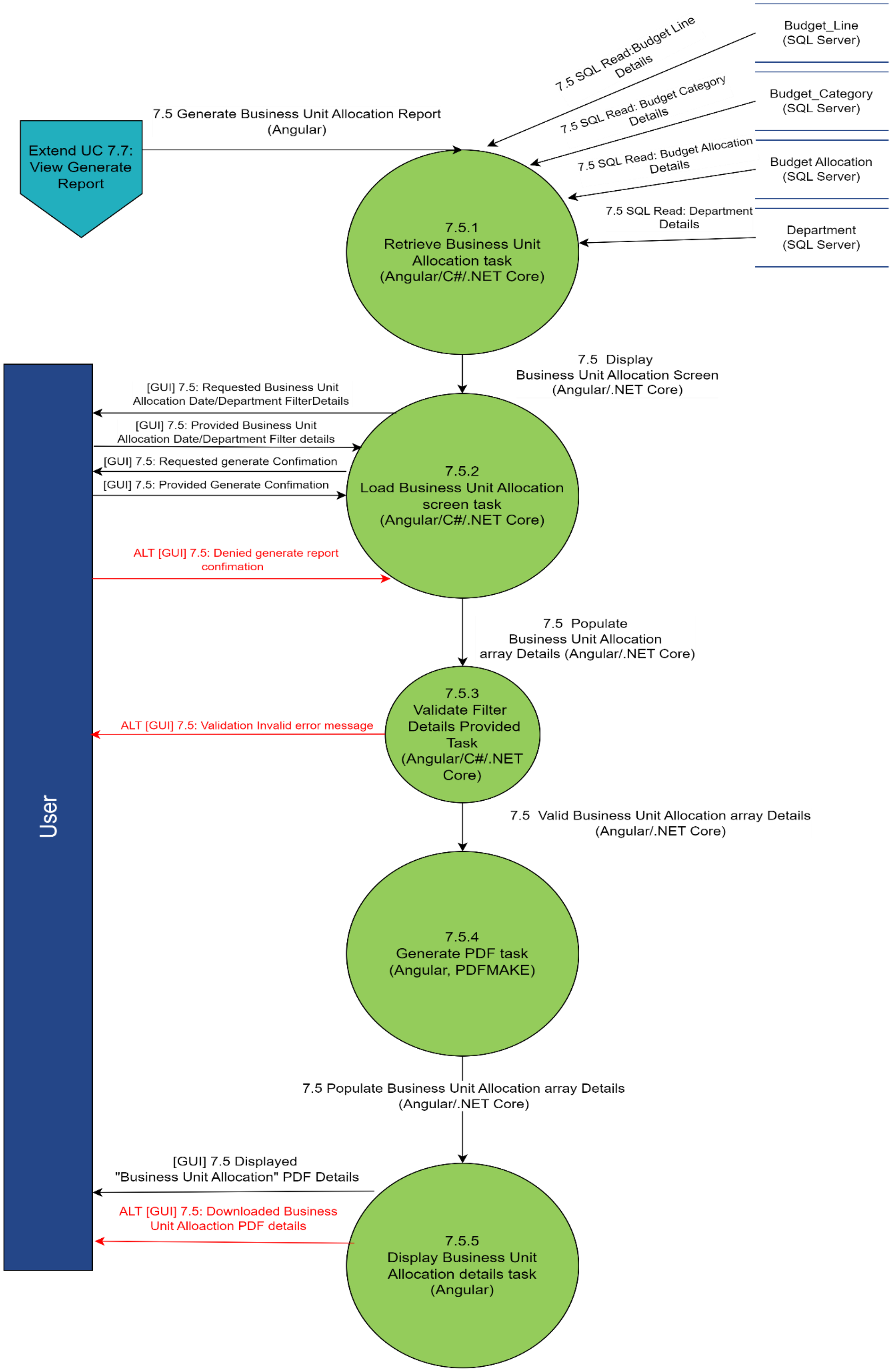


Figure 49 7.5 Generate Business Unit Allocation Report Tech Prim

USE CASE 7.6

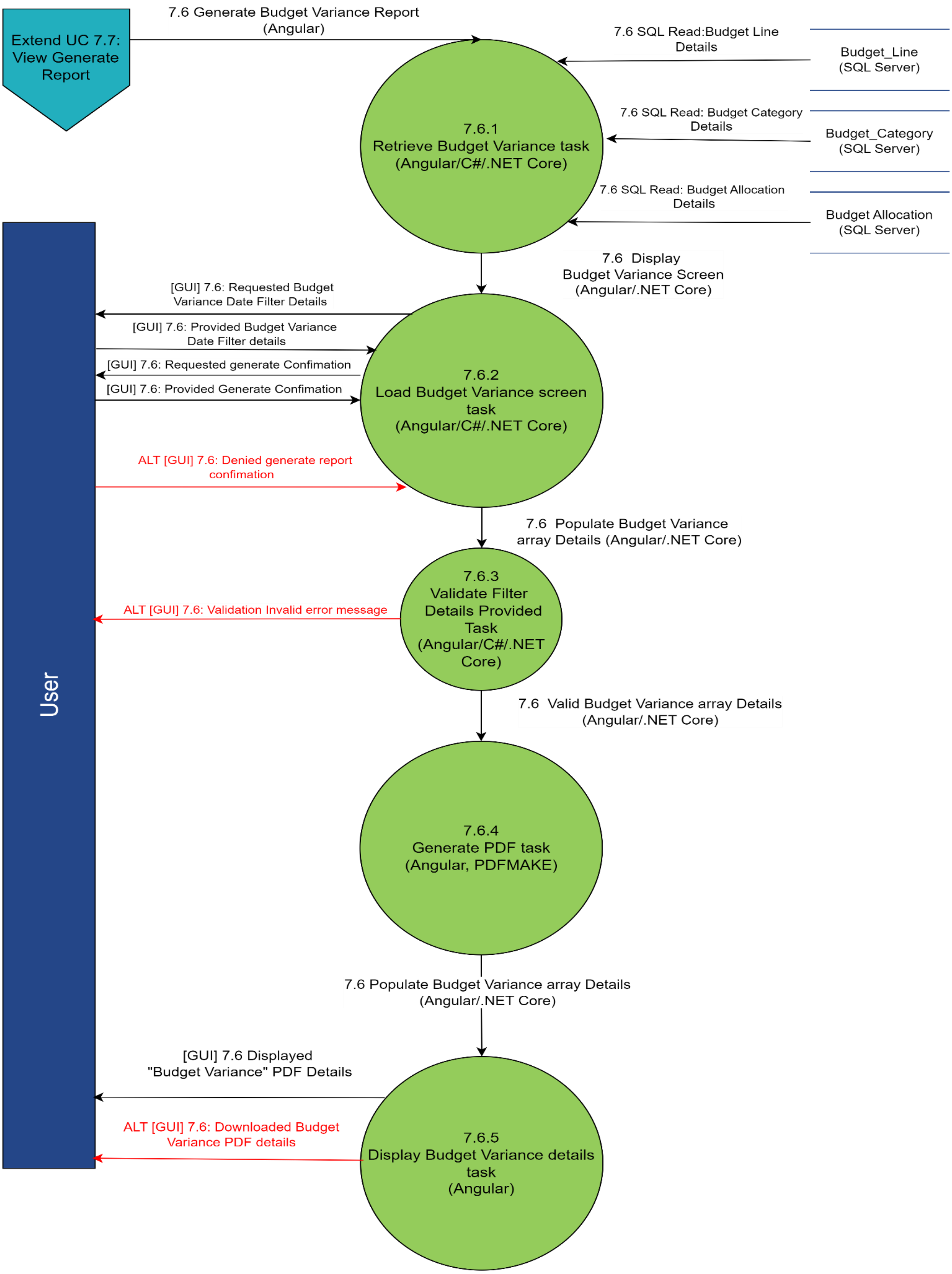


Figure 50 7.6 Generate Budget Variance Report Tech Prim

USE CASE 7.7

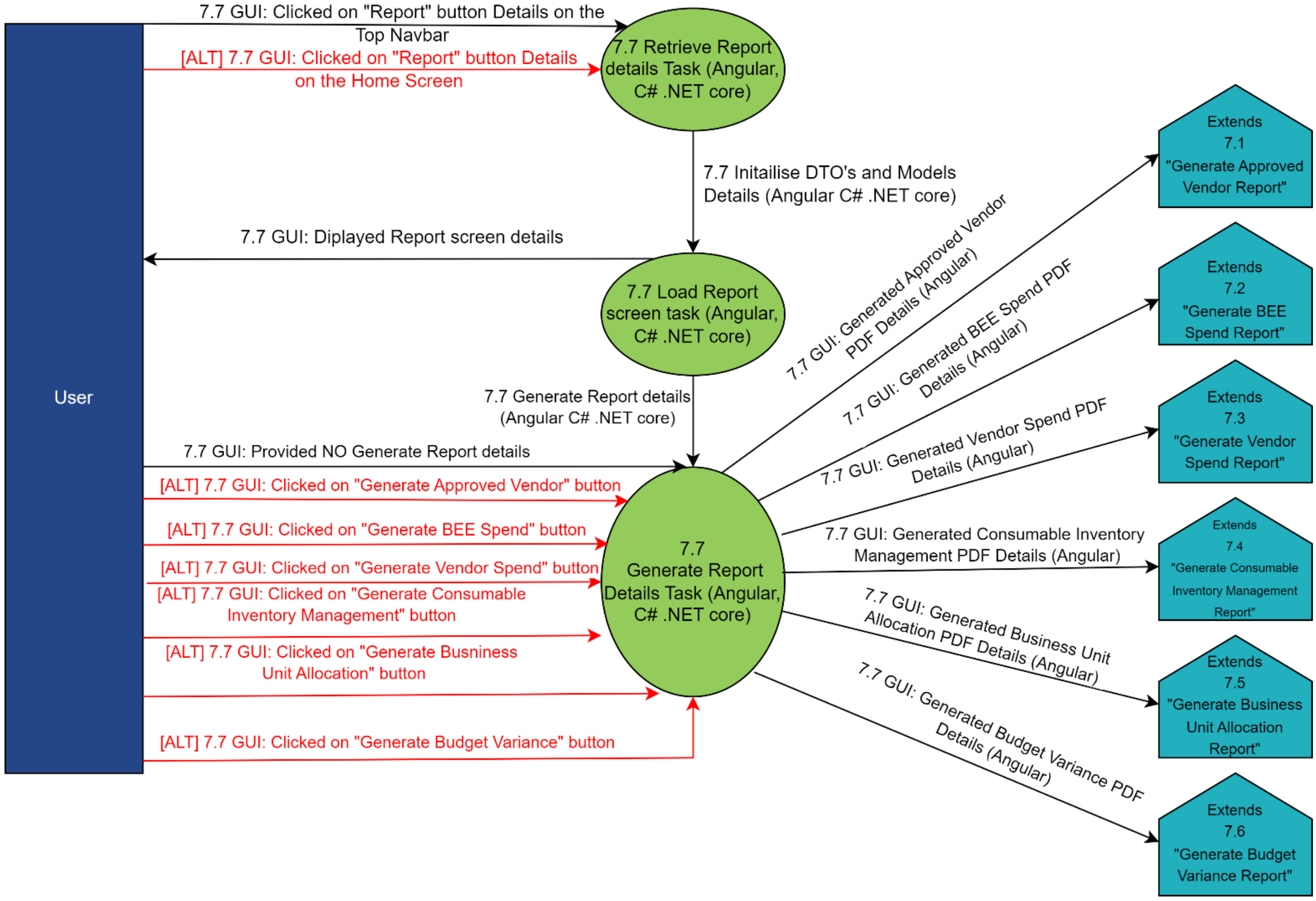


Figure 51 7.7 View Generate Report Tech Prim

7.3 CONCLUSION

This Concludes the visual representation of the Technical Primitive level diagrams for the Procion system.

8. ACTIVITY DIAGRAM

8.1 INTRODUCTION

In this section we will display the visual representation of the Activity diagrams for the following use cases: 2.26,2.37, 4.10, 5.9, 7.1-7.7, of the Procion system.

8.2 ACTIVITY DIAGRAM

USE CASE 2.26

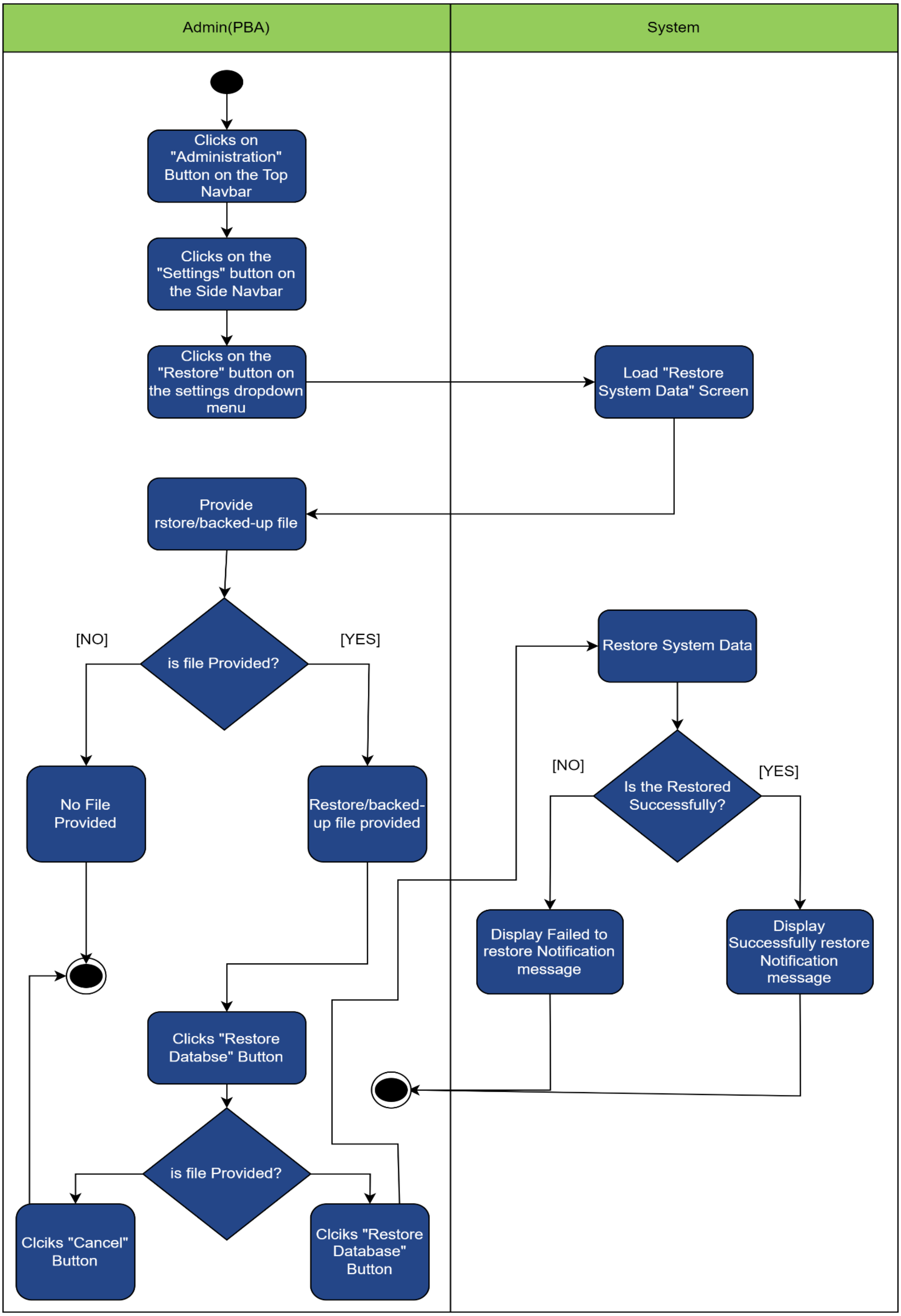


Figure 52 2.26 Restore System Data Activity Diagram

USE CASE 2.37

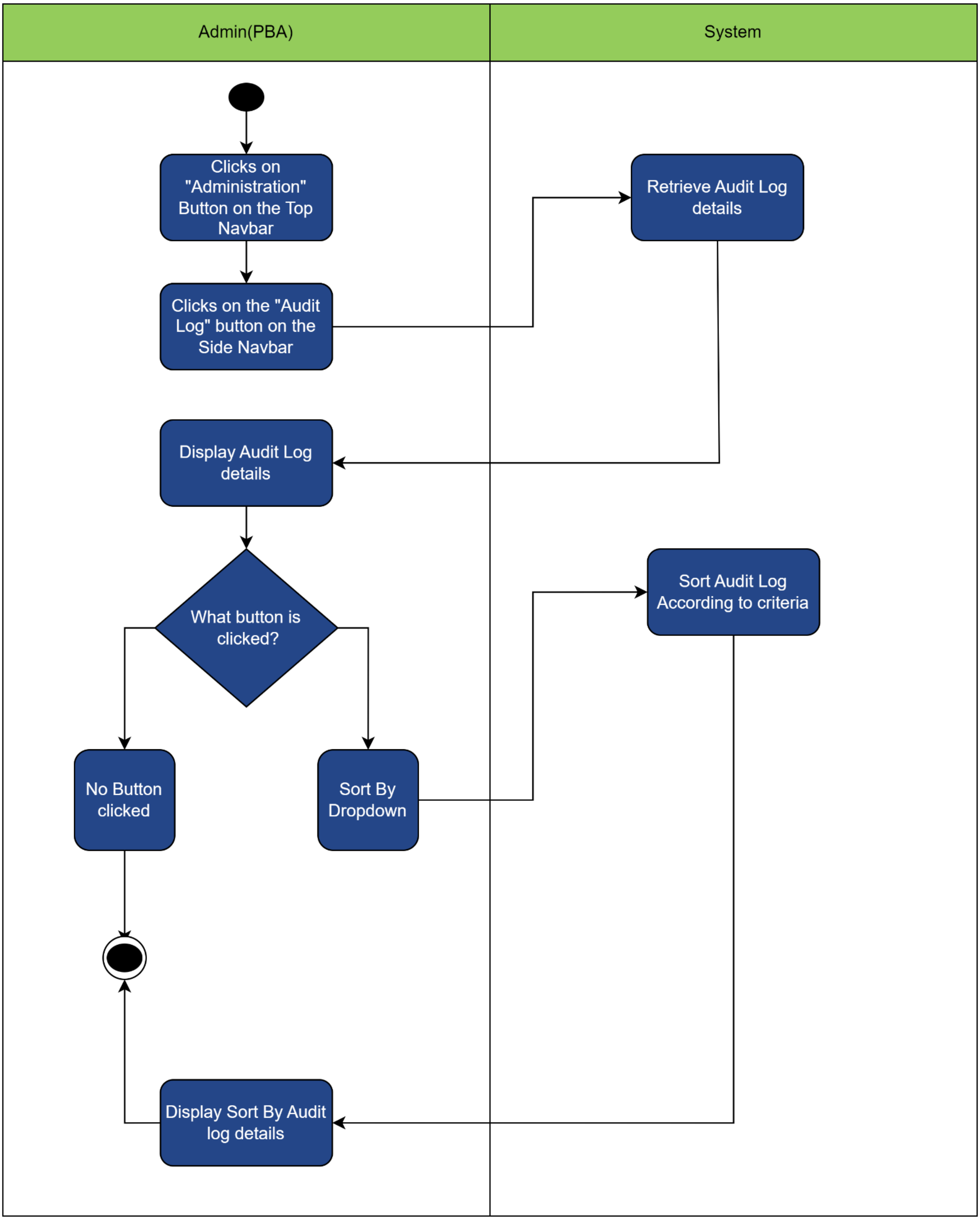


Figure 53 2.37 View Audit log Activity Diagram

USE CASE 4.10

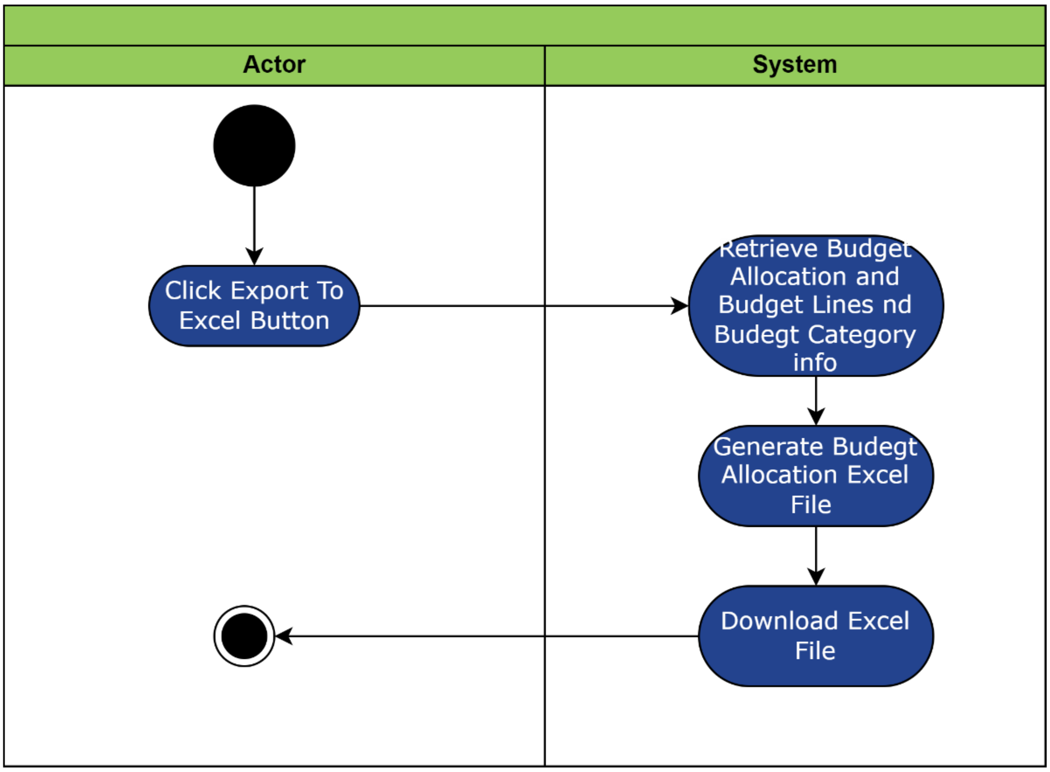


Figure 54 4.10 Export Budget Allocation Activity Diagram

USE CASE 5.9

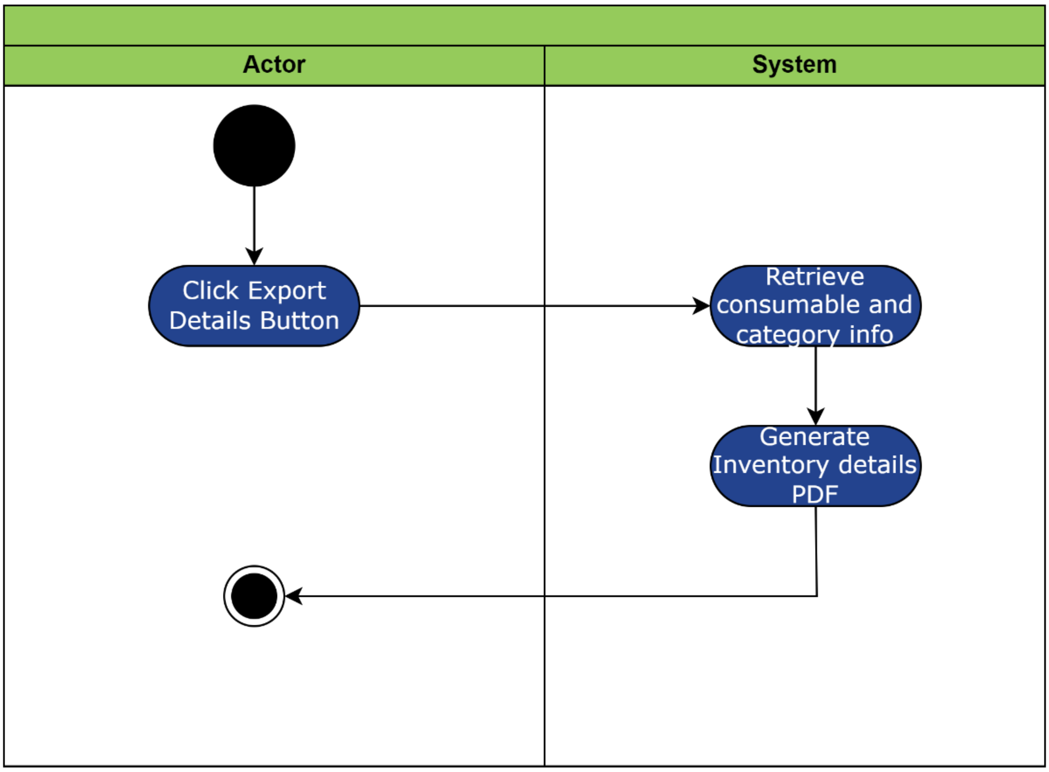


Figure 55 5.9 Export Inventory Details Activity Diagram

USE CASE 7.1

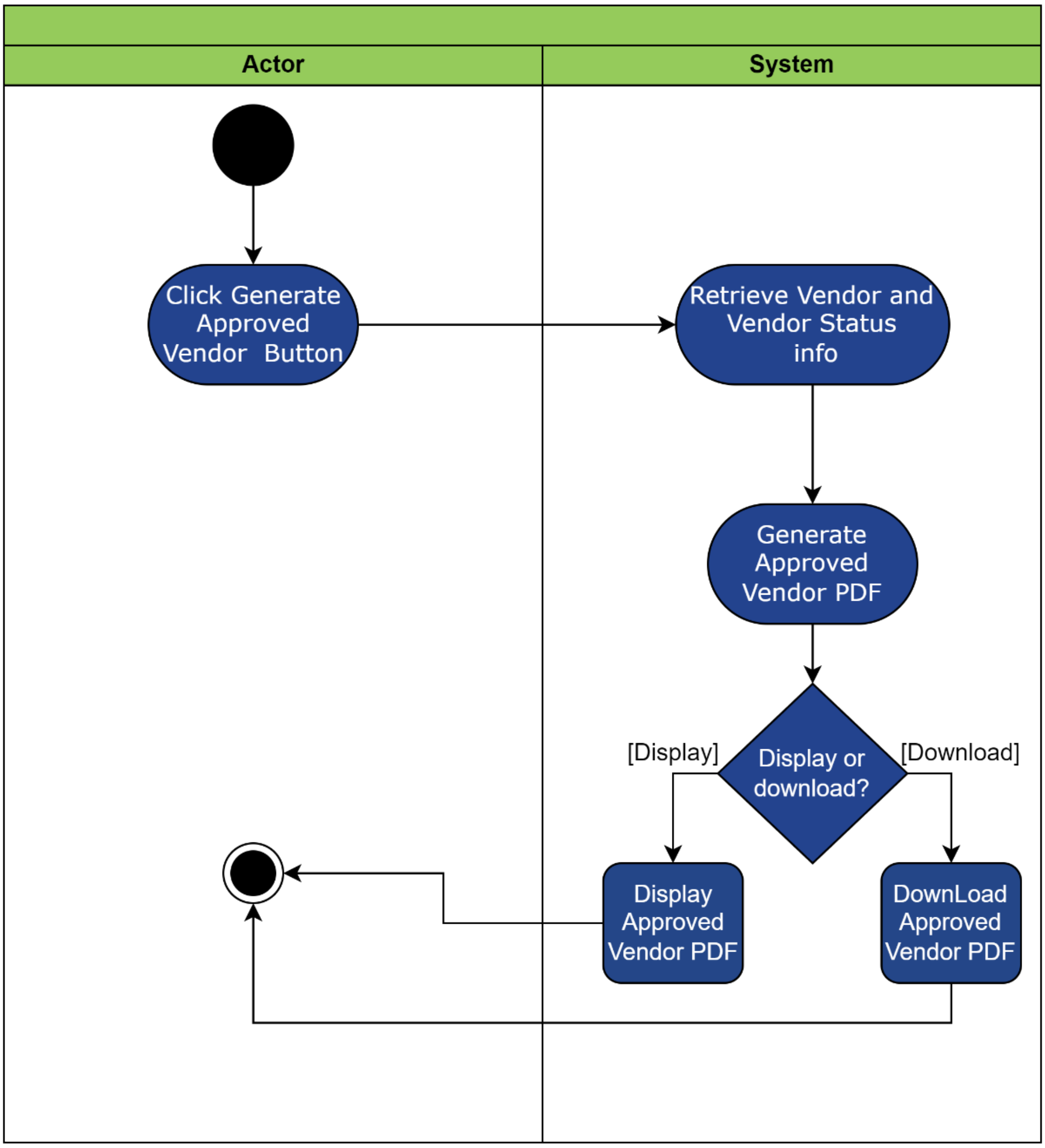


Figure 56 7.1 Generate Approved Vendor Report Activity Diagram

USE CASE 7.2

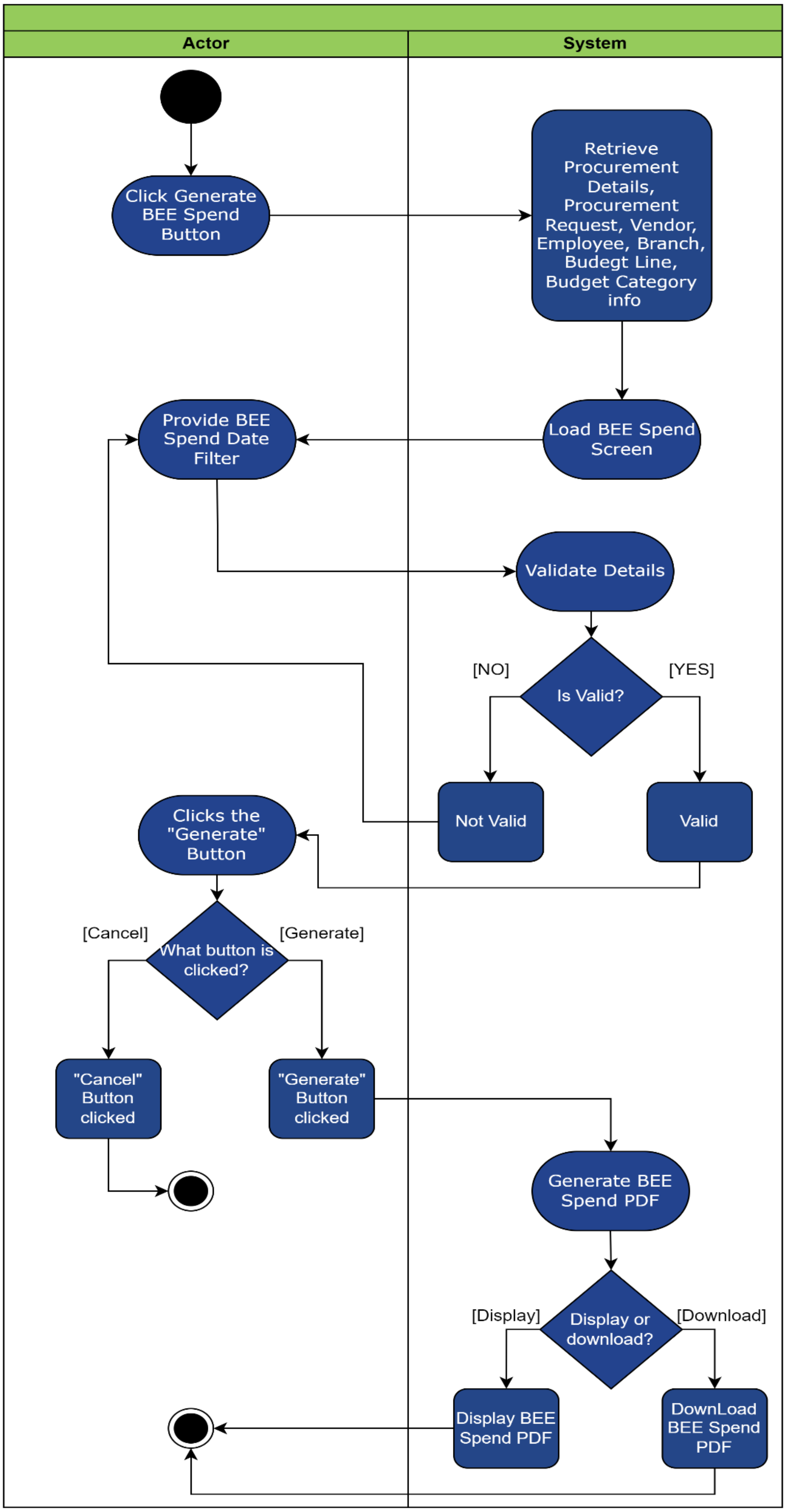


Figure 57 7.2 Generate BEE Spend Report Activity Diagram

USE CASE 7.3

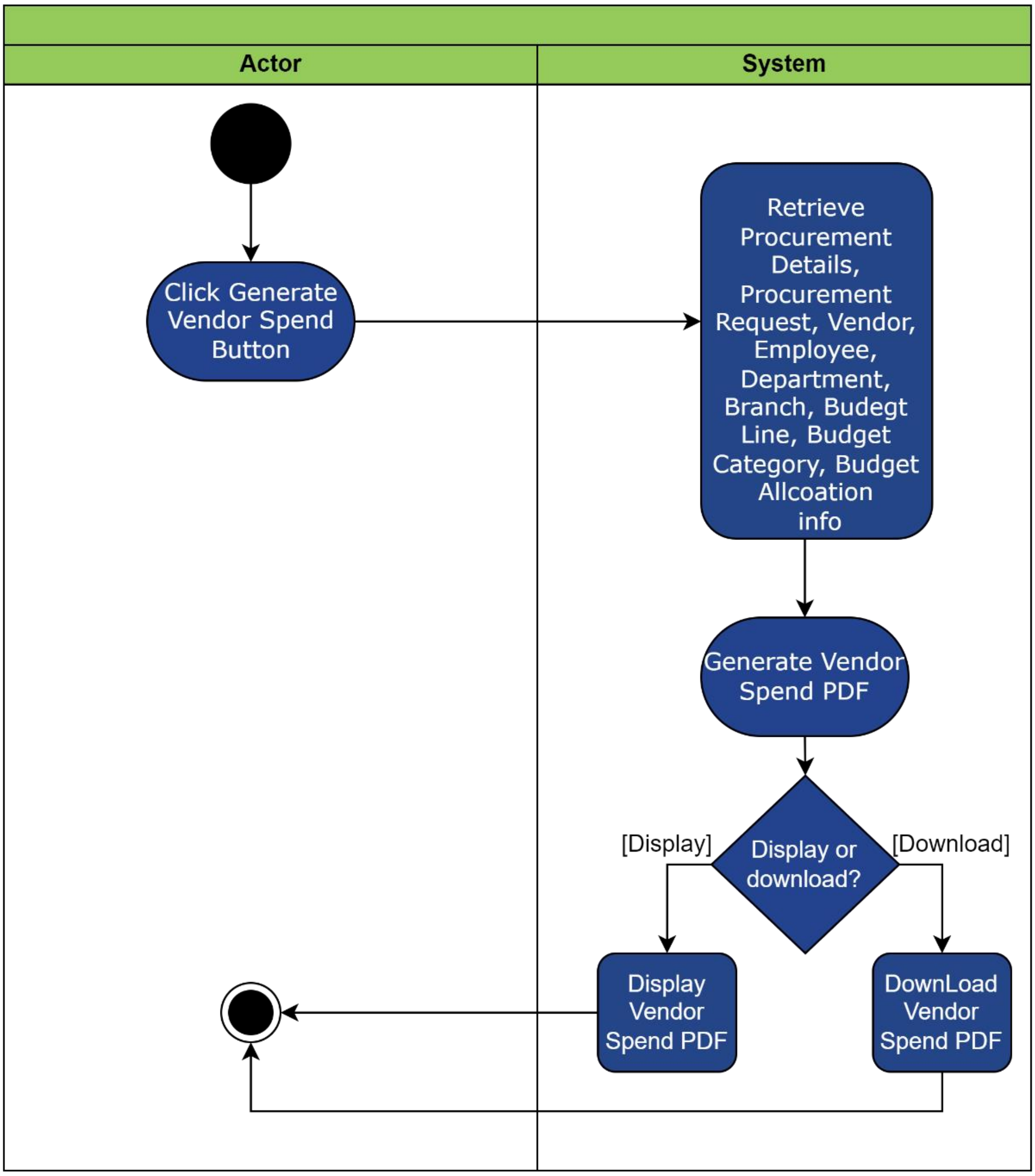


Figure 58 7.3 Generate Vendor Spend Report Activity Diagram

USE CASE 7.4

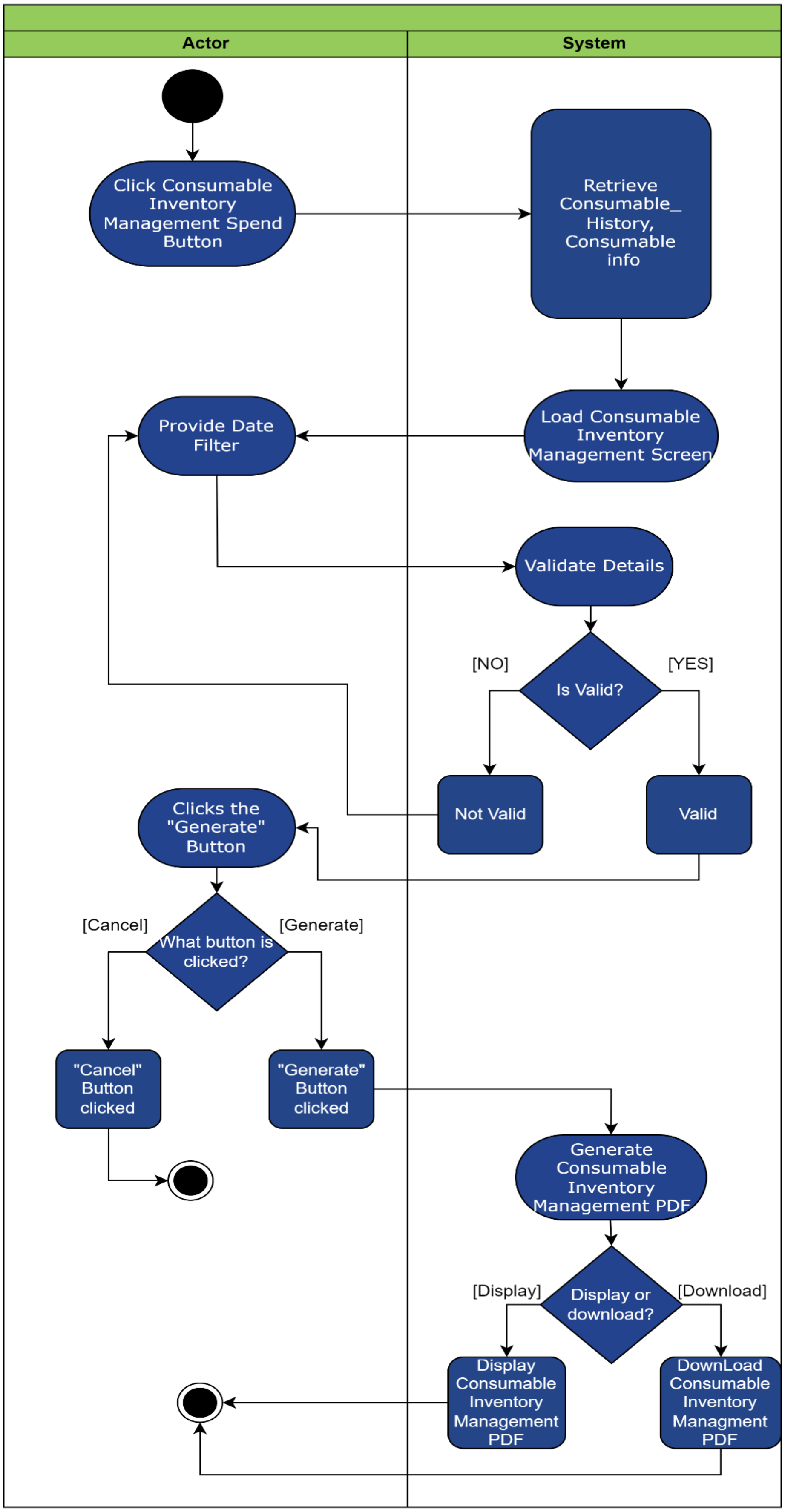


Figure 59 7.4 Generate Consumable Inventory Management Report Activity Diagram

USE CASE 7.5

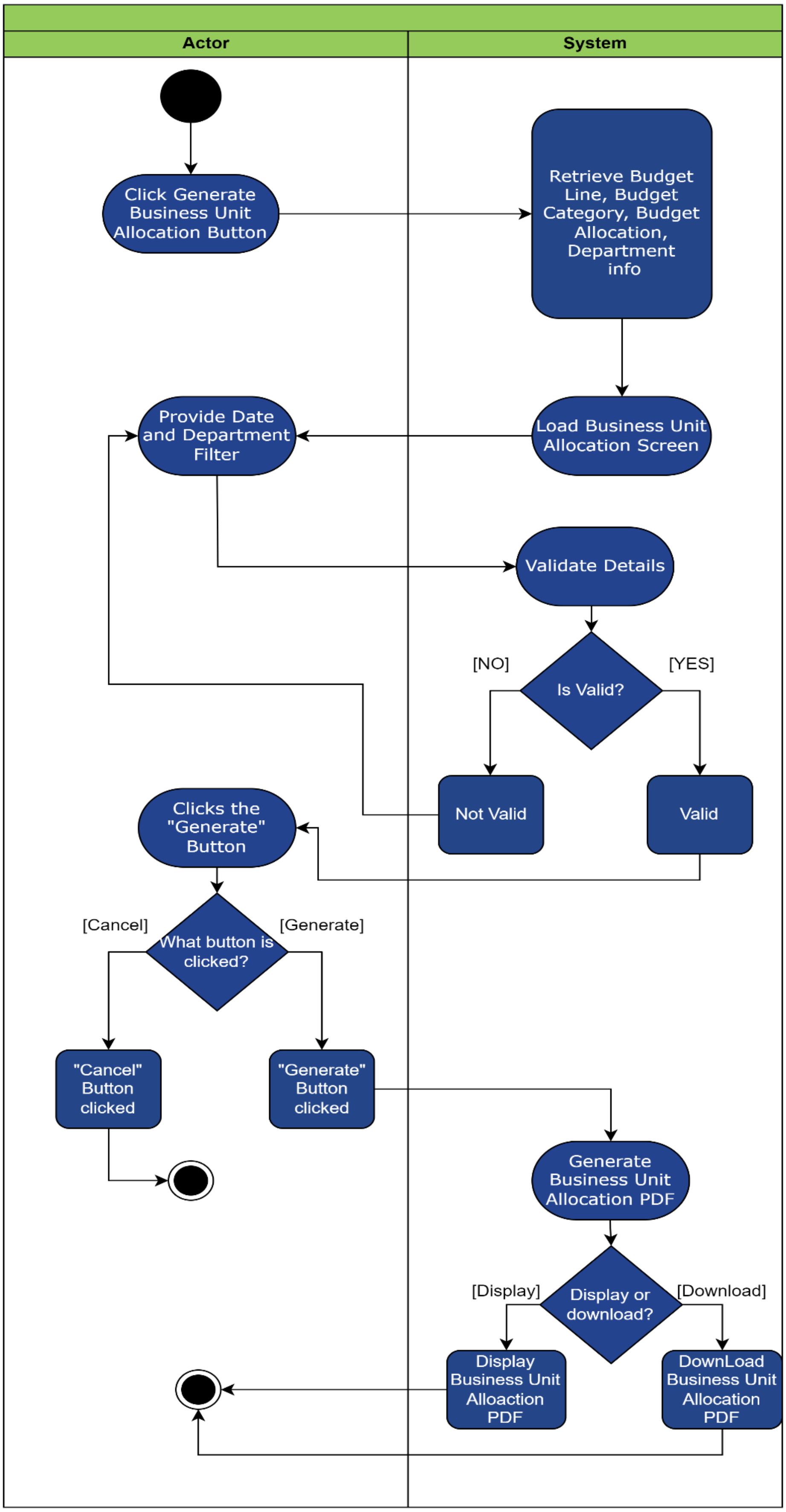


Figure 60 7.5 Generate Business Unit Allocation Report Activity Diagram

USE CASE 7.6

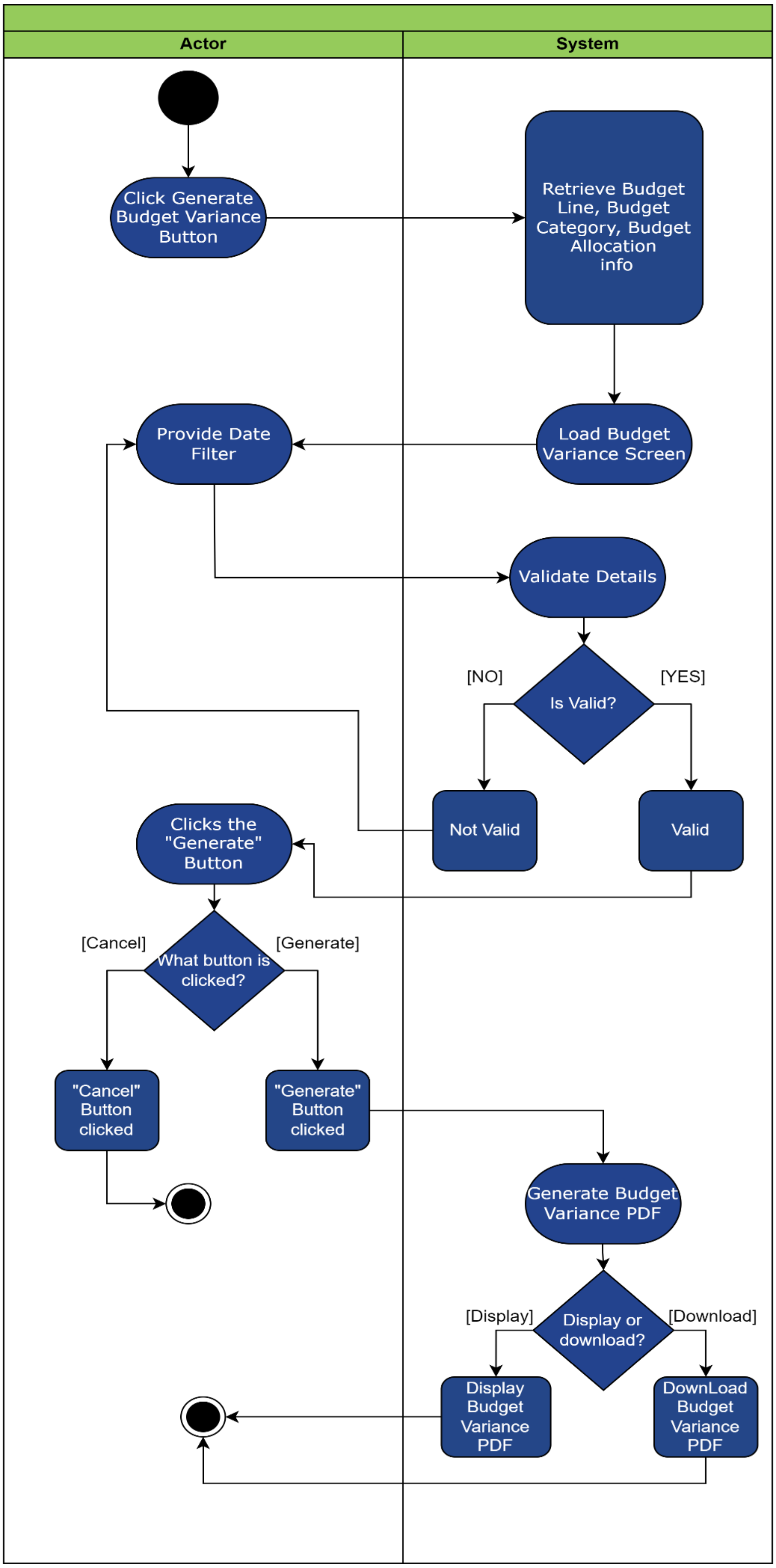


Figure 61 7.6 Generate Budget Variance Report Activity Diagram

USE CASE 7.7

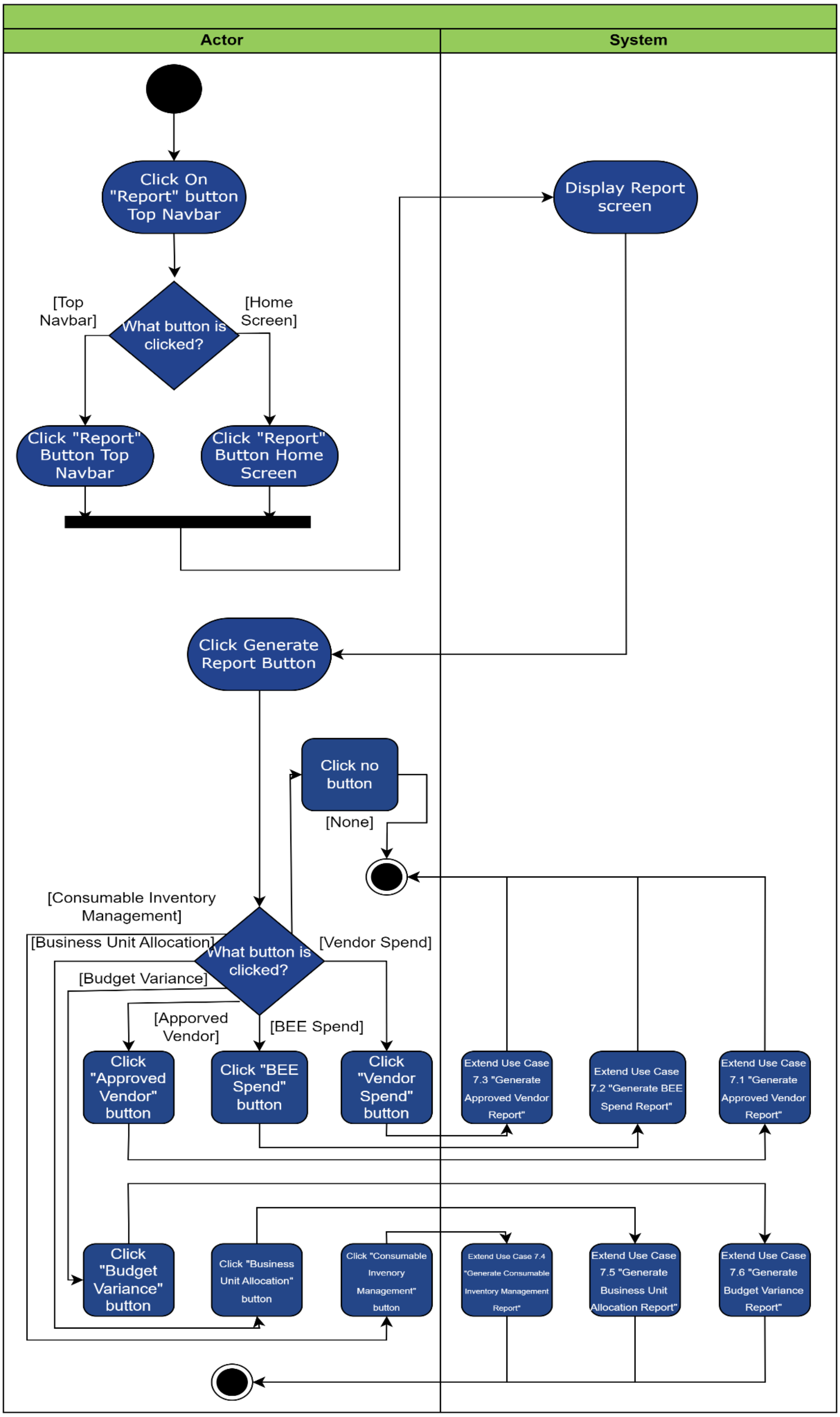


Figure 62 7.7 View Generate Report Activity Diagram

8.3 CONCLUSION

This Conclude the visual representation of the Activity diagram with for the respective use case diagrams for the Procion system.

9. SEQUENCE DIAGRAM

9.1 INTRODUCTION

In this section we will display the visual representation of the Sequence diagrams for the following use cases: 2.26, 2.37, 4.10, 5.9, 7.1-7.7 respectively, of the Procion system.

9.2 SEQUENCE DIAGRAM

USE CASE 2.26

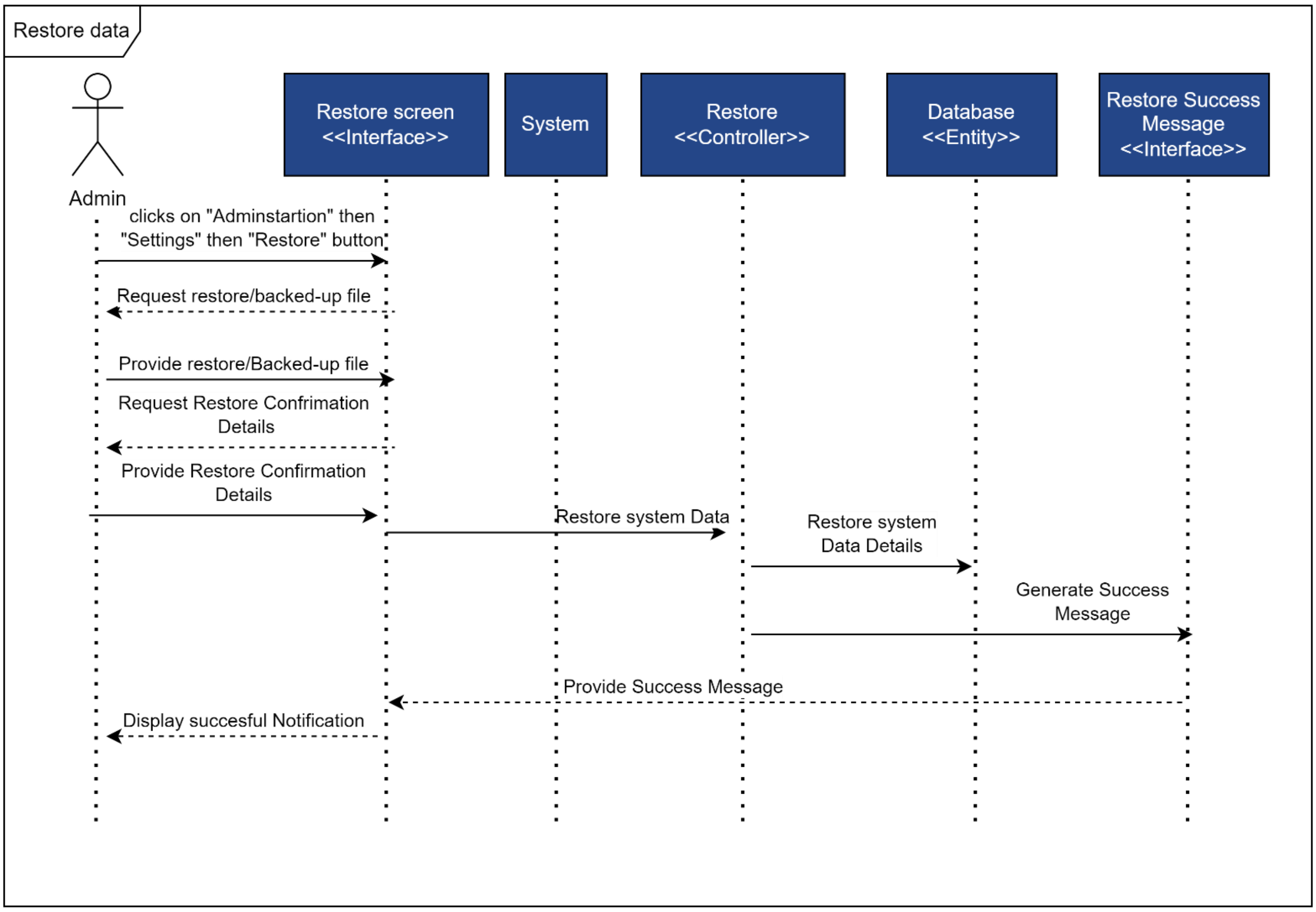


Figure 63 2.26 Restore System Data Sequence Diagram

USE CASE 2.37

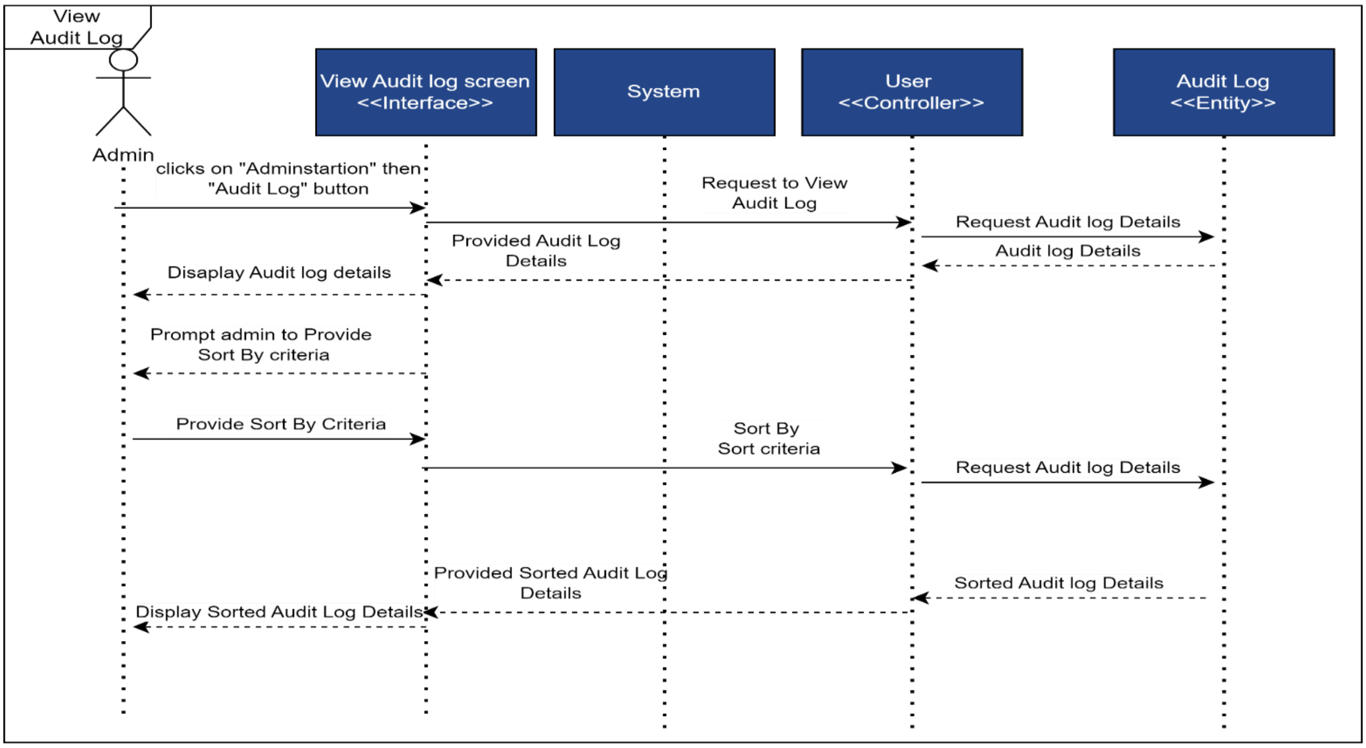


Figure 64 2.37 View Audit log Sequence Diagram

USE CASE 4.10

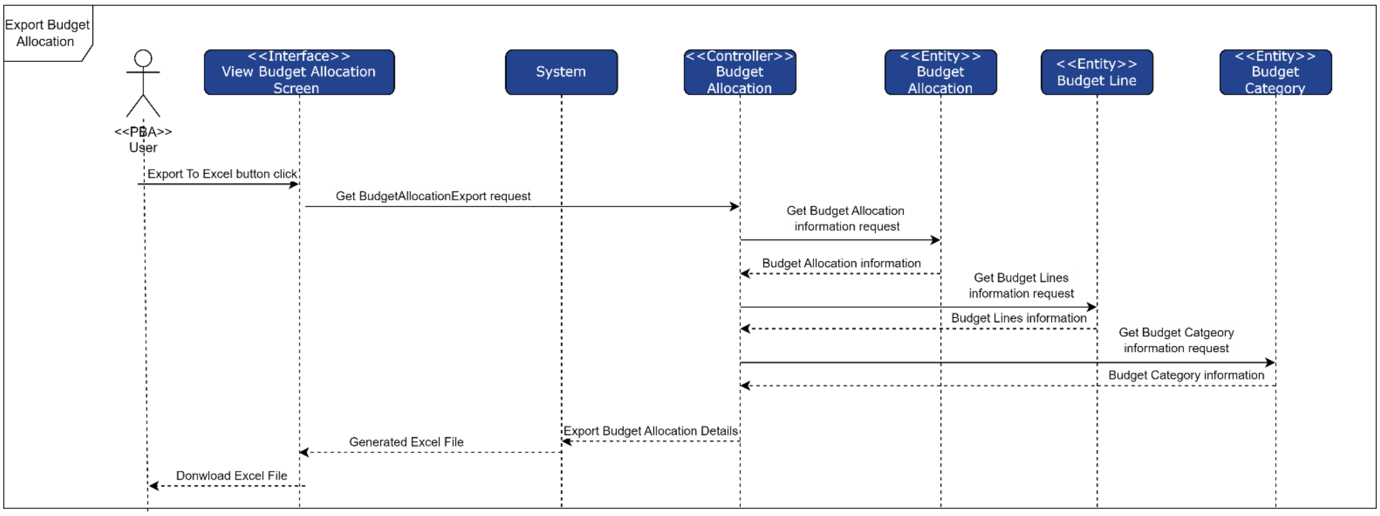


Figure 65 4.10 Export Budget Allocation Sequence Diagram

USE CASE 5.9

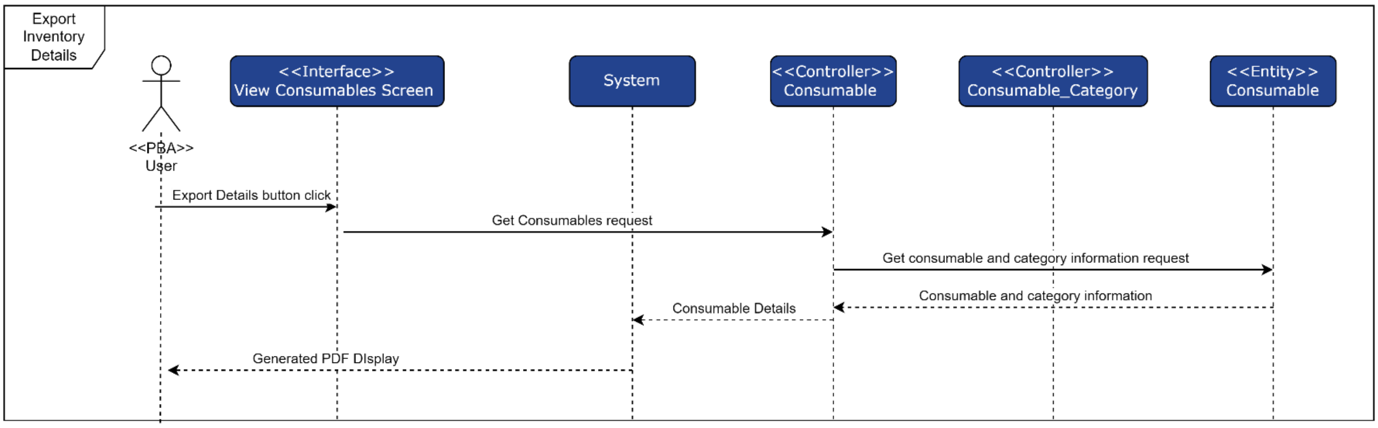


Figure 66 5.9 Export Inventory Details Sequence Diagram

USE CASE 7.1

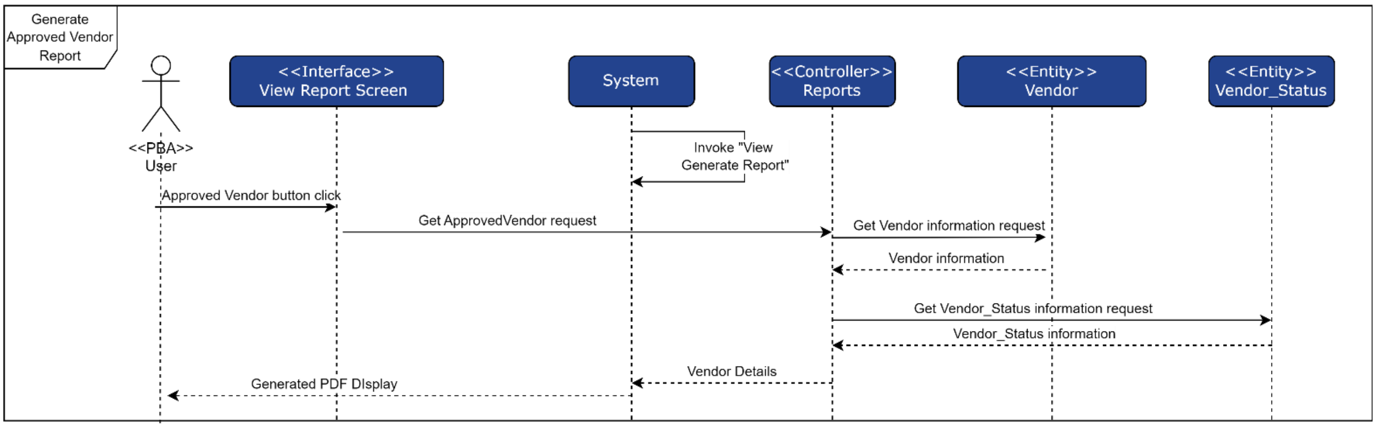


Figure 67 7.1 Generate Approved Vendor Report Sequence Diagram

USE CASE 7.2

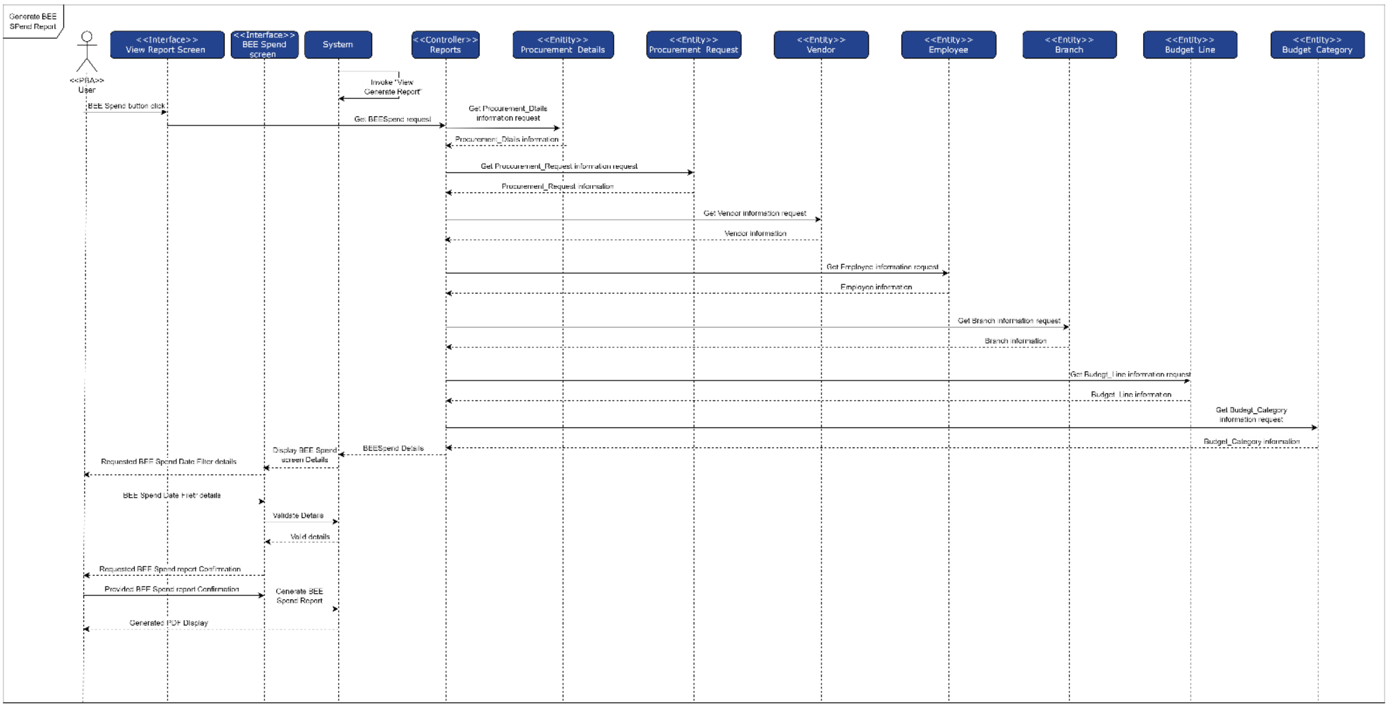


Figure 68 7.2 Generate BEE Spend Report Sequence Diagram

USE CASE 7.3

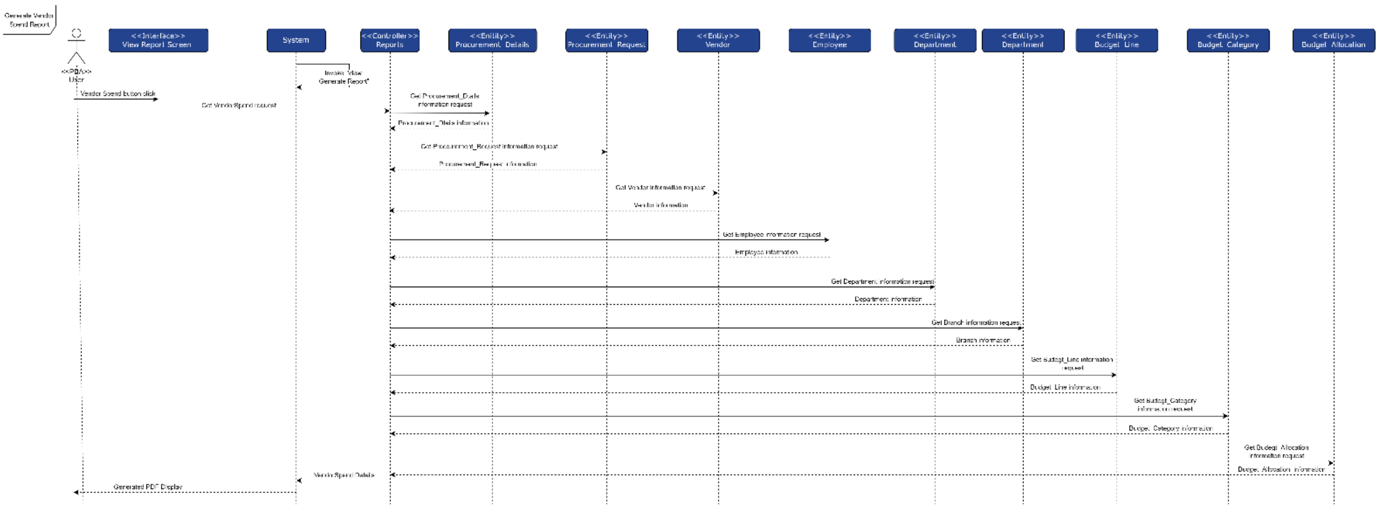


Figure 69 7.3 Generate Vendor Spend Report Sequence Diagram

USE CASE 7.4

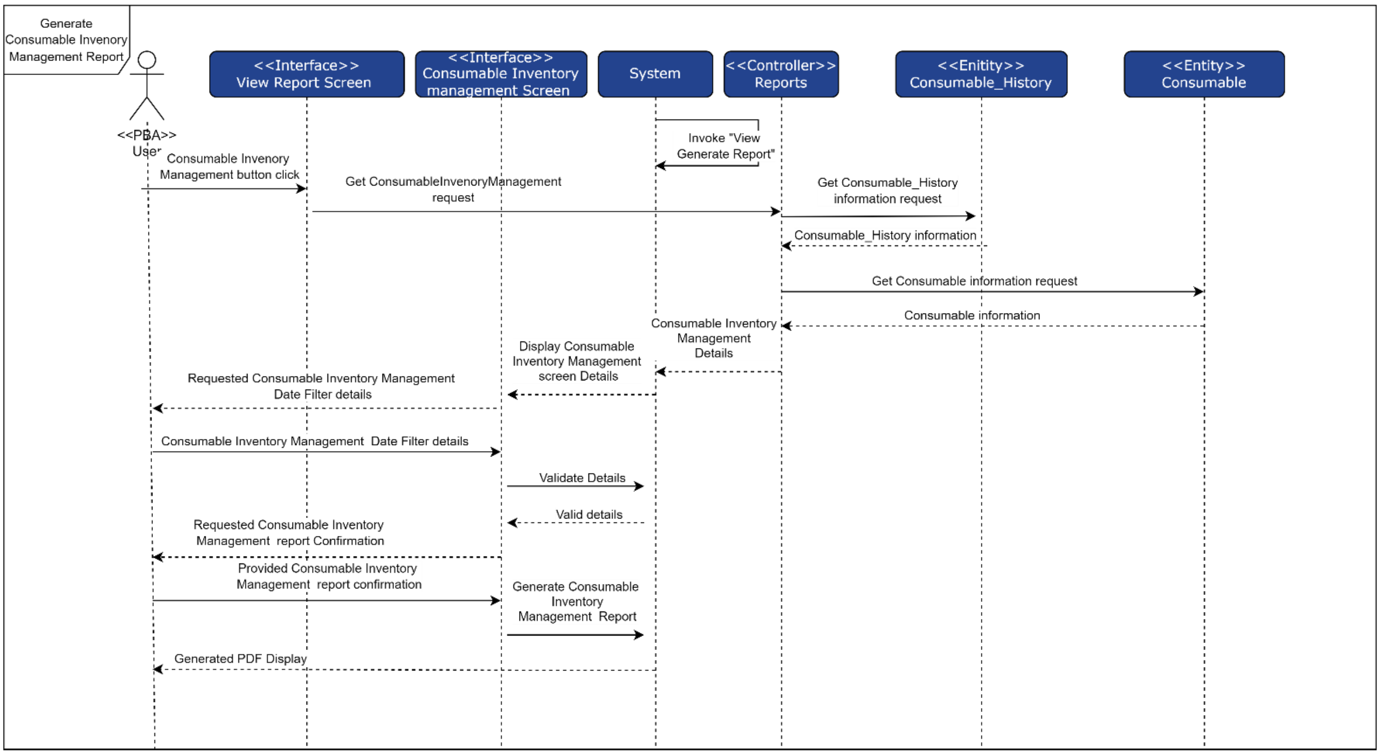


Figure 70 7.4 Generate Consumable Inventory Management Report Sequence Diagram

USE CASE 7.5

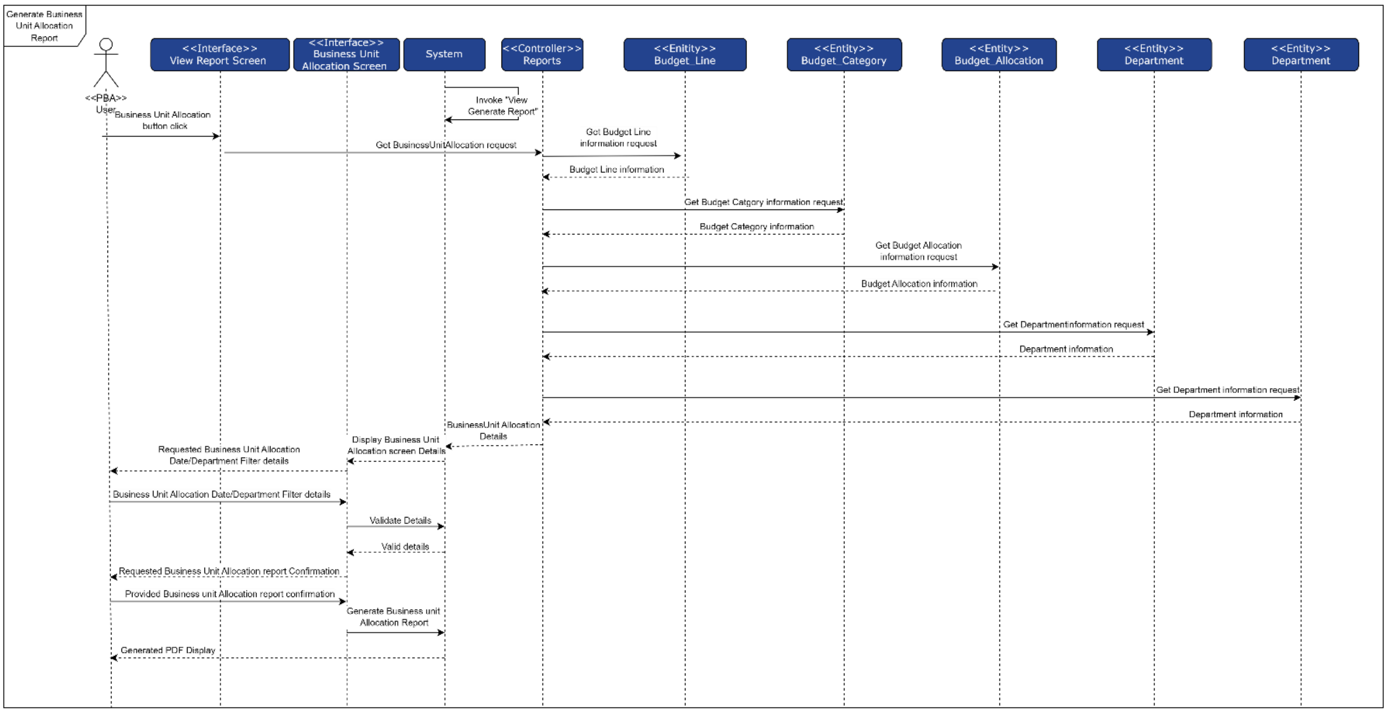


Figure 71 7.5 Generate Business Unit Allocation Report Sequence Diagram

USE CASE 7.6

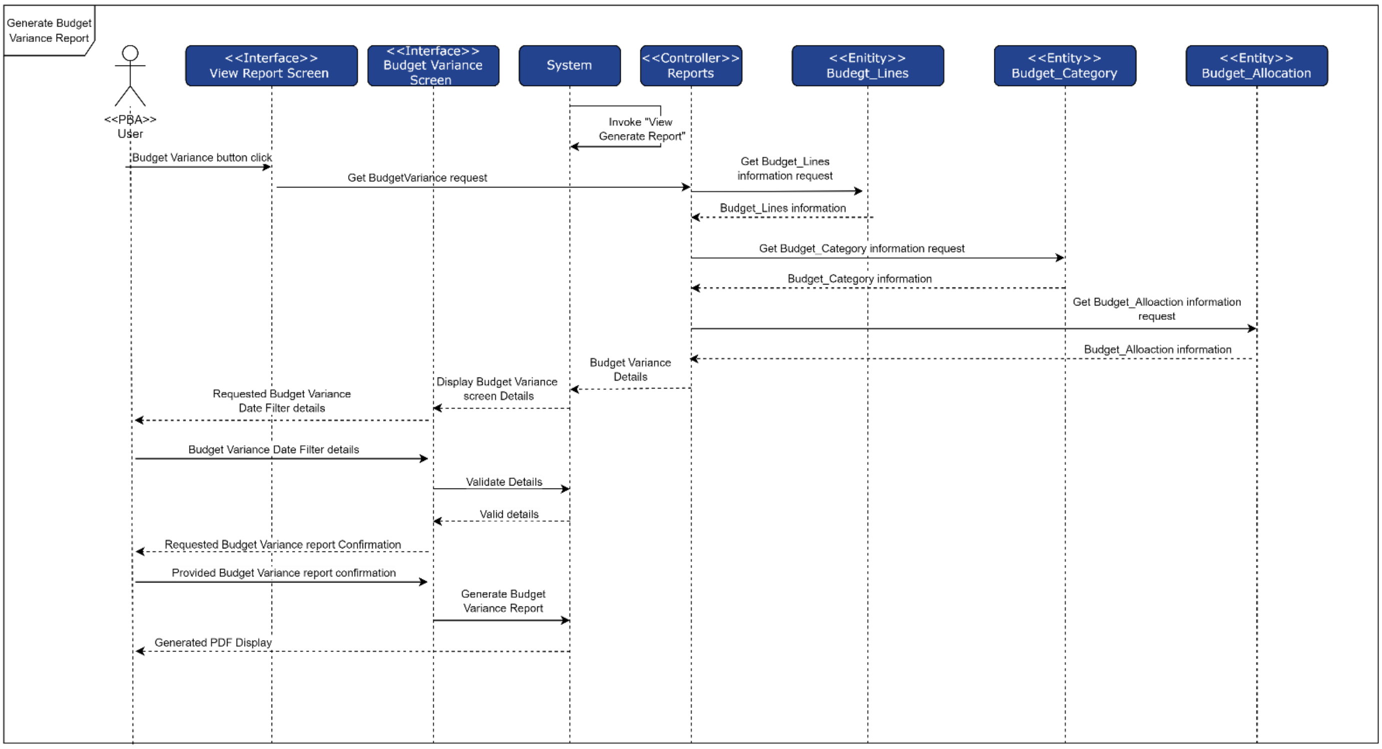


Figure 72 7.6 Generate Budget Variance Report v Diagram

USE CASE 7.7

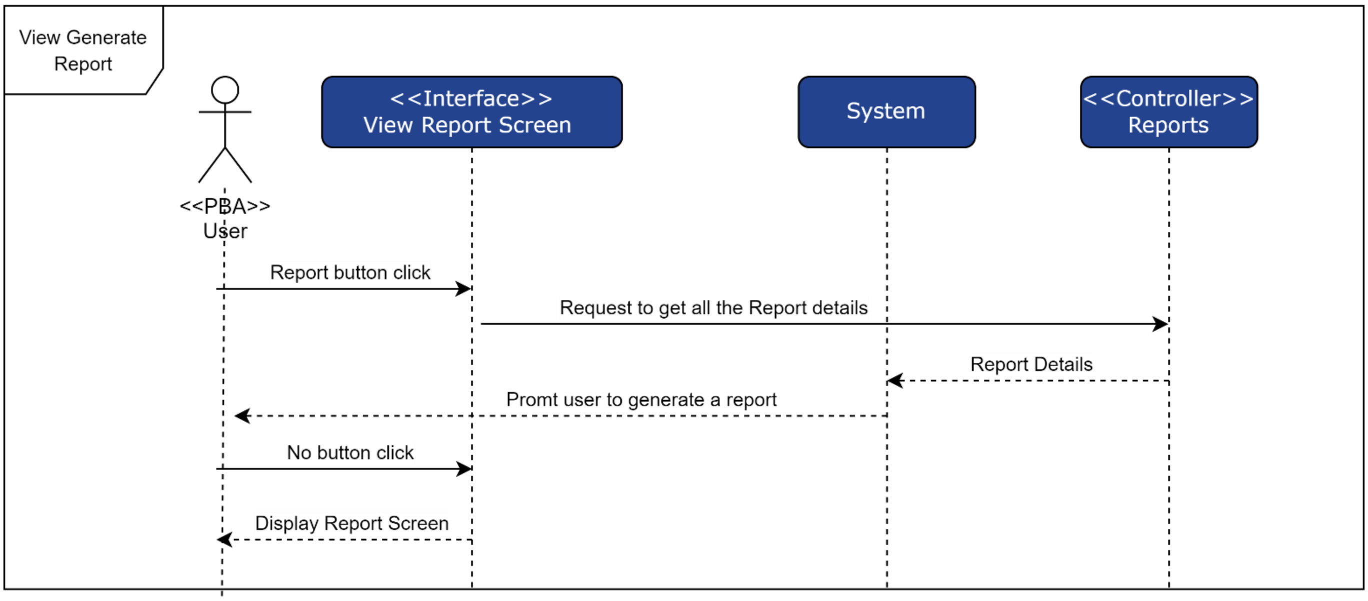


Figure 73 7.7 View Generate Report Sequence Diagram

9.3 CONCLUSION

This Conclude the visual representation of the Sequence diagram with for the respective use case diagrams for the Procion system.

10. UPDATED ENTITY RELATIONSHIP DIAGRAM

10.1 INTRODUCTION

This section goes in depth into a visual representation of our updated technical ERD. The ERD complies with referential integrity and is fully attributed as well as in 3rd normal form. This serves as an indication of how we will structure our database.

10.2 ENTITY RELATIONSHIP DIAGRAM

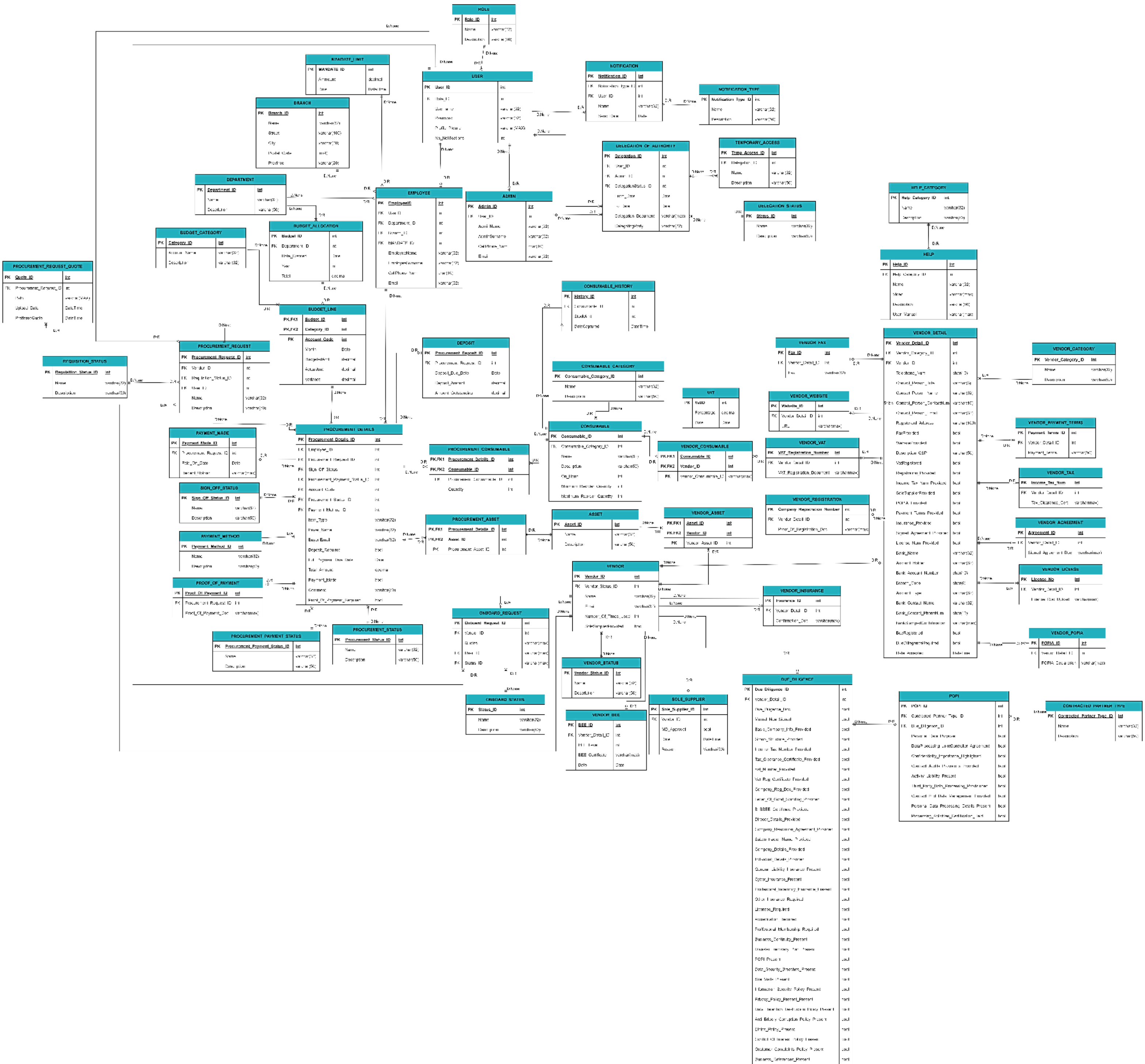


Figure 74 Technical ERD

10.3 CONCLUSION

This concludes the updated entity relationship diagram with all the entity tables and relationships between the entities within the system that we will develop.

11. STATE DIAGRAM

11.1 INTRODUCTION

In this section we will display the visual representation of the State diagram for the vendor subsystem.

11.2 STATE DIAGRAM

- NONE

11.3 CONCLUSION

This concludes the updated entity relationship diagram with all the entity tables and relationships between the entities within the system that we will develop.

12.UPDATED COMPLEXITY MATRIX

12.1 INTRODUCTION

In this section we show the updated complexity matrix with all the technology that we will be using, the Procion system. This shows each technical requirements that the system will have and the Total complexity mark of the system.

12.2 COMPLEXITY MATRIX

INF 370

mplexity Matrix Mark Sheet

Topic	Level		Marks	Indicated by group	Marks given	MAX	Total
	For online applications: Responsive web design		3	3		42	0
	For desktop applications: Form design according to design principles (Schneiderman's golden rule on navigation applies here)	*		3			
	Appropriate use of grids/tables		3	3			
	Appropriate use of tabs/links		3	3			
	Use of graphs in an appropriate business context		4	4			
	The storage and display of graphical information, like photos with a good business reason		3	3			
	Working e-mail automatically generated from the database in an appropriate business context		2	2			
	SMS messages automatically generated from the system in an appropriate business context		2				
	Extensive user-friendly search facility		3	3			
	At least one use of a tree to display data from the database		3	0			
	Able to dynamically modify a data tree structure and in doing so adjusting the data in the database		4	0			
	At least one use of a calendar view of data (not a date/time picker; not a plug-in such as Google calendar)		3	3			
	Uploading a file into the system with appropriate business reason		3	3			
	The use of audio/video in an appropriate business context		3	0			
1. Special GUI	At least one use of an administrator configurable timer in an appropriate business context		3	3			
	At least 30 tables used (4 member groups) or 40 tables used (5 member groups)	*	6	6		15	0
	Full referential integrity on all tables	*	6	6			
2. Database access	At least one use of master-detail table relationships (Schneiderman's golden rule on system status applies here)	*	3	3			
	At least 3 simple list reports in a reporting tool (no control breaks, no graphs, single table)	*	3	3		15	0
	At least 2 transactional report with 2 or more control breaks (with heading and calculated values/totals, multiple tables)	*	6	6			
	At least 1 report with adjustable criteria		3	3			
3. Reports	At least 1 management report using a graph		3	3			
	hard coded but maintained in a sub-module of the		6	6		12	0
4. Flexibility	maintained in a sub-module of the system.		6	6			
	consistent, user-friendly error messages are		6	6		12	0
5. Error handling	Appropriate data validation on **all** input fields		6	6			
	up a complete help document (HTML, PDF, Help-file)		3	3		15	0
	a specific screen/function will automatically open		6	6			
	Search Facility on Help		3	3			
6. Help	Extensive use of hints		3	3			
	user profiles		3	3		13	0
	business reason.		3	3			
	Encrypted passwords in database	*	1	1			
7. Security	user profiles that will enable/disable access to		6	6			
	showing at least date, time, user, transaction type,		6	6		9	0
8. Audit Trail	following: date, user, transaction type		3	3			
	installation disks that take care of application		3	0		15	0
	application to a publicly accessible web server		3	3			
	Market place (such as the PlayStore or the		6	0			
9. Deployment	server		3	3			
10. Backup and Restore	backup/restore all data (system may exit during		3	3		3	0
	data in it based on the parameters provided (with a		6	6		9	0
11. Import/Export Data	Importing or Exporting of data (with good business		3	3			
	and-play technology, such as a swipe card reader,		3	0		18	0
	specific software. Data or images must seamlessly		6	0			
12. External INPUT device	device specific software. Data or images must		9	0			
13. External APPLICATION / Services	application (with good business reason)		3	3		9	0
	application system exists and the interface must be		6	0			
	alternative platform.		3	3		27	0
	platform (i.e. data is displayed from the system's		3	3			
	alternative platform (i.e. both reading and writing		9	3			
	onto the system's database.		3	3			
14. Multiplatform processing for an appropriate business reason	reading and writing data from the system's		9	9			
	between your database and your business layer		3	3		12	0
	your business layer and your presentation layer		6	6			
15. Programming Principles	triggers and/or jobs.		3	3			
system	system (e.g. machine learning, AI, block chain, text	#	1-9	9		9	0
	Complexity Marks Obtained			**187**	0	235	0
	Maximum Complexity Marks		**235**				
	Complexity Marks Required for Del 5 (5 members in team)		150				
	Complexity Marks Required for Del 5 (4 members in team)		130				

12.3 CONCLUSION

This concludes the updater complexity matrix section of this document.

12. DOCUMENT CONCLUSION

This concludes the iteration 6 document regarding the Design and the development of the following Use cases: 2.26,2.37, 4.10, 5.9, 7.1-7.7 of the Procion system that we are developing. Here we try and gather insight about the third update functional requirements, Logical Use Case Narratives, Logical context diagrams for those narrative, Screen designs for those narrative, Technical Use Case Narrative, Technical Context diagram for those narratives, Technical primitive level diagram for those narrative, Activity diagram for those narratives, Sequence diagram for those narratives, State diagram for those narratives(only complex narratives that has a change in state involved), and Demonstrate Functionality of the use cases, and the updated complexity matrix with the updated ERD to help us achieve the start for the development of the Procion system. This will bring us closer towards our end goal of building the Procion system.

13. TEAM SIGN OFF

PROCION SYSTEM: Design & Development

We, team 11, understand and acknowledge, have reviewed the information contained in the document (Technical Specifications), and we confirm and understand the contents and that the information provided is true.

BY SIGNING BELOW, WE AGREE TO THE FOLLOWING:

1. We understand the goals and the objectives of the project, as well as the specific tasks and responsibilities assigned to us.
2. We agree that our assigned tasks have been completed and meet the required quality standards.
3. We agree that any issues, risks, and concerns have been appropriately addressed and resolved.
4. Acknowledge that any changes or revisions to the document have been made and agreed upon by the project team.
5. That each team member has contributed equally to the iteration

Signed by:				
Name: Jason vd Merwe	**Name:** Leon Combrinck	**Name:** Bupe Chindongo	**Name:** Werner Schutte	**Name:** Emil Wonigkeit
Date: 2023-07-30	**Date:** 2023-07-30	**Date:** 2023-07-30	**Date:** 2023-07-30	**Date:** 2023-07-30
Signature: 	**Signature:** 	**Signature:** 	**Signature:** 	**Signature:** 

14. CLIENT DOCUMENT SIGN OFF

PROCION SYSTEM: Design and Development

Business Name: MOYO Business Advisory

I,_____________________, Hereby confirm that I have read the document and agree that the information contained therein is complete, accurate and meets the requirements of the project.

BY SIGNING I CONFIRM THAT:

1. **I have reviewed the deliverable as outlined in the project.**
2. **I accept the deliverable as complete and satisfactory.**

Signed by:
Name:
Date:
Signature: