REQUIREMENT ELICITATION AND ANALYSIS

EWBFS SYSTEM

October 29, 2022

PREPARED BY

Application Development and Quality Assurance Division, Information System Department

Enat Bank Sc.

Table of Contents

CHAPTER ONE

INTRODUCTION

1.1 Overview

This document is a software requirement specification and complete description for EWBFS system .it is organized into three sections. The first section describes the introduction, the scope, limitations of the system to be developed, objectives of the project, and methodologies. The second section describe what the proposed system can do, functional and non-functional requirements of the new system. The final and the third section cover designing of the system model.

1.2 Problem Definition

In our bank, EWBFS is currently done manually using excel files. So, this leads time-consuming to follow the workflows and customers data management issue.

1.3 Purpose

This document aims to provide a brief description about a proposed solution, web-based loan follows up system.

1.4 General Objective

The aim of this project is automating a web-based application that allows users of the system to facilitate there works and remove anomalies.

1.5 Specific Objective

- To maintain and manage products
- To maintain and manage customers
- To allow users of the system to follow the status of each loan and saving-related requests from branches
- To enable requests’ assignment to the respective approval committee heads
- To enable request escalation if the task is not completed with a set of standard days of request assignment.

1.6 Methodology

The project was proposed by the women banking solution department and Branch employees to address customers financial problem, deliver their service and follow and their customers saving and loan statuses. We get the requirements from the shared excel sheet and yenta gebeta product approved documents. After the system definition, the next step will be to bring the system that was only in our head to ground. In order to specify the required components, the project was divided into small modules and a comprehensive plan was set up. Loan follow up system comprises: A web application that stores and retrieves the necessary loan-related requests information, reassign requests and also give authorization for the requests. After these assumptions we will follow the next procedures: The Conceptual and Logical models will be designed for the database that includes all the entities involved and their relations. After which a Physical model will be designed and normalized and then implemented using Ms Server database management system. The next step will be designing necessary APIs (Services) where the internal logical programming will be performed using spring framework for backend scripting. The API will be able to be connected to the database thus allowing the data to be retrieved and stored on the database. After finishing the backend scripting, the user interface will be performed using Angular Framework for the web application. This is a very important part of the process as this is the part where the user interacts with it needs to be as easy to understand as possible.

After finishing with the GUIs and Services, the integration testing will be done for both frontend and backend applications. Then it will be debugged and made error correction again and again till the final product fulfils the requirements. Then the final functional system will be deployed.

1.7 Project Scope

The scope of this project is developing a web-based application for Enat Bank' all branches.

1.8 Development Tools

Table 1: System Development resource used

Hardware requirement

- Computer: RAM =4GB, ROM=500GB

Software requirement

- For backend development: Java spring boot
- To store design and store data on database: MS SQL server 2012
- For frontend development: angular
- Operating System: window 10 or Linux
- Web browsers: chrome, fire-fox, Microsoft edge, brave
- To draw ER diagrams and use cases: Microsoft Vision
- for development: IntelliJ idea
- For containerization of applications: docker
- For continuous integration: Jenkins
- Gitlab: For continuous integration and version control

1.9 Feasibility Analysis

Technical Feasibility: The system is technically feasible since the required resources are available and help for accomplishing it is available. Moreover, we will use the latest technologies, and we are also familiar with the tools and programming languages we will use to develop this project.

Operational Feasibility: The system is operationally feasible since the branches and women's banking solution can use it after it has developed because it solves the problem faced. We will develop the system in such a way that it will have a Help menu, and we will provide training to those actors of the system.

Economic Feasibility: The system is economically feasible since there is no additional cost incurred at the banks.

Legal Feasibility: The system is legally feasible since it will consider legal concerns through data collection.

1.10 Target Beneficiary of the System

Branch Employers

The duties of a branch employee will include:

- Register new customers
- Update customer data
- Generate report
- Approve customer’s request
- Decline customer’s request
- disbursed customer’s request

System Administrator

The duties of a system administrator will include:

- Create new user
- Register new customers
- User role assignment
- Update user
- Disable user

Women banking solution department

The duties of a system administrator will include:

- Create new enat product
- Update enat product
- Disable enat product

CHAPTER TWO

PROPOSED SOLUTION

2.1 Overview

The proposed system will use the latest standard design patterns and system development. And also, it will use a powerful lightweight framework such as spring as back-end scripting tool and Angular as Front-end development tool.

The system's operational work flow would begin with system user enrolment and role assignment, which would be handled by system administrators.

After the user maintenance, the women banking solution department division would login and create primary Maintenances necessary for latter system flow such as enat products, form of business, committees with members definition, if necessary, request type (customers' loan and saving facilities the bank provides) and setting standard days for each request type with respective approval committees.

After the primary system elements have been defined, the branch system users will raise customer requests using the designed form, which includes the requester's recommendation, and the request will be assigned to the appropriate approval committee head based on the outstanding amount and the standard approval committee scales.

The approval committee chairman would then review the pending request and allocate it to one of the committee members.

2.2 System Design

While system designing, we will use the agile model of software development life cycle, which is a combination of an incremental and iterative approach. It is focused on setting flexible requirements. In this software development method, the project is separated into small parts and is delivered in iterations. Using this approach, we have proceeded through the requirement analysis as per the shared excel then specified the functional requirement and designed the high-level system design through figma interface design tool. After which we will analyze and test, then it will be ready for implementation.

In the system implementation phase will also be interactive including the testing process via jasmine and Junit. Finally, the end-to-end testing will be made to test two or more units of an application that works together to perform specific task.

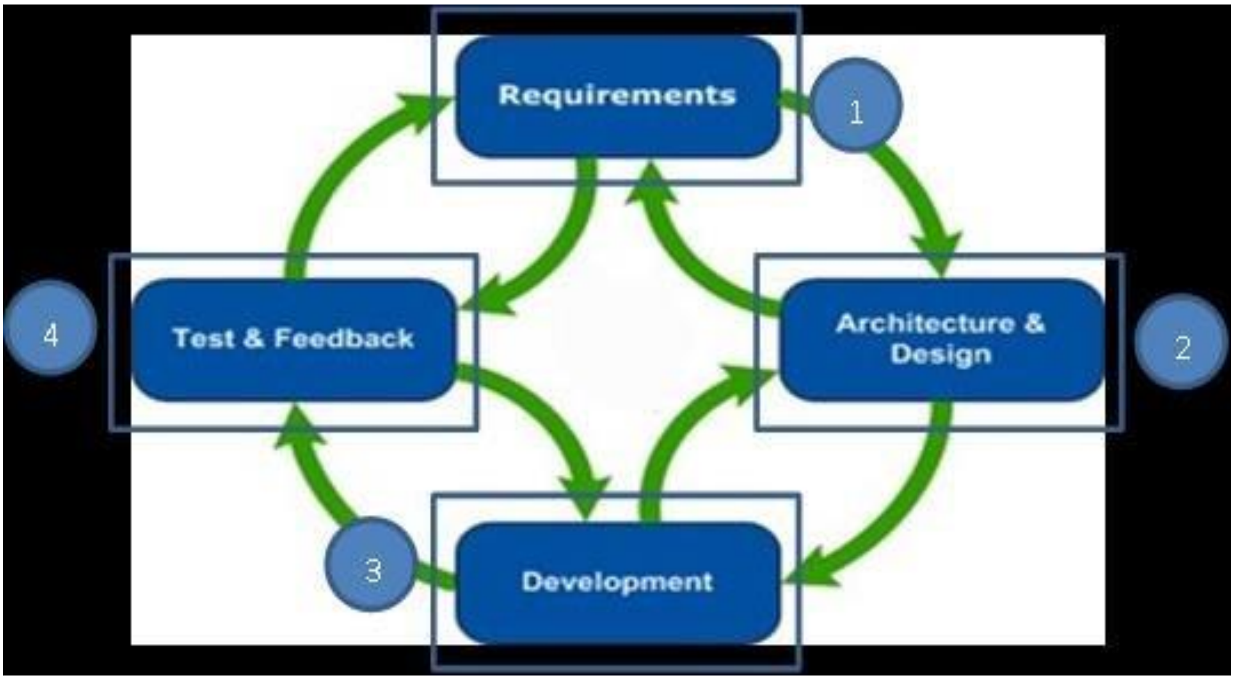


Figure 1: Agile Methodology

2.3 Functional Requirement

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish.

Functional requirements that must be included in the system are:

- Register new users
- Register new customers.
- Register and maintain enat products
- Register and maintain business sectors
- Follow loan and saving statuses of customers
- Generate reports
- Manage different users
- Assign approvers
- Approve request
- Decline request

2.4 Non-Functional Requirement

A non-functional requirement relates to the technical aspects a system must fulfil, such as performance, security and user interface and reliability issues. Some of the non-functional requirements are:

Security issues: -The system must be protected from unauthorized users. To protect requests' data and system misuse, the system should provide restrictions in using system functionality and information access by its user.

Robustness: The system shall validate data entry and prompt the user when the invalid data is entered.

Availability: - The system should easily be available to authorized users.
Performance: - The system should be well fit to perform operations clearly without any problem.
User friendly/ system interface: - The system interface that will be developed must be interactive and easily understandable.

2.5 System Models

Use case Diagram

A use case diagram is a representation of a user's interaction with the system and show the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.

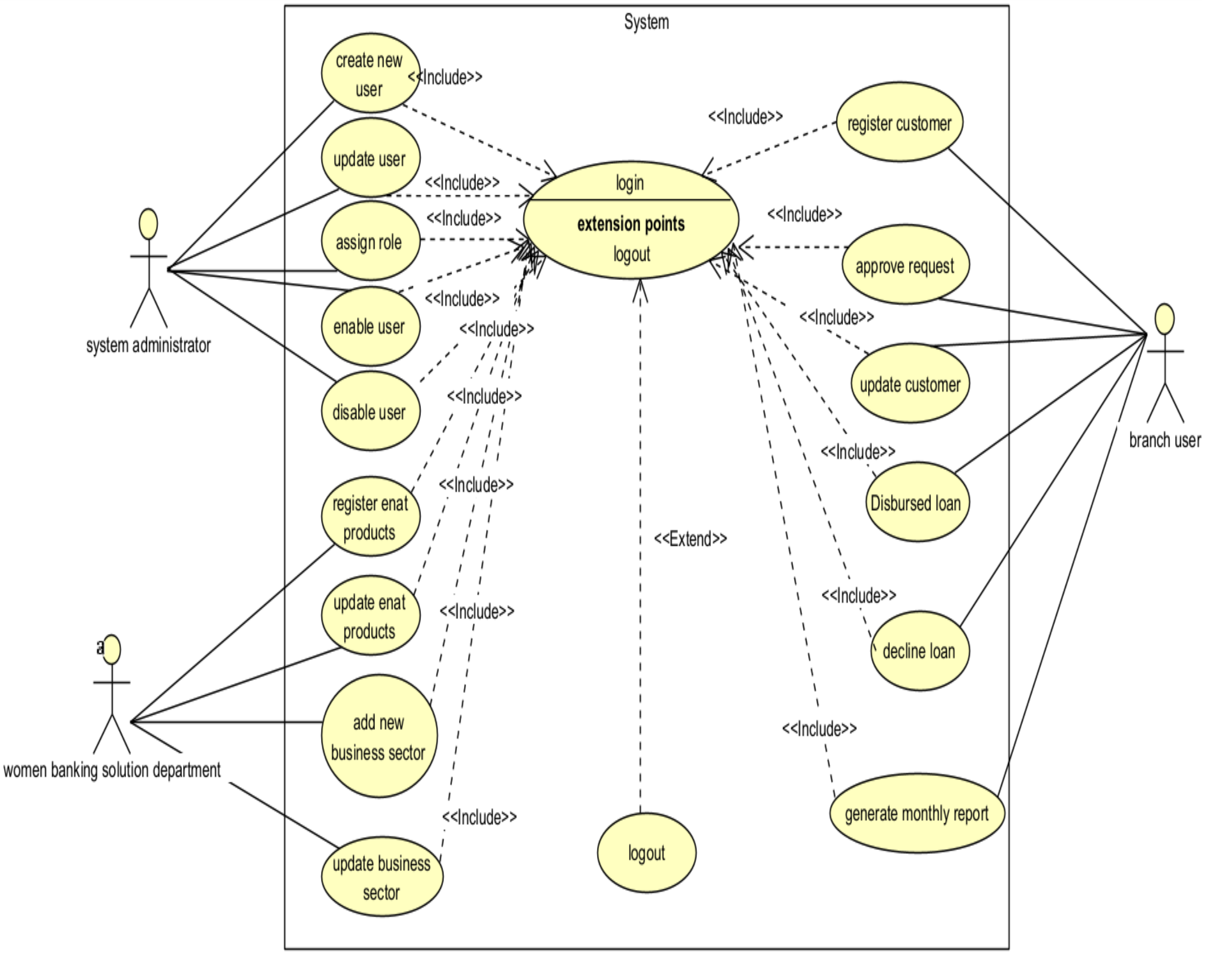


Figure 2: Use case Diagram

Use Case Description

Use case description includes text descriptions of processes with sections like Actors, Precondition, post Condition, flow event and alternative action.

1. Login

Table 1: Use case UC01

Use case name	Login	
ID	UC01	
Actor	system administrator, Branch requester, credit analyst, committee heads, head office credit department division managers	
Description	A mechanism for entering an authorized username and password in order to access the system.	
Pre-condition	The user should have a username and password.	
Post condition	Successful login to the system for the user.	
Basic course action	**User action**	**System response**
	User browses the URL. 3. A user enters legitimate username and password to the login forms field and then click login button.	2. The system displays the Login Form 4. If the credentials are valid, the System displays the appropriate page according to the role of the user. Else *Alt1*
Alternative course of action	*Alt1:* if the username/password is invalid, the system shows an error message and redirects the user to step 2.	

Table 1: login Use Case Description

2. Create User

Table 2: Use case UC02

Use case name	Create User	
ID	UC02	
Actor	System Administrator	
Description	Registering user to the system for later use.	
Pre-condition	System Administrator must be already logged in and should have role.	
Post condition	User saved successfully.	
Basic course action	**User action**	**System response**
	1. system admin clicks on create new user button 3. System admin fills the form and clicks Register button	2. The system displays user registration Form. 4. System stores data to the database. else *Alt1*
Alternative course of action	*Alt1*: If the system fails to store the data, the system displays an error message and redirects the user to step 2.	

Table 2: create new user Use Case Description

3. Update User

Table 3: Use case UC03

Use case name	Update User	
ID	UC03	
Actor	System Administrator	
Description	Update user's data.	
Pre-condition	System Administrator must be already logged in and should have role.	
Post condition	User's data is updated or modified.	
Basic course action	**User action**	**System response**
	1. system admin clicks on edit user button 3. System admin modifies the field that needs to be updated and clicks update button	2. The system displays a form filled with existing user details. 4. System stores the modified user's data to the database. Else *Alt1*
Alternative course of action	*Alt1*: If the system fails to store the modified user's data, the system displays an error message and redirects the user to step 2.	

Table 3: update user Use Case Description

4. Role assignment

Table 4: Use case UC04

Use case name	Role assignment	
ID	UC04	
Actor	System Administrator	
Description	Assign roles to user.	
Pre-condition	System Admin must be already logged in and should have role.	
Post condition	Role is assigned for intended user.	
Basic course action	**User action**	**System response**
	1.System administrator selects user. 3.The system administrator selects the **Role Mappings** tab, select required role and clicks on the transfer icon.	2. The system displays user detail information. 4. The role would be transferred to the user's effective role list. Else *Alt1*
Alternative course of action	*Alt1*: If the system fails to assign role to user, the system displays an error message and redirects the user to step.	

Table 4: role assignment Use Case Description

5. Disable user

Table 5: Use case UC05

Use case name	Disable user	
ID	UC05	
Actor	System Administrator	
Description	Disable users account from accessing loan follow-up system.	
Pre-condition	System Admin must be already logged in and should have role.	
Post condition	User's account is disabled.	
Basic course action	**User action**	**System response**
	1.System Administrator selects the user. 3. The System Administrator switch the enabled button to off.	2.The system displays user detail information. 4.The user would be disabled.

Table 5: disable user Use Case Description

6. Register customer

Table 6: Use case UC06

Use case name	Register customer	
ID	UC06	
Actor	Branche employee	
Description	Register new customer into the system.	
Pre-condition	Branch employee must be already logged in and should have role.	
Post condition	New customer’s data is registered into the system.	
Basic course action	**User action**	**System response**
	1. branch employee clicks on create new customer button 3. a branch employee fills forms and then click register button.	2. The system displays customer registration Form. 4. System stores customer’s data to the database. Else *Alt1*
Alternative course of action	*Alt1*: If the system fails to store the customer’s data, the system displays an error message and redirects the user to step 2.	

Table 6: register new customer Use Case Description

7. Update customer

Table 7: Use case UC07

Use case name	Update customer	
ID	UC07	
Actor	Branch employee	
Description	System allows to edit any customer information.	
Pre-condition	An employee must be already logged in and should have role.	
Post condition	customer’s data is modified and saved into the system.	
Basic course action	**User action**	**System response**
	1. branch employee clicks on update customer button. 3. modifies the field that needs to be modified and then click update button	2.The system displays operator update form with customer’s data. 4.System stores modified data to the database. Else *Alt1*
Alternative course of action	*Alt1*: If the system fails to store the modified data, the system displays an error message and redirects the user to step 2.	

Table 7: update customer Use Case Description

8. Register new enat products

Table 8: Use case UC08

Use case name	Register enat products	
ID	UC08	
Actor	Women banking solution department manager	
Description	Register new enat products into the system.	
Pre-condition	manager must be already logged in and should have role.	
Post condition	New enat products data is registered into the system.	
Basic course action	**User action**	**System response**
	1. manager clicks on create new customer button 3. fills forms and then click register button.	2. The system displays enat products registration Form. 4. System stores the new data to the database. Else *Alt1*
Alternative course of action	*Alt1*: If the system fails to store the data, the system displays an error message and redirects the user to step 2.	

Table 8: register new enat product Use Case Description

9. Update enat products

Table 9: Use case UC09

Use case name	Update enat products	
ID	UC09	
Actor	Women banking solution department manager	
Description	System allows to edit any enat products information.	
Pre-condition	Manager must be already logged in and should have role.	
Post condition	data is modified and saved into the system.	
Basic course action	**User action**	**System response**
	1. manager clicks on update enat products button. 3. modifies the field that needs to be modified and then click update button	2.The system displays enat products update form with enat products data. 4.System stores modified data to the database. Else *Alt1*
Alternative course of action	*Alt1*: If the system fails to store the modified data, the system displays an error message and redirects the user to step 2.	

Table 9: update enat product Use Case Description

10. Register new business sector

Table 10: Use case UC10

Use case name	Register new business sector	
ID	UC10	
Actor	Women banking solution department manager	
Description	Register new business sector into the system.	
Pre-condition	manager must be already logged in and should have role.	
Post condition	New business sector data is registered into the system.	
Basic course action	**User action**	**System response**
	1. manager clicks on create new business sector button 3. fills forms and then click register button.	2. The system displays business sector registration Form. 4. System stores the new data to the database. Else *Alt1*
Alternative course of action	*Alt1*: If the system fails to store the data, the system displays an error message and redirects the user to step 2.	

Table 10: add new business sector Use Case Description

11. Update business sector

Table 11: Use case UC11

Use case name	Update business sector	
ID	UC11	
Actor	Women banking solution department manager	
Description	System allows to edit any business sector information.	
Pre-condition	Manager must be already logged in and should have role.	
Post condition	data is modified and saved into the system.	
Basic course action	**User action**	**System response**
	1. manager clicks on update business sector button. 3. modifies the field that needs to be modified and then click update button	2.The system displays enat product business sector update form with business sector data. 4.System stores modified data to the database. Else *Alt1*
Alternative course of action	*Alt1*: If the system fails to store the modified data, the system displays an error message and redirects the user to step 2.	

Table 11: update business sector Use Case Description

12. Approve Request

Table 12: Use case UC12

<table>
<tr><td>Use case name</td><td colspan="2">Approve Request</td></tr>
<tr><td>ID</td><td colspan="2">UC12</td></tr>
<tr><td>Actor</td><td colspan="2">Branch employees</td></tr>
<tr><td>Description</td><td colspan="2">The request would marked as <i>APPROVED</i> by Branch employees and add his/her recommendation and remark.</td></tr>
<tr><td>Pre-condition</td><td colspan="2">Branch employees must be already logged in and should have role.</td></tr>
<tr><td>Post condition</td><td colspan="2">The request status would be changed to <i>APPROVED</i> and branch requester analyst shall continue the loan and saving processing with the client.</td></tr>
<tr><td rowspan="2">Basic course action</td><td><b>User action</b></td><td><b>System response</b></td></tr>
<tr><td>1. Branch employees selects the request which get approval.
3. Branch employees clicks on <b>Approve</b> button.
4.The user selects the approved status, add recommendation and clicks on <b>Submit</b> button.</td><td>2. The system would display the request detail.
4. The user would be redirected to request approval page.
5. The request’s updates would be saved.Else <i>Alt1</i></td></tr>
<tr><td>Alternative course of action</td><td colspan="2"><i>Alt1</i>: The system would display error message and redirected to request summary page.</td></tr>
</table>

Table 12: approve request Use Case Description

13. decline Request

Table 13: Use case UC13

Use case name	Decline Request	
ID	UC13	
Actor	Branch employees	
Description	The request would marked as *declined* by Branch employees and add his/her recommendation and remark.	
Pre-condition	Branch employees must be already logged in and should have role.	
Post condition	The request status would be changed to *declined* and branch requester analyst shall stop or change the loan and saving processing with the client.	
Basic course action	**User action**	**System response**
	1. Branch employees selects the request which get declined. 3. Branch employees selects **decline**. 4.The user selects the declined status, add recommendation and clicks on **Submit** button.	2. The system would display the request detail. 4. The user would be redirected to request approval page. 5. The request’s updates would be saved. Else *Alt1*
Alternative course of action	*Alt1*: The system would display error message and redirected to request summary page.	

Table 13: decline request Use Case Description

Sequence Diagram

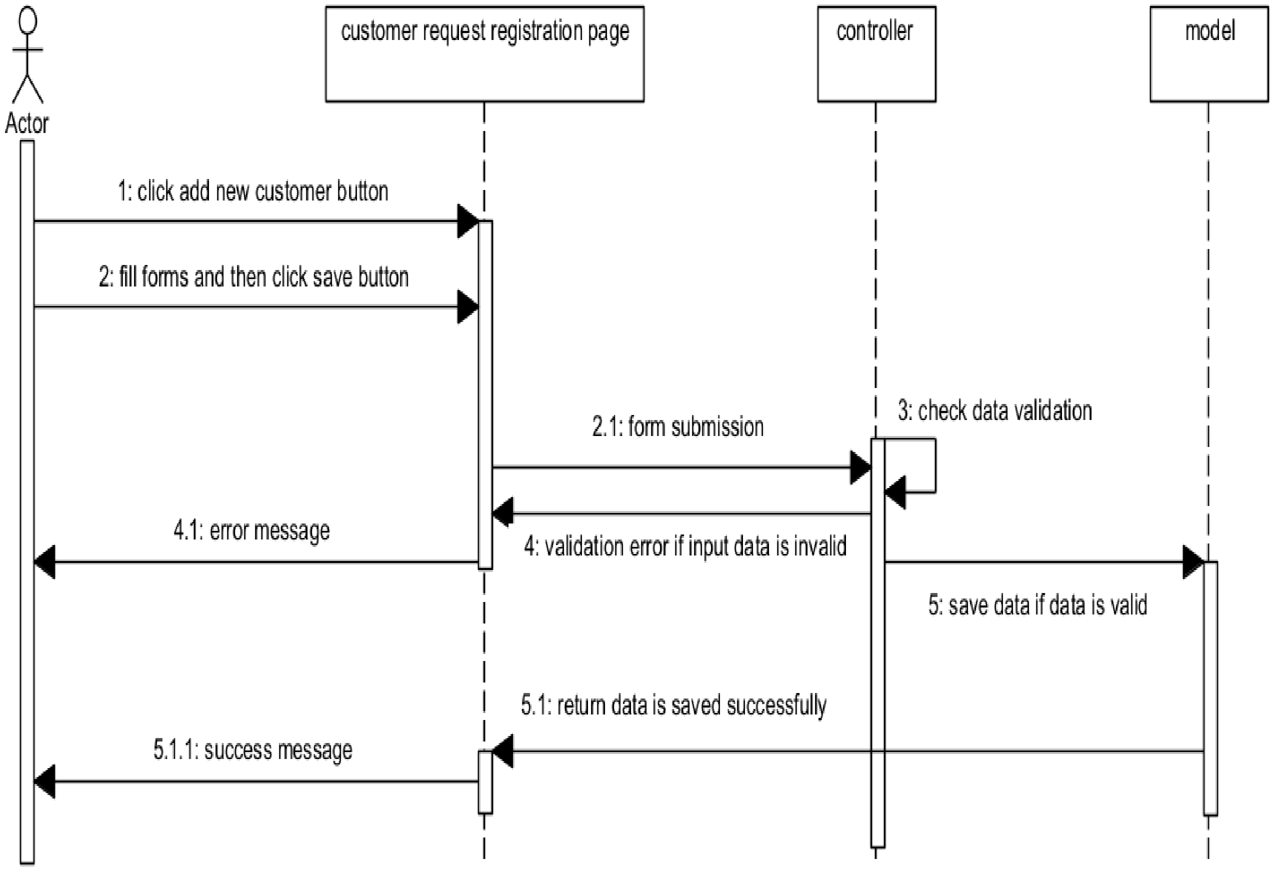

Actor
customer request registration page
controller
model
1: click add new customer button
2: fill forms and then click save button
2.1: form submission
3: check data validation
4: validation error if input data is invalid
4.1: error message
5: save data if data is valid
5.1: return data is saved successfully
5.1.1: success message

Figure 3:add new customer Sequence Diagram

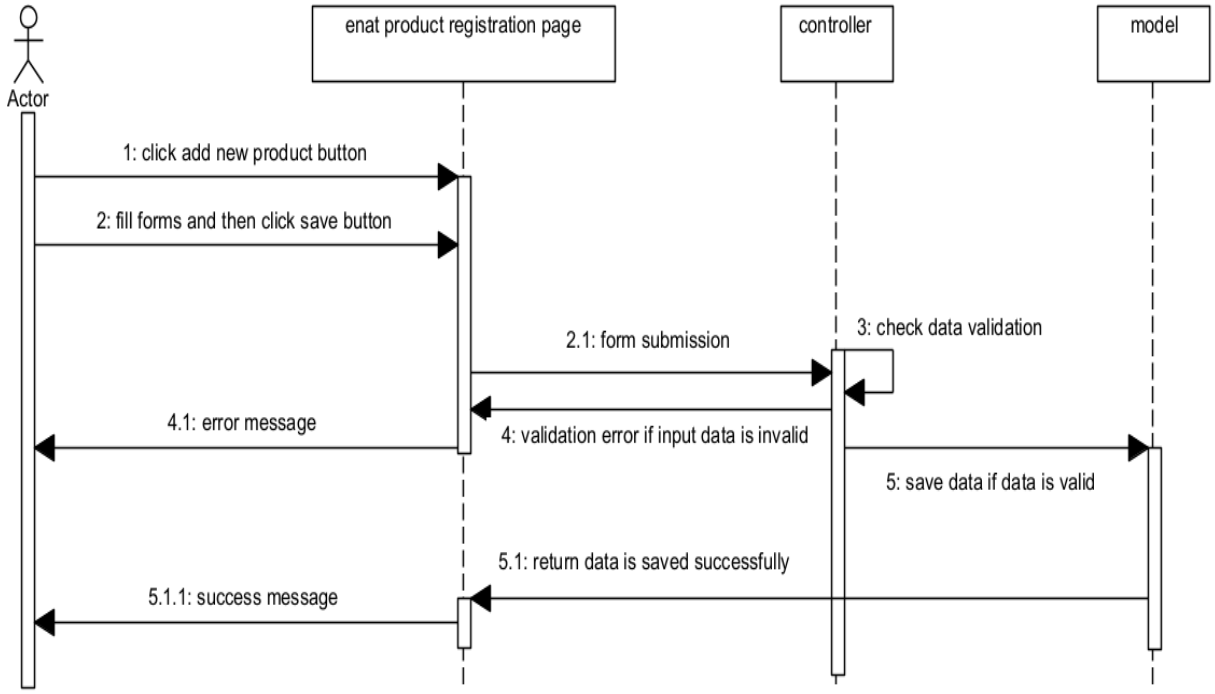

Actor
enat product registration page
controller
model
1: click add new product button
2: fill forms and then click save button
2.1: form submission
3: check data validation
4.1: error message
4: validation error if input data is invalid
5: save data if data is valid
5.1: return data is saved successfully
5.1.1: success message

Figure 4: add new product Sequence Diagram

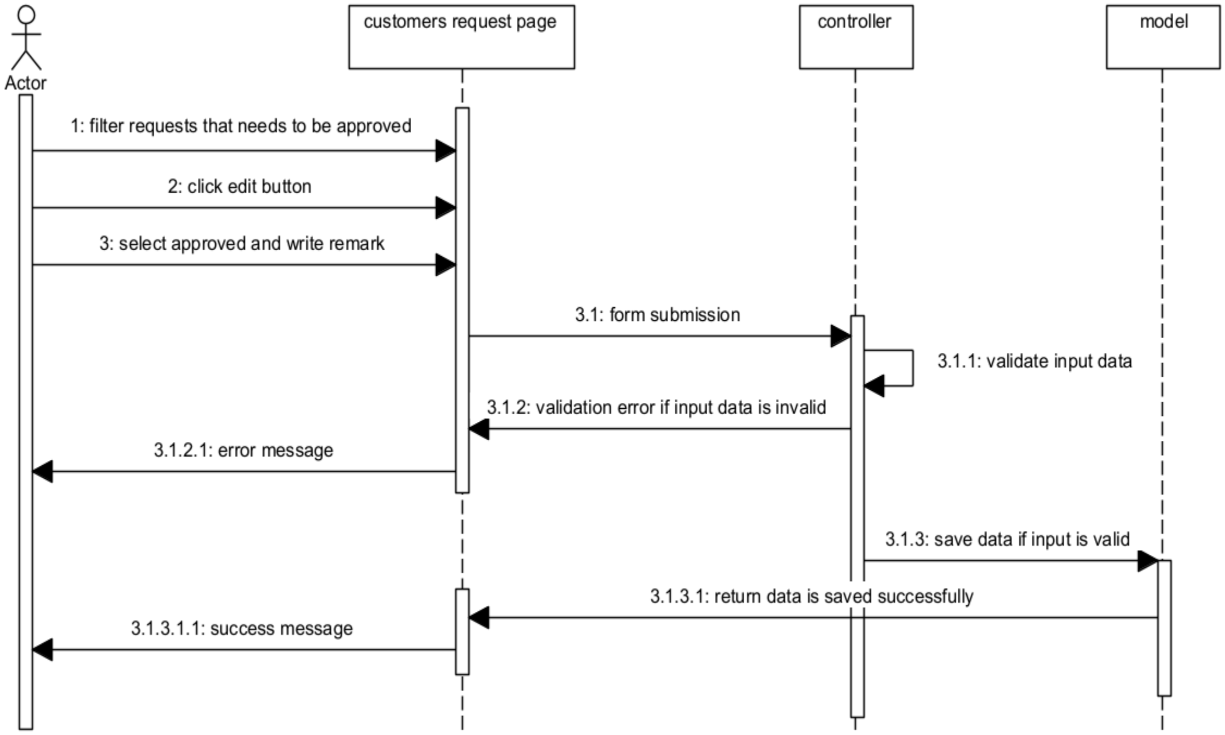
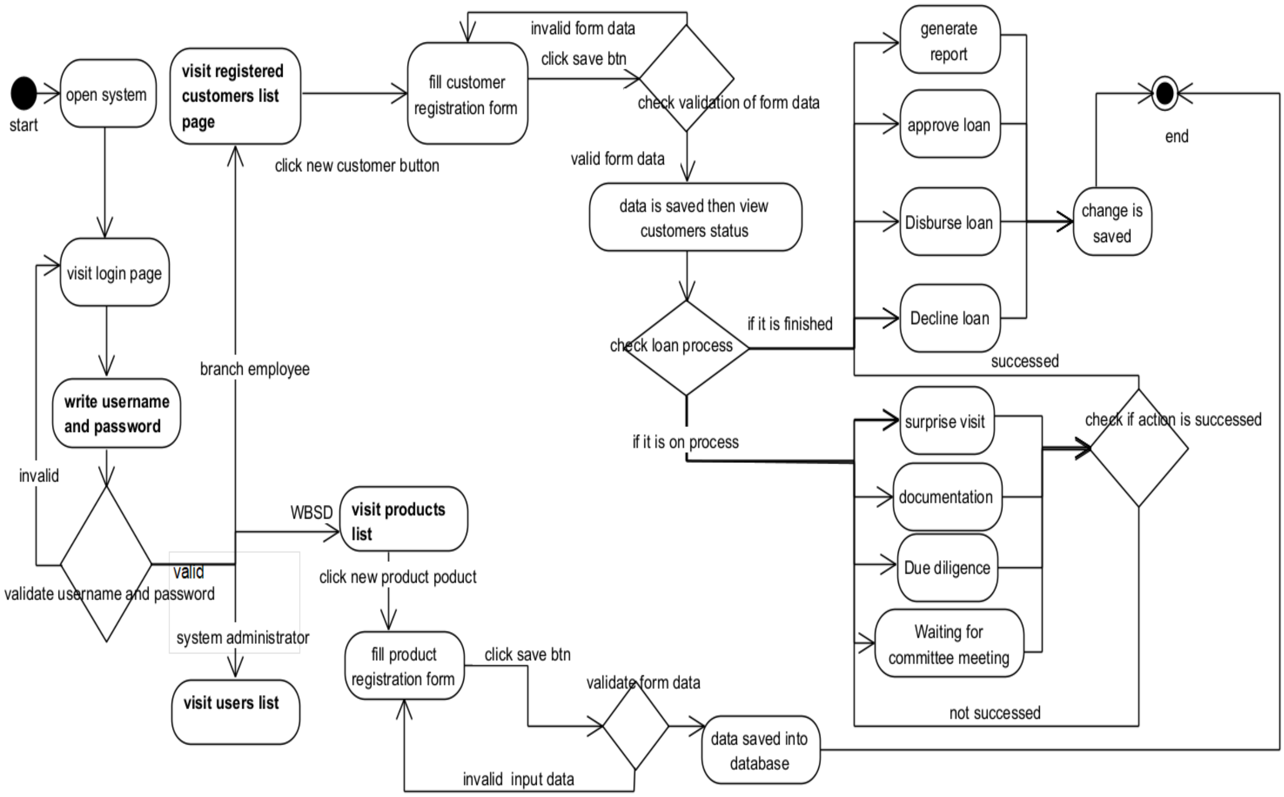

Actor
customers request page
controller
model
1: filter requests that needs to be approved
2: click edit button
3: select approved and write remark
3.1: form submission
3.1.1: validate input data
3.1.2: validation error if input data is invalid
3.1.2.1: error message
3.1.3: save data if input is valid
3.1.3.1: return data is saved successfully
3.1.3.1.1: success message

Figure 5: approve request Sequence Diagram

Activity diagram

We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.



Database Design

The major elements of a system are data flow, data stores, processes, and entities. The Data Dictionary describes all these elements of a system. It is an electronic glossary of items. It defines each element encountered during the analysis and design of a new system.

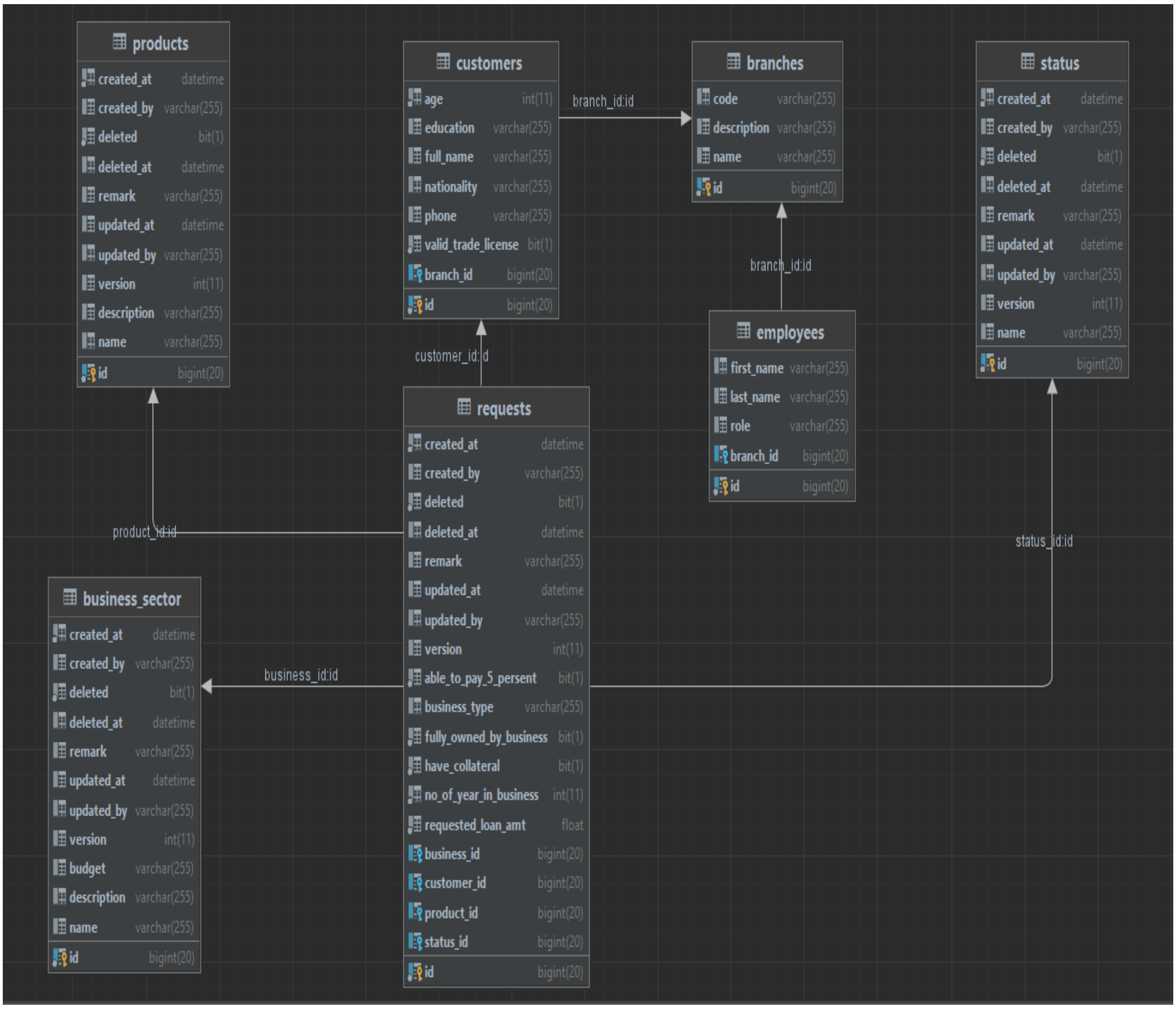


Figure 6: database design

CHAPTER THREE

CONCLUSION

3.1 Conclusion

These days, a lot of manual labor is done to follow up on the loan processing of clients from branches up women banking solution departments and the head office. The automated system will provide a reliable, user-friendly, practical, and secure system that will allow users to accept loan and saving-related requests, assign cases the intended employee, serve customers efficiently, and track cases until they are closed.