



Wolkite University
We Strive for Wisdom!

WOLKITE UNIVERSITY
COLLEGE OF COMPUTING AND INFORMATICS
DEPARTMENT OF INFORMATION TECHNOLOGY
PROJECT TITLE: WEB BASED SUPRMARKET
MANAGEMENT SYSTEM FOR AGAPE SUPERMARKET

Prepared by:

NAME OF THE STUDENTS

ID NO

1. Alehgn Adinew.....CIR/167/10
2. Godie MawunCIR/190/10
3. Zelalem AddisCIR/239/10

Project advisor: Mr. Adisalem Hadush

AUGUST, 2013

WOLKITE UNIVERSITY
COLLEGE OF COMPUTING AND INFORMATICS
DEPARTMENT OF INFORMATION TECHNOLOGY
PROJECT TITLE: WEB BASED SUPERMARKET MANAGEMENT
SYSTEM FOR AGAPE SUPERMARKET
SUBMITTED TO DEPARTMENT OF INFORMATION TECHNOLOGY IN
PARTIAL FULFILMENT OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE IN INFORMATION
TECHNOLOGY
BY

NAME OF THE STUDENTS		IDNO
1.	Alehgni Adinew	CIR/167/10
2.	Godie Mawun	CIR/190/10
3.	Zelalem Addis	CIR/239/10

PROJECT ADVISOR: MR. ADISALEM HADUSH (MSC)

Wolkite University, Wolkite, Ethiopia

AUGEST, 19, 2013 E.C

DECLARATION

This is to declare that this project work which is done under the supervision of Mr.

Adisalem Hadush and having the title **Web based Supermarket Management System for Agape Supermarket** is the sole contribution of:

1. Alehgni Adinew
2. Godie Mawun
3. Zelalem Addis

No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if the violation of this declaration is proven.

Date: 19/12/2013 E.C

Group Members:

Full Name

Signature

Alehgni Adinew

Godie Mawun

Zelalem Addis

APPROVAL SHEET

This is certified that the project entitled on Web Based Agape supermarket management system for collage of computing and informatics developed and submitted by

1. Alehgni Adinew
2. Godie Mawun
3. Zelalem Addis

A project proposal submitted to the Department of Information Technology of Wolkite University in Partial Fulfilment of Requirements for the Degree of Bachelor of science in Information Technology

Advisor Name

Signature

Date

Mr.Adisalem Hadush

APPROVED BY

Examining Board

Signature

Date

1. _____

2. _____

3. _____

Wolkite University, Wolkite, Ethiopia

Acknowledgment

We want to appreciate the almighty of God for the gift of life, protection, Health, academic success and for giving us the strength to preparing the first-semester final project documentation. We want to also in a special way appreciate our Parents for the support and sponsorship of our education from initial to preparing the first-semester final project documentation, for care, patience, and love we pray the Almighty God grants them long life on earth to enjoy the fruit of their labor. We want to in a special way acknowledge our project advisor Adisalem Hadush (MSc.). for his support in the realization of this project work. We appreciate in a special way our department head Mr. Amanual Tamirat who providing different tools that help for doing this project. We appreciate all our friends and well-wishers whose name were not mentioned we say our Lord reward you all what you want.

Table of Contents

DECLARATION	ii
Acknowledgment	iv
List of figures	x
List of table	xii
LIST OF ABBREVIATIONS	xiii
ABSTRACT	xiv
CHAPTER ONE	1
1. INTRODUCTION	1
1.1. Background of the organization	1
1.2. Statement of the Problem	2
1.3. Objective of the Project.....	2
1.3.1. General Objective	2
1.3.2. Specific objectives	3
1.4. Feasibility of the project.....	3
1.4.1. Technical Feasibility	3
1.4.2. Economic Feasibility	3
1.4.3. Operational Feasibility.....	3
1.4.4. Schedule Feasibility	4
1.4.5. Legal Feasibility.....	4
1.5. Scope of the project.....	4
1.6. Limitation of the system.....	5
1.7. Significance the project.....	5
1.8. Beneficiaries of the Project	5
1.9. Methodology of the project	6

1.9.1.	Data Collection method/techniques	6
1.9.2.	System Analysis and Design.....	6
1.9.3.	System Development Model.....	7
1.9.4.	Software and Hardware to be used	7
1.10.	Budget Cost Breakdown.....	7
1.11.	Project plan activity	8
1.12.	Project Schedule	8
1.13.	Team member Origination.....	9
CHAPTER TWO		10
2.	EXISTING SYSTEM	10
2.1.	Introduction to Existing System.....	10
2.2.	Users in the existing system	10
2.3.	Major function of the existing system.....	11
2.4.	Forms and Other Documentation of Existing System.....	11
2.5.	Draw back of the Existing System	11
2.6.	Business rules.....	12
2.7.	Report Generated in the Existing System	13
CHAPTER THREE		14
3.	PROPOSED SYSTEM	14
3.1.	Overview of proposed System	14
3.2.	Functional Requirements.....	14
3.3.	Non Functional Requirements.....	14
3.3.1.	User Interface and Human Factors	15
3.3.2.	Hardware consideration	15
3.3.3.	Security Issue	15

3.3.4.	Performance consideration.....	15
3.3.5.	Error Handling and Validation.....	16
3.3.6.	Quality Issues.....	16
3.3.7.	Backup and Recovery	16
3.3.8.	Physical Environment	16
3.3.9.	Resource Issues	17
3.3.10.	Documentation	17
CHAPTER FOUR.....		18
4.	SYSTEM ANALYSES.....	18
4.1.	Introduction	18
4.2.	System Models	18
4.2.1.	Use case diagram	18
4.2.2.	Use case description.....	21
4.2.3.	Use case scenario	31
4.3.	Object model	32
4.3.1.	Class diagram.....	32
4.3.2.	Data dictionary.....	34
4.4.	Dynamic model	36
4.4.1.	Sequence diagram	36
4.4.2.	Activity diagram modeling	44
4.4.3.	State chart diagram	55
CHAPTER FIVE		63
5.	SYSTEM DESIGN	63
5.1.	Design Goal.....	63
5.1.1.	Performance	63

5.1.2.	Dependability	63
5.1.3.	Maintainability	63
5.1.4.	End user	64
5.2.	Proposed system Architecture	64
5.2.1.	Subsystem Decomposition and Description	65
5.2.2.	Hardware/Software Mapping.....	68
5.2.3.	Detailed Class Diagram	69
5.2.4.	Persistent Data Management.....	71
5.2.5.	Access Control and Security	73
5.3.	Packages	73
5.4.	Algorithm Design.....	75
5.5.	User interface design.....	76
CHAPTER SIX.....		78
6.	IMPLEMENTATION AND TESTING	78
6.1	Implementation of the Database	78
6.2	Implementation of the Class Diagram	82
6.3	Configuration of the Application Server.....	96
6.4	Configuration of Application Security.....	96
6.5	Implementation of User Interface	96
6.6	Testing.....	97
6.6.1	Test case.....	97
6.6.2	Testing Tools and Environment.....	97
6.6.3	Unit Testing	97
6.6.4	System Testing.....	98
6.6.5	Acceptance Testing.....	99

CHAPTER SEVEN	100
7 CONCLUSION AND RECOMMENDATION.....	100
7.1 Conclusion	100
7.2 Recommendation	100
APPENDICES	101
Reference	108

List of figures

Figure 1 Use case diagram.....	20
Figure 3 login sequence diagram	37
Figure 4 Manager create account sequence diagram	38
Figure 5 customer create account sequence diagram.....	39
Figure 6 View item sequence diagram.....	40
Figure 7 Add item sequence diagram	41
Figure 8 Search item sequence diagram	42
Figure 9 Generate report sequence diagram	42
Figure 10 Item order sequence diagram.....	43
Figure 11 Bill generate sequence diagram.....	44
Figure 12 login activity diagram.....	45
Figure 13 Create account activity diagram	46
Figure 14 Add item activity diagram	47
Figure 15 Update item activity diagram	48
Figure 16 activity diagram search item.....	49
Figure 17 view item activity diagram	50
Figure 18 order item activity diagram	51
Figure 19 Bill generate activity diagram	52
Figure 20 generate report activity diagram.....	53
Figure 21 delete account activity diagram	54
Figure 22 login state chart diagram	55
Figure 23 create account state chart diagram.....	56
Figure 24 add item state chart diagram.....	57
Figure 25 report generate state chart diagram.....	58
Figure 26 order item state chart diagram	59
Figure 27 bill generate state chart diagram.....	60
Figure 28 generate report state diagram.....	61
Figure 29 delete account state chart diagram.....	62
Figure 30 Proposed system architecture	65

Figure 31 Subsystem Decomposition	67
Figure 32 Deployment diagram	69
Figure 33 Detailed class diagram.....	70
Figure 34 Persistent Data Management	72
Figure 35 Package diagram.....	74
Figure 36 Home page user interface	76
Figure 37 Login user interface	77

List of table

Table 1 show budget cost breakdown	8
Table 2 show project schedule	8
Table 3 show team member organization	9
Table 4 login use case description.	21
Table 5 manager create account use case description.....	22
Table 6 Customer create account use case description	23
Table 7 View item use case description.....	24
Table 8 Item add use case description.	25
Table 9 Search item use case description	26
Table 10 employee add use case description	26
Table 11 change password use case description.	27
Table 12 Generate report use case description	27
Table 13 Item order use case description.....	28
Table 14 Delete employee use case description	29
Table 15 Bill generate use case description.....	30
Table 16 Update item use case description.....	31
Table 17 Customer data dictionary	34
Table 18 Item data dictionary	34
Table 19 Sales Item info data dictionary	35
Table 20 Order item data dictionary	35
Table 21 Account data dictionary	36
Table 22 Access Control and Security.....	73
Table 23 Tast case.....	97
Table 24 Unit tast.....	98

LIST OF ABBREVIATIONS

- ASMAgape Supermarket
- ASMMS.....Agape supermarket management system for Agape supermarket
- CSS Cascading Style Sheets.
- GUIGraphical User Interface.
- Ms-word:.....Microsoft word
- My-SQL:My Structured Query Language
- OOSAD.....Object Oriented System Analysis and Design
- PHP:.....Hypertext Preprocessor
- SNNPR:..... Southern Nations, Nationalities and Peoples Region
- UML..... Unified Modeling Language
- Xamp Server:Windows, Apache, MYSQL, PHP

ABSTRACT

Web based Agape supermarket management system is a system that will be used to manage the proper aspects of the super market. The project supermarket management system is developed with the objective of making the system more reliable, easier, and informative. There is a lot of reason for introduction to this system. In manual system, there is the number of insufficiencies that a sales person faces. Large record books have to be maintained where relevant and irrelevant information to be stored. But our system reduces paper work. On the other hand, there are inherent problems that exist in manual system. Usually they lack efficiency. It is necessary to know the existing system of a given organization (Agape supermarket) to develop a better system. Existing system currently performs different activities such as product selling, product registration, daily item sales report are carried out using a traditional approach or manual based method. This approach lacks the capability to alleviate problems like susceptibility to errors, incompleteness, performance, time consumption, security, efficiency and ineffective data for analysis in the agape supermarket's day to day business processes. As we try to describe above the existing system faces some problems and the problem's primarily resulted from the manual system of running the activities and we try to put an alternative solution to the problems which are described in the above problem identifying in the existing system section. The best alternative solution to the existing system is to change the existing manual system in to computerized and producing an online system to reduce the time consumption, and human power of the customer.

CHAPTER ONE

1. INTRODUCTION

Online Supermarket management system is a system that used to manage the proper aspects of the super market. These aspects involve managing information about the various products, managers, employee, customers, and billing. This system provides an efficient way of managing the supermarket information. The project we have prepared “online supermarket management system also aimed to solve some problems occurred in the agape supermarket. The problem that occur in agape supermarket are include lack of security of data, doesn’t differentiate the expire date of the product, needs manual calculations. Because of this, the team members try to solve the problem of agape supermarket by gathering requirement analysis, by identifying the problems that exist in supermarket. After Designed existing system more informative, reliable, fast and easy for all the stake-holders. And also the system must perform the easily search the item, determine the expire date of the item by generating notification, calculate the total profit of the product, sale transaction, generate bill, generate report and to provide the detail information about the item (1).

1.1. Background of the organization

Agape supermarket is found in Southern Nations, Nationalities and Peoples Region(SNNPR) state in Gurage Zone Butajera Town, found around 120km away from Addis Ababa and 84km from Wolkite Town, located on the center of the Town of main road near to bright hotel building. It was established in 2008 E.C. When it started two employees and performs some services. But now days it expands its service based on the customers need with additional employees. This super market includes different items like packed foods, fresh dairy product, soft drinks, vegetables, jus and different cosmetics Items other shopping materials with different quality. The main aim of this market is to provide “satisfy customer need” which describe our company 100% providing our customer needs and full fill their demand. Since it is a big super market it has a good income.

1.2. Statement of the Problem

The Supermarket Management System is a project that deals with supermarket automation and it includes both purchasing and selling of items. This project is designed with a goal to making the existing system more informative, reliable, fast and easier. There are many reasons for the starting of the project because in the selling of items through the manual system. Such as it requires handling of large record books that consist of both irrelevant and important information's thus making it difficult to find out the required information as per necessity, it's also hard to tell the items that are out of stock, also items soon going to be expired, performance is also another problem since a customer has to wait in line while another customer is being served, security is also one problem since there is no way of protecting the supermarket information, customers also waste their time and energy. Due this problem the web based supermarket management system for agape supermarket will be implemented to facilitate efficient services for customers by avoiding those previously mentioned problems. Which is rarely functional manipulating things particularly search item, check expired date via manual system. Consequently, a number of problems are being observed in super market management system.

Accordingly, we have got the following problems:

- Expired items will not be identified early and sometimes there might be selling them without knowing that they are out of date.
- Loss of record due to inefficient and traditional file handling system.
- Generate bill and report manually
- Difficult to calculate the amount of item in store.
- Can't easily search the required item
- Needs manual calculations So; Difficult to calculate the amount of item sold per Day.
- Wastage of time
- Lack of security of data.
- It needs a huge human power.
- There is no centralized database system

1.3. Objective of the Project

1.3.1. General Objective

The main objective of this project to design and develop web based Supermarket management system for agape supermarket.

1.3.2. Specific objectives

To achieve the general objective mentioned above the following are specific objective:

- ✓ To analysis and study the existing system
- ✓ To identify or study customer need
- ✓ Identifying the problems under the current system
- ✓ To propose a solution to the identified problem.
- ✓ To design and develop models the proposed system using object oriented approach
- ✓ Implementations of the system
- ✓ Evaluating the system

1.4. Feasibility of the project

Feasibility study is an important phase in the software development process of this project.

The four aspects of feasibility study need to be considered are:

1.4.1. Technical Feasibility

The system will be developed using the object oriented development technique, and the team member have the knowledge of the methodologies and the tools to develop this system.

By using the different techniques, we can perform the work up to the desire of the organization, even though fulfilling human being's need in all direction may be difficulty and impossible. Therefore, the organization has enough capability to be easily applied the proposed system and the technology is able to handle new report and forms required of the new system. For this reason, the project is expected to be technically feasible.

1.4.2. Economic Feasibility

After the system implemented the cost relies to do some activities is reduces. Like to add item, update item info using paper and pen then after the cost of these activities is will be reduced.

- Fastest processing time and reduced processing error.
- Reduce cost for manual data management
- Easy update & retrieval items

1.4.3. Operational Feasibility

It determines how the proposed system will satisfy the organizations need and it also offers Secure, accurate and efficient system to the organization. The system performs all of its tasks in a user-friendly manner. In which, anyone without an advanced technical background can use the

system. The new system will increase and improve the activity of the super market management system by replying the reliable data, increasing throughput, providing adequate response time, and increasing efficiency of work in the super market since the system is operationally feasible.

1.4.4. Schedule Feasibility

The schedule for this project will be feasible due to wealthy information exchange between the developing team, advisor and the organization. And also the time set to develop the system is enough to complete at the predefined day and time. Therefore, this project will be feasible as it will be completed within the time scheduled.

1.4.5. Legal Feasibility

The project team members built the system without violating rules and regulations of the governments as well as the organization. The system being built is for the importance of productivity of the organization, so that the project is legally feasible. Legal feasibility determines whether the proposed system conflicts with the legal requirement or not.

1.5. Scope of the project

The system that is going to be developed is primarily focused on recording item details and the possible transaction during shopping process.

The system covers the following major functionalities:

- The system provides logon facility to the users. The Manager is responsible for managing user accounts.
- The customer order the item via online
- Display notification: The system Display notification about products which are about to reach their expire date.
- Sale transaction: the system will control the sales transaction by calculating the total cost an item to be sold.
- Generate bill: the system will generate bill for each type of sales.
- Generate report: the system generates report for stock Items and sell transaction.
- Calculate the total profit: the system calculates profit.

1.6. Limitation of the system

- Staff management system. The system does not cover the scheduling and payroll system of staffs in the supermarket.
- The system doesn't support barcode.
- Does not support online payment system. The system needs credit card system; in the current context we are unable to do with.

1.7. Significance the project

Generally, the system we are going to develop will give a number of benefits.

- ✓ Reduction of paper and pen.
- ✓ To keep records of the customer and product in the database
- ✓ Easily accessing of product information
- ✓ Reduce work load
- ✓ Faster service time
- ✓ Give better and effective service.
- ✓ Error reduction.
- ✓ Increase security.
- ✓ Increase efficiency.
- ✓ Improve resource management.

1.8. Beneficiaries of the Project

For Team members

- The project has been initiating our team to get knowledge of how to develop the required system.
- ✓ Get financial benefit by selling the system/website.
- ✓ Get moral satisfaction by reducing the workload of employee.

For Supermarket

- Save time
- Reduce over load
- To solve problems those are associated with the manual system.
- To use the document as a reference material for related projects.

- Easily data reporting and accessing.

For customer

- The customers can simply order items of their choice by just visiting the site and creating an account. Because of this the customer can save their time and energy.
- Reduce transport cost.

1.9. Methodology of the project

1.9.1. Data Collection method/techniques

The main purpose of this project will be to shift the super market management system from traditional or manual in to computerized or automated system. To do so, relevant data will be required and collected from different areas. Thus, our team members will use different data collection methods throughout the course of this project i.e. analyzing the existing system and develop a new system that will completely replace the existing system. Those methods include:

Interview: - to get the basic information and background information about the existing super market management system, the team interviewed some owners of supermarket and some employees about the services that are assigned to them, and the problems associated with that environment.

1.9.2. System Analysis and Design

In this project, our team will use object-oriented system development methodology (OOSD) for the design. This technique has several phases some of them are:

- ➡ During this phase, the team models the function of the system (use case modeling), find and identify the business objects, organize the objects and identify the relationship between them and finally model the behavior of the objects in detail.
- ➡ During this phase, our team uses E-draw Max 9.2 software to refine the use case model and those for designing the class diagram, sequence, collaboration, activity diagrams, and state diagram and to model object interactions and behavior that support the use case scenario.

1.9.3. System Development Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

1.9.4. Software and Hardware to be used

We will use the latest version of the following software and hardware.

Software tools: are software installed on a computer for a different purpose from documentation up to the implementation

- ✓ Notepad++ and sublime: To write the php code for the system
- ✓ E-draw Max: For designing UML diagram associated with the project.
- ✓ My-SQL: To store and retrieve user's information into the database
- ✓ MS-Word: To create and edit document
- ✓ Xamp Server: To test the system by running.
- ✓ Language: PHP

Hardware tools

- ✓ Computer: We will use computer to do any activity
- ✓ Printer: to print a document

1.10. Budget Cost Breakdown

In order to conduct this project a number of items and related costs estimated are described below in the table

Number	Items	Unit	Unit price
1	Laptop	2	34000
2	MS word 2010		Free
3	External hard disk	External hard disk	4000
4	Antivirus	2	Free

5	Application SW	2	Free
6	Paper	3 packets	450
Total of price			42050

Table 1 show budget cost breakdown

1.11. Project plan activity

To develop the system, the team will follow project plan time in order to make it easy and usable project.

Communication Plan

The team will interview the owner of the super market and contact some user.

Managing Task

Every task of this project is being done by all member of the project.

The Resource Plan

Computers' laboratory of Wolkite University is being used mainly to develop the proposed system.

Assessing Risk: Different system of advising and checking of how the system is going to prevent from failure is the concern of the team.

1.12. Project Schedule

Before planning the project, it is expected to divide project into sub tasks and identifying time needed in day or week or month for each task to be performed. To show the project schedule of our project, we use the following Gantt chart.

Project contains	March	April	May	June and July
Project proposal				
Existing system				
Proposed system requirement analysis				
System design				
Implementation and testing				

Table 2 show project schedule

1.13. Team member Origination

The following table shows how the team members organized to undertake the development of the new system. Our team has three (3) members for the accomplishment of the project to automate supermarket management system. The table below shows the way of work break down structure of our project and how we manage our time and cost. But, all of our members have a responsibility to participate and do any task. Then task assigned to individual will not performed by him/herself only doing cooperatively.

Team member organization	Assigned role
Alehegn Adinew	Project proposal, requirement, system design
Godie Mawun	Project proposal, existing system, and requirement analysis
Zelalem Addis	Requirement analysis, system design implementation and tasting

Table 3 show team member organization

CHAPTER TWO

2. EXISTING SYSTEM

2.1. Introduction to Existing System

Currently, agape supermarket's day to day business operations such as product selling, product registration, daily item sales report are carried out using a traditional approach or manual based method. This approach lacks the capability to alleviate problems like susceptibility to errors, incompleteness, and ineffective data for analysis in the agape supermarket's day to day business processes. Information regarding stocks, products, sales and purchases are still in pen and paper. Which is not properly organized and managed. As a result, it is difficult in managing item, to check expire date of item, updating item. Since, the users of existing system have one manager, two employees, and many customers. The employee works cooperatively with each other with good approach. On the other hand, the customer visits the supermarket to physically sales and purchase the products from the supermarket. Those users have their own task in supermarket and the major activities of the existing system are registering product or item, generating report, registering employees. Since the existing system use forms during the period of selling process such as receipt of forms and also for employee control input and output items. In this case there are many draw back of the existing system such as performance, security, efficiency, and service.

2.2. Users in the existing system

The users involved in the existing system are:-

- **Manager:** a person who has power in controlling all activities performed within in the supermarket.
- **Employee:** - a person who is responsible give receipt to customer, generates report to manager it is responsible report to the general manager of the supermarket about the detail description of items in a store and item to be bought.
- **Customer:-**a person who buys items in supermarket.

2.3. Major function of the existing system

The major functions of existing system are:-

- **Product Registration:-** The employee register the new product
- **Check expire date:-**employee check expire date of the item and customer check the expire date of the item during buying the item.
- **Generate Report:** The employee report to daily, weekly and monthly to the manager which product is sold, and which is store in cart and which type of product completed or finished.
- **Give receipt:** The employees give receipt manually to the customer when the customers buy the product.
- **Calculate profit:** The Employee view profit
- **Arrangement of item:-**the item or product arranged their shelf number.

2.4. Forms and Other Documentation of Existing System

There are a number of forms and documents which are used by the existing system. The forms are used during the period of selling process such as receipt forms and also for the employee to control input output items. The uses of these forms are to assure the correctness of their activity and to generate reports. In addition to this, these forms are used to increase the relationship with customers in such a way that customers can believe the cost of each materials of the organization if they can get receipt form for each material to which they have purchased from the supermarket.

2.5. Draw back of the Existing System

The existing system used manually under this the following disadvantage

Performance: - It is time wastage and less performance since every task is done manually and it has taking much time and to complete the task requires a long period of time.

- ✓ Regarding on lack of speed the system will generate output with slow speed.

Information: - Redundancy of data.

-Not accurately captured.

Efficiency: It operates over specified long period of time interval. Generally, the above current system mentioned is inefficient.

Security: In terms of the security existing system does not provide any privilege to the employee of the supermarket like protecting the documents. In other words, the documents are not protected from any unauthorized access. In addition, if there is a natural disaster or other thing that can damage the documents, there is no means of getting the documents back. So the security of the supermarket is in danger.

Service: It takes less time to give as well as get the services to the customer.

Time consumption: - the performance of the existing system is very low & it takes long time. When the manager register item, the redundancy will be occur.

2.6. Business rules

Business rule: A business rule defines or constrains one aspect of your business that is intended to assert business structure or influence the behavior of your business.

The business unit also implements its own business rule and regulation which are followed to perform work in easier and best manner. The businesses control the sales i.e. whenever the customer purchases any item. The business unit must provide the receipt for each item that is sold to the customer. After that the original receipt will be given to the customers and the copy of the original receipt is left for the business unit and this receipt helps both the customers and the owner of the business unit for the privacy purposes or else to certify either the customer or the owner would purchase with an appropriate or exact price of the items or not because almost all items has some time guaranties given by the business unit. The customer can only pay in cash not using any other system. The overall description of business rule is:

Name: manager

Identifier: - BR1

Description: The manager must control and manage all activities of the super market and also receive product with full information.

- ✓ The manager view report daily, weekly, monthly.

Name: employee

Identifier: BR2

Description: the employee must do the following activities such as, sell product and preparing bill to the customer in appropriate way.

- ✓ The employees print receipt to the customer.

- ✓ The employee update the item
- ✓ The employee first add new item: when the product is finished

Name: customer

Identifier: - BR3

Description: the customer order product and then buy item based on their own interest.

2.7. Report Generated in the Existing System

In order to control and manage over all activities of the supermarket report must be generated. These reports are generated day to day but are generated in a very routine and boring manner. These reports indicate performance and profitability of the supermarket and are used to change how the organization works to improve its profitability. These reports can be generated by the employee for submission to the manager of the supermarket. The following data are used to generate the reports.

- The daily purchased items grouped into categories
- The number of newly added items or products.
- The number of sales and store item to be inform

CHAPTER THREE

3. PROPOSED SYSTEM

3.1. Overview of proposed System

As previously mentioned in statement of the problem, there are a lot of problems associated with the existing system of the supermarket.

The main aim of our proposed system is to implement web based supermarket management system .This system must allow add new item, update item info , generates report, bill generate, search item and over comes the aforementioned problems (2).

3.2. Functional Requirements

Since we are going to develop a web based supermarket management system, the system will be used to manage and process data according to the rules and regulations of the supermarket. It will also provide report generation facilities to the manager of supermarket that can use it for decision making purpose. The web based system will have the following functionalities.

- ✚ The manger must manage account, update profile, view report, view notification, manage employee, and manage category
- ✚ The employee must keep the items in track of stock level(in category)
- ✚ The employee must send receipt to the customer, manage item, bill generate, check expire date of item, view notification
- ✚ The proposed system must Calculate profit, print receipts, notify expired date of items automatically.
- ✚ The employee must review product sales (sales item info), sales and expenses analysis (report), and receive request (when the customer is order an item).
- ✚ The proposed system must display detail information about goods.
- ✚ The employee and customer will search item from the database.
- ✚ The customer will order items online, create account.

3.3. Non Functional Requirements

The following lists of non-functional requirements are expected from the system:

3.3.1. User Interface and Human Factors

The window format and the form prepared for the information are easy for users they can easily understand the system, shall be designed according to standard and the system shall replace the existing system. The user who don't know how to use the system they can easily access the function of the proposed system because the system will provide user documentation for its user.

3.3.2. Hardware consideration

The organization should have computers having typical storage capacity and processing speed. The software product to be developed should run on existing standard computers. The system will be portable that can run on any type of computer and it supports any type of browsers.

3.3.3. Security Issue

The system should have a security privilege that secures the system .and also, there must be a physical security hat secures especially the server computer. That means the server computer is only allowed for the server admin. Security is implemented by requiring users to login before using the system. The login procedure is password protected. The system has to be well protected unauthorized access by encrypting password.

- ✓ All major transaction of the system must be logged to database and seen only by the administrator. If the system has login part the external intrusions or malicious users cannot get access.
- ✓ Our proposed system has session control: when the user registered once the system can save the user data.

3.3.4. Performance consideration

The end user computer should have medium processor and the server computer should have large processor.it is measured by its speed of processor. The proposed system performs its operations within a minimum amount of time and the user gets the expected result within a few seconds and the system is effective.

- **Response Time** upon request for inquiry the system under normal condition should display results as quickly as possible.

- **Processing Time** Since the system is developing with efficient programming language and database upon request for user's activities the system under normal condition should process the request as quickly as possible.

3.3.5. Error Handling and Validation

Our system handles error by showing the message “invalid input “when the user enters invalid input. Generally, if an error occurs, the system will identify the error and notify the user so that he/she can take the appropriate corrections rather than terminating the system and the system must handle the error.

3.3.6. Quality Issues

The system, it will be tested, verified and, validated and if the error occurs during the execution time it will give an exception. And, because the user will be involved in testing the quality of the system, it will satisfy the user's needs.

- ✚ **Availability:** the system should have to be function at any given time. Which means our proposed system gives service 24 hours per day with maximum response time. The server should be always on to be available.
- **Reliability:** the proposed system will minimize crash during its runtime, since more than one user could use the system simultaneously. the system should be reliable and matured enough in giving its service

3.3.7. Backup and Recovery

The system should have backup using external hard disk. The backup is taken weekly and the system will store what filled on the form and it will help the user to get the information they filled when the connection comes back and it also helps the users those who forgets their password through sending a security number on their phone number which they used during registration.

3.3.8. Physical Environment

The system must be compatible with any environment. But for more feature we recommend that the system to deployed on the cloud that is free from any disaster.

3.3.9. Resource Issues

The system needs the resource like cable, computer, connection, browser etc.

3.3.10. Documentation

Our system is well documented, which include user documentation and system documentation and if we describes each by ones. User documentation was very important because the might need some fundamental training.

CHAPTER FOUR

4. SYSTEM ANALYSES

4.1. Introduction

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish its purpose. Analysis model contains three models: functional, object and dynamic models. The functional model can be described by use case diagrams. Class diagrams describe the object model. Dynamic model can also be described in terms of sequence, state chart and activity diagrams. For the purpose of this project, we have described the analysis model in terms of the functional model and dynamic models using use case, sequence diagrams, Activity diagram, State diagram, and class diagram (3).

4.2. System Models

System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system.

4.2.1. Use case diagram

Use case diagrams are used for capturing functional requirements of the system. It is the functionality of the system or the service provided by the system. In use case diagram it better to consider the following elements.

The system we have proposed has four actors. These actors have their own activities.

System

- Generate notification
- Calculate profit
- Print receipt

Manager

- Manage account
- Manage category
- Manage employee
- View report
- View notification

Employee

- manage item
- generate bill
- View item
- View category
- Generate report(sales and expenses analysis)
- Search item
- Sales item info
- View notification
- Change password
- Send receipts
- Receive request
- Change password
- Sales item info

Customer

- Order item
- Create account
- View item
- Receive receipt
- View category
- Check expire date of item
- Change password

4.2.2. Use case description

Use case id	UC01	
Use case name	Login	
Actors	manager, employee, customer	
Description	All actors must be login to system in order to accomplish their tasks	
Pre-condition	User must have valid username and password	
Flow of event	<u>Actor action</u> <u>step1.</u> Users open the system. <u>step3.</u> User click login link. <u>Step5.</u> User fill username and password then click login button.	<u>System response</u> <u>step2.</u> System displays login link. <u>Step4.</u> System display login form. <u>Step6.</u> The system verifies account form database. <u>Step7.</u> System display user page. <u>Step8.</u> Use case end.
Alternative of action	If user entered an invalid username and password The system displays invalid user name and password warning message. The user back to step 5 of basic course of action.	
Post-condition	The login success and go to user page.	

Table 4 login use case description.

Use case id	UC02
Use case name	Create Account
Actors	Manager
Description	Manager create accounts
Pre-condition	Manager must know required data for creating account

Flow of event	<u>Actor action</u> <u>step1.</u> Manager click on login link. <u>Step3.</u> Manager fills username and password and click login button. <u>Step5.</u> The manager click on account link, then click on create account link. <u>Step7.</u> The manager fill data and then click on create button.	<u>System response</u> <u>step2.</u> System display login page for manager. <u>Step4.</u> System verifies account from database and login to manager page. <u>Step6.</u> The system displays account page. <u>Step8.</u> The system checks validity of data and create account. <u>Step9.</u> The use case end.
Alternative of action	If the manager entered an invalid username or password. The system displays invalid username or password warning message. The manager back to step 3 of basic course of action. If manager enter invalid data for account. The system display invalid data warning message and manager goes back to step7.	
Post-condition	The account successfully created.	

Table 5 manager create account use case description

Use case id	UC03
Use case name	Create Account
Actors	Customer
Description	customer create accounts
Pre-condition	customer must know required data for creating account

Flow of event	<u>Actor action</u> <u>step1.</u> Customer click on login link. <u>Step3.</u> Customer fills username and password and click login button. <u>Step5.</u> The customer click on account link, then click on create account link. <u>Step7.</u> The customer fill data and then click on create button.	<u>System response</u> <u>step2.</u> System display login page for customer. <u>Step4.</u> System verifies account from database and login to customer page. <u>Step6.</u> The system displays account page. <u>Step8.</u> The system check validity of data and create account. <u>Step9.</u> The use case end.
Alternative of action	If the customer entered an invalid username or password. The system displays invalid username or password warning message. The customer back to step 3 of basic course of action. If customer enter invalid data for account. The system display invalid data warning message and customer goes back to step7.	
Post-condition	The account successfully created.	

Table 6 Customer create account use case description

Use case id	UC04
Use case name	View item
Actors	Employee, Customer,
Description	All actors can view all items information
Pre-condition	User must have valid username and password

Flow of event	<u>Actor action</u> <u>step1.</u> The user click view category button <u>step3</u> User select category and click view item button.	<u>System response</u> <u>Step2.</u> System display list of category <u>Step4.</u> The system display selected item background information <u>Step5:</u> Use case end
Alternative of action	No	
Post-condition	Successfully completion the system shows list of item Failure condition If there is no internet connection the system does not display list of item	

Table 7 View item use case description

Use case id	UC05	
Use case name	Add item	
Actors	Employee	
Description	Employee register/add a new item	
Pre-condition	Employee must login to the system.	
Flow of event	<u>Actor action</u> <u>step1.</u> The Employee click category button. <u>Step3 :</u> the Employee select categories and press selected category button <u>Step5 :</u> then the Employee click add item button <u>Step7.</u> Employee Fill item information and clicks add button <u>Step10 :</u> then the Employee click ok	<u>System response</u> <u>step2.</u> System display list of categories <u>step4:</u> the system display selected category <u>Step6.</u> The system display item list form <u>Step8:</u> the system display are you want to add this item <u>Step9:</u> the system display the item successfully added to it. <u>Step11.</u> Use case end.

	to confirm	
Alternative of action	7 If Employee entered invalid data for adding item 7.1. The system displays invalid input warning message 7.2. the use case continue to step7.	
Post-condition	The item added successfully.	

Table 8 Item add use case description.

Use case id	UC06	
Use case name	Search item	
Actors	Employee, customer	
Description	looking for a specific item	
Pre-condition	User must login to the system.	
Flow of event	<u>Actor action</u> step1. User clicks on the search icon on the home page. Step3: user enters the text in search form.	<u>System response</u> step2. The System will generate a search form Step4. The system validate and displays the information of item that can search Step5. Use case end.
Alternative of action	3.1 If User entered invalid data for search item 1. The system displays invalid input warning message 2. The use case goes back step 3	
Post-condition	display item information	

Table 9 Search item use case description

Use case id	UC07	
Use case name	Add employee	
Actors	Manager	
Description	Manager must add employee	
Pre-condition	Manager must login to the system.	
Flow of event	<u>Actor action</u> step1. Manager click employee register button. step3. Manager Fill the data for employee and click register button	<u>System response</u> step2. System display user employee register form for manager. Step4. The system register users employee successfully Step5. Use case end.
Alternative of action	If the manager entered an invalid data. The system displays invalid input warning message The manager back to step 3 of basic course of action.	
Post-condition	The employees register successfully.	

Table 10 employee add use case description

Use case id	UC08	
Use case name	Change password	
Actors	Manager, Employee, Customer	
Description	All actors can change their own account password of the system in order to accomplish their tasks	
Pre-condition	User must have valid username and password	
Flow of event	<u>Actor action</u>	<u>System response</u>

	<u>step1.</u> User click on change password link. <u>Step3.</u> User password and new password, then click on change button.	<u>step2.</u> Systems ask the user to enter Old password and new password. <u>Step4.</u> The system change password successfully. <u>Step5.</u> Use case end
Alternative of action	If user entered an invalid data The system displays invalid data warning message.	
Post-condition	The password changed successfully.	

Table 11 change password use case description.

Use case id	UC09	
Use case name	Generate report	
Actors	Employee	
Description	employee can generate report	
Pre-condition	Employee must have valid username and password	
Flow of event	<u>Actor action</u> <u>step1.</u> The employee click generate report button. <u>step3.</u> The employee fill the form and press button <u>step5.</u> The employee press ok	<u>System response</u> <u>step2.</u> System display generate report form <u>Step4.</u> Are you want to generate this report <u>Step6.</u> The system display success message. <u>Step7.</u> The use case end
Alternative of action	3.1 If the employee entered an invalid input 3.2 The system displays warning message. 3.3 The user back to step 3 and try again	
Post-condition	The report is generate is successfully.	

Table 12 Generate report use case description

Use case id	UC10	
Use case name	Order item	
Actors	Customer	
Description	The category must contain an available items	
Pre-condition	Customer must have valid username and password	
Flow of event	<u>Actor action</u> Step1. the clicks category button Step3: the customer selects items Step5. The customer clicks order button	<u>System response</u> Step2. System display category list Step4. The system display selected item and their detail description like price, expire date. Step6. Use case end
Alternative of action		
Post-condition	The item order successfully.	

Table 13 Item order use case description

Use case id	UC011	
Primary actor	Manager	
Use case name	Delete employee	
Description	This activity is performed when the manager want to delete an existing user.	
Pre-condition	The manager is logged on to the system.	
Basic course of action	Actor action	System response

	<u>step1.</u> The manager select delete user employee account home page. <u>step3.</u> The manager enters the user employee account to be	<u>step2.</u> The system displays delete employee account home page. <u>step4.</u> The system displays the user employee and details.
	deleted <u>step5.</u> Manager deletes the account <u>step7.</u> The manager confirms the deletion.	<u>step6.</u> system asks for confirmation <u>step8.</u> The system deletes the employee account. <u>step9.</u> the system displays employee and is deleted successfully message
Alternative course of action	If the manager inserts invalid employee name The system displays invalid data input message. If the manager inserts valid input The system deletes the employee account	
Post condition	Employee account will be deleted.	

Table 14 Delete employee use case description

Use case id	UC12	
Use case name	Generate bill	
Actors	Employee	
Description	employee can generate bill	
Pre-condition	The employee must login to the system	
Flow of event	<u>Actor action</u> <u>step1.</u> Employee select generate bill payment option from menu bar <u>step3.</u> Employees fill bill payment form.	<u>System response</u> <u>Step2.</u> System display bill payment form <u>Step4.</u> System calculates the total price. <u>step5.</u> The system generate bill

		<u>Step6.</u> The use case end
Alternative of action	The system displays invalid data input message The Employee back to step 3 and try again	
Post-condition	The bill is generate is successfully.	

Table 15 Bill generate use case description

Use case id	UC13	
Use case name	Update item or product	
Actors	Employee	
Description	Employee must update product when the item finished or can goes to expire.	
Pre-condition	Employee must know required data for update products and must login to the system.	
Flow of event	<u>Actor action</u> <u>step1.</u> The Employee click category button <u>step3:</u> the Employee select item from category list <u>step5.</u> Then the Employee click update item button. <u>Step7.</u> The Employee fills the updated information form and click update.	<u>System response</u> <u>step2.</u> The system displays the categories list <u>Step4.</u> The system displays the selected item. <u>Step6.</u> The system display updates list form. <u>Step8:</u> the system display successfully updated message <u>Step9.</u> Use case end.
Alternative of action	7.1 If there is any invalid entry 1. The system displays error message. 2.the use case continue to step5	
Post-condition	The item updated successfully.	

Table 16 Update item use case description

4.2.3. Use case scenario

A scenario is a sequence of steps that represents a single use case execution. It is made up of a number of simple, discrete steps that are designated as being performed by either the “System” or a “User”.

The following are scenario of the proposed agape supermarket management system.

Name of scenarios: add item

Actor name: kebede

Flow of events:

1. He opens the system.
2. Insert username and password then click login button.
3. Clicks on the item add button.
4. Fill data on item add form then click on add button.

Name of scenarios: view item

Actor name: Eyob

Flow of events:

1. He starts the system.
2. Insert username and password then click login button.
3. Click on view item button.
4. View item background information.

Name of scenarios: order item

Actor name: Daniel

Flow of events:

1. He opens the system
2. Insert username and password click login button.
3. Clicks on the order button.
4. Fill data on the form and click order button.

Name of scenarios: bill generate

Actor name: Estifanos

Flow of events:

1. He opens the system
2. Insert username and password click login button.
3. Clicks on the item form.
4. Fill data on the form and click bill button.

Name of scenarios: create account

Actor name: fiseha

Flow of events:

1. He opens the system
2. Insert username and password then click on login button.
3. Click on create account button.
4. Fill data for creating account and click on create button.

4.3. Object model

An object model is a logical interface, software or system that is modeled through the use of object-oriented techniques. It enables the creation of an architectural software or system model prior to development or programming. In our system we use object model like class diagram and data dictionary.

4.3.1. Class diagram

A Class Diagram in Software engineering is a static structure that gives an overview of a software system by displaying classes, attributes, operations, and their relationships between each other. This Diagram includes the class name, attributes, and operation in separate designated compartments. Class Diagram helps construct the code for the software application development.

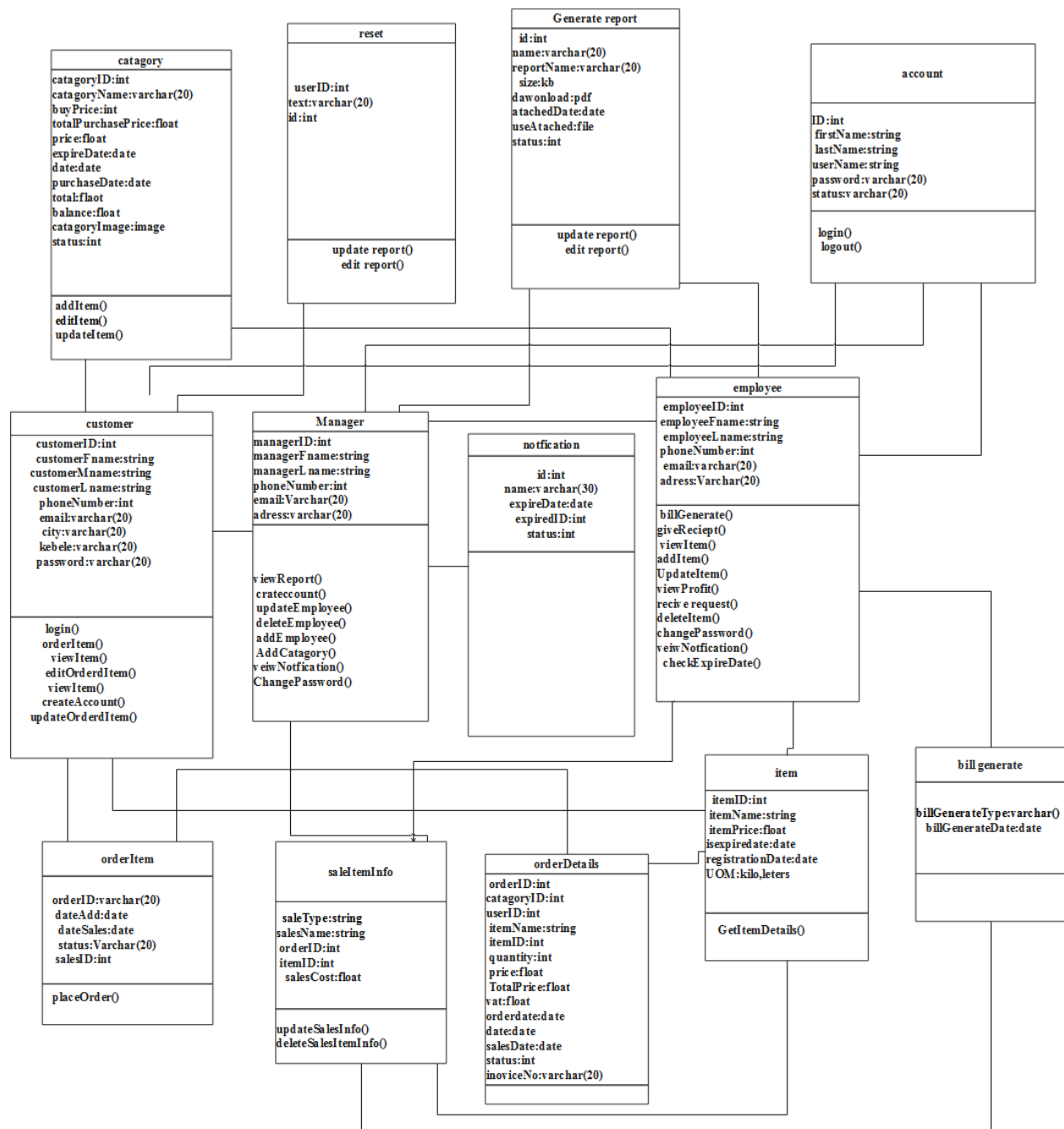


Figure 2 class diagram

4.3.2. Data dictionary

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects, and other data. In our ASMMS ecommerce system the following table describe data dictionary.

Attribute	Datatype	Default	Relation
CustomerId	Int	Not null	Primary
FirstName	String	Not Null	
LastName	String	Not Null	
PhoneNumber	Int	Not Null	
Email	Varchar(20)	Not Null	
customerHousNumber	Varchar(20)	Not Null	
Address	varchar (10)	Not Null	

Table 17 Customer data dictionary

Attribute	Datatype	Default	Relation
ItemId	int	Not null	Primary
ItemName	string	Not Null	
ItemCatagory	string	Not null	
ItemPrice	float	Not Null	
Quantity	Int	Not Null	

Table 18 Item data dictionary

Attribute	Datatype	Default	Relation
ItemId	int	Not null	Primary
OrderedID	Int	Not Null	Foreign key
SalesType	string	Not Null	
SalesCost	float	Not Null	

Table 19 Sales Item info data dictionary

Attribute	Datatype	Default	Relation
OrderID	Int	Not null	Primary
DateAdd	date	Not Null	
DateSales	date	Not Null	
Status	Varchar(20)	Not Null	
salesID	int	Not Null	Foreign key()

Table 20 Order item data dictionary

Attribute	Data type	Default	Relation
Id	Int	Not Null	Primary key
FirstName	String	Not Null	
LastName	String	Not Null	
Username	String	Not Null	
Password	Varchar(20)	Not Null	
userType	Varchar(20)	Not Null	
Status	Boolean	Not Null	

4.4. Dynamic model

A dynamic model represents the behavior of an object over time. It is used where the object's behavior is best described as a set of states that occur in a defined sequence. In our system we use sequence diagram, activity diagram, and State diagram.

4.4.1. Sequence diagram

Diagrams model the dynamic aspects of a software system. The emphasis is on the “sequence” of messages rather than relationship between objects. A sequence diagram maps the flow of logic or flow of control within a usage scenario into a visual diagram enabling the software architect to both document and validate the logic during the analysis and design stages. Sequence diagrams provide more detail and show the message exchanged among a set of objects over time. Sequence diagrams are good for showing the behavior sequences seen by users of a diagram shows only the sequence of messages not their exact timing. Sequence diagrams can show concurrent signals

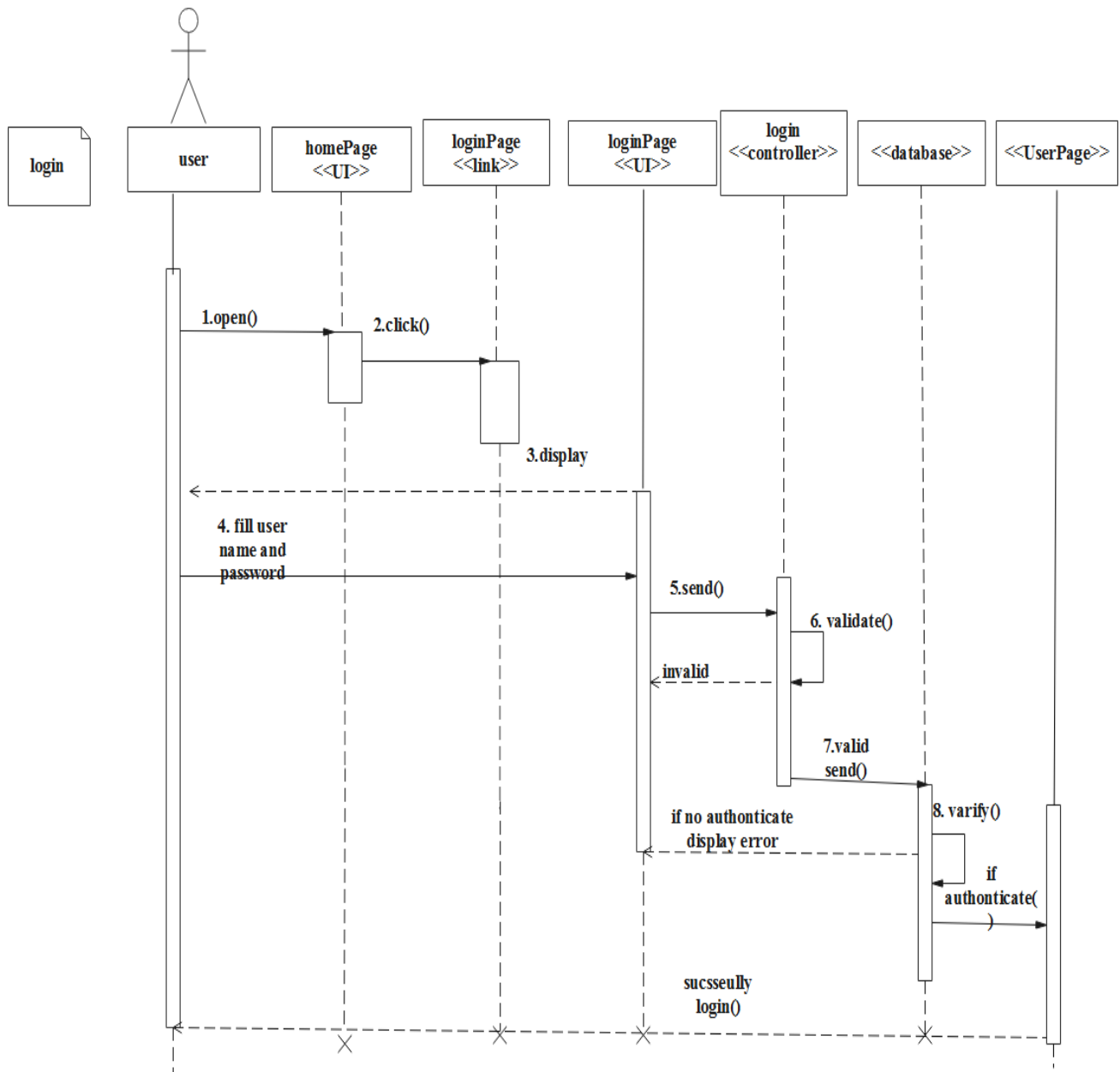


Figure 2 login sequence diagram

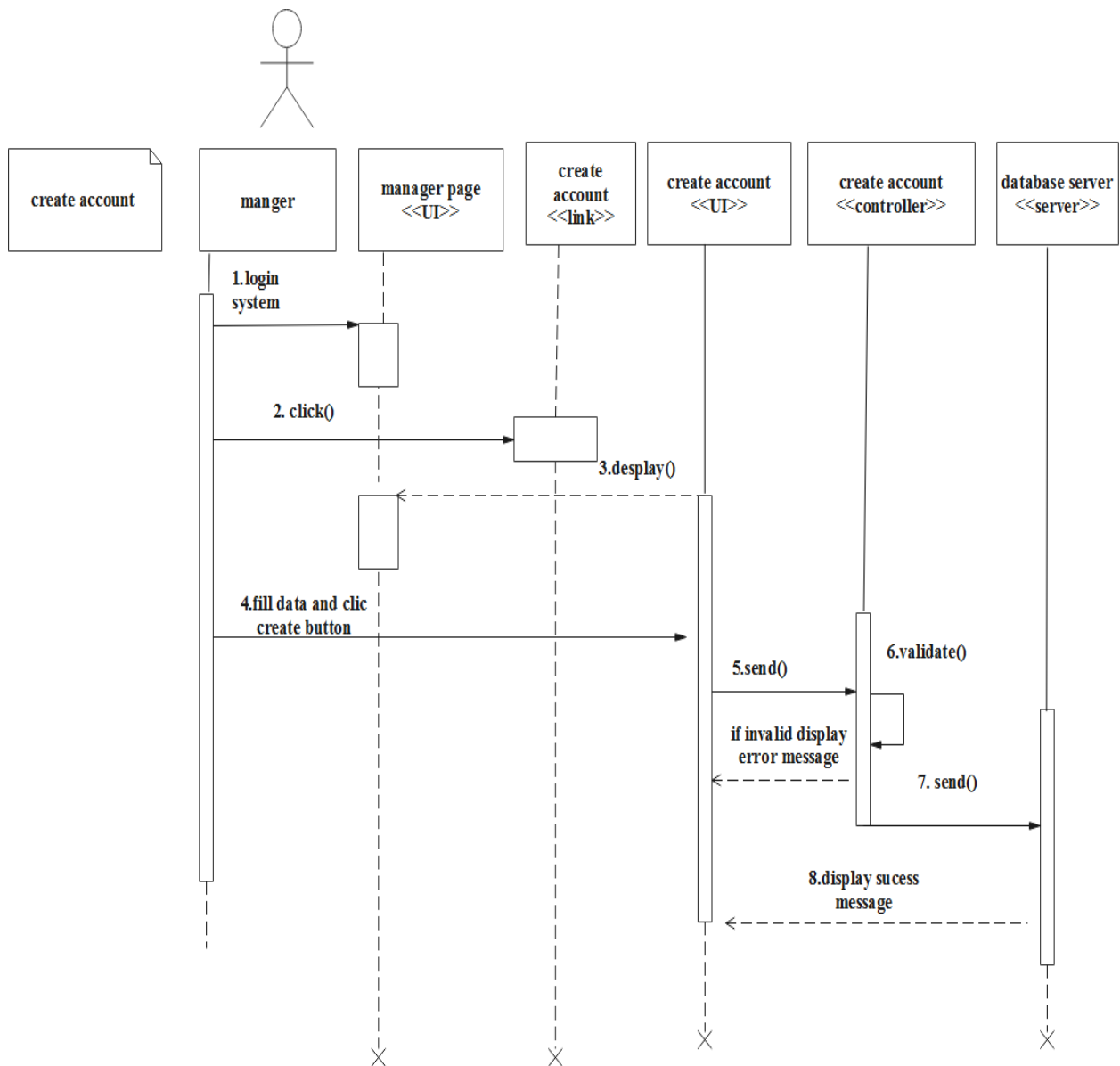


Figure 3 Manager create account sequence diagram

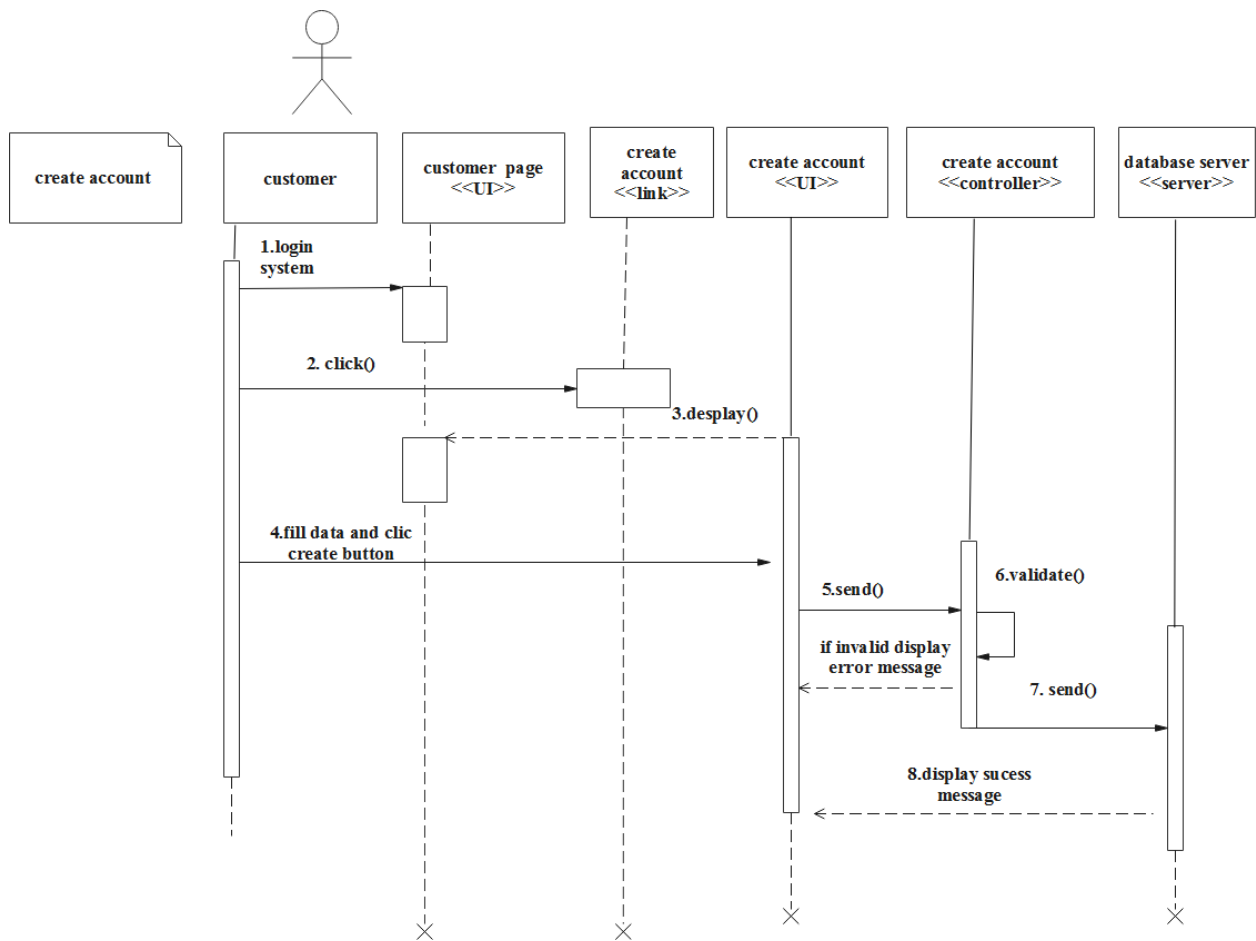


Figure 4 customer create account sequence diagram

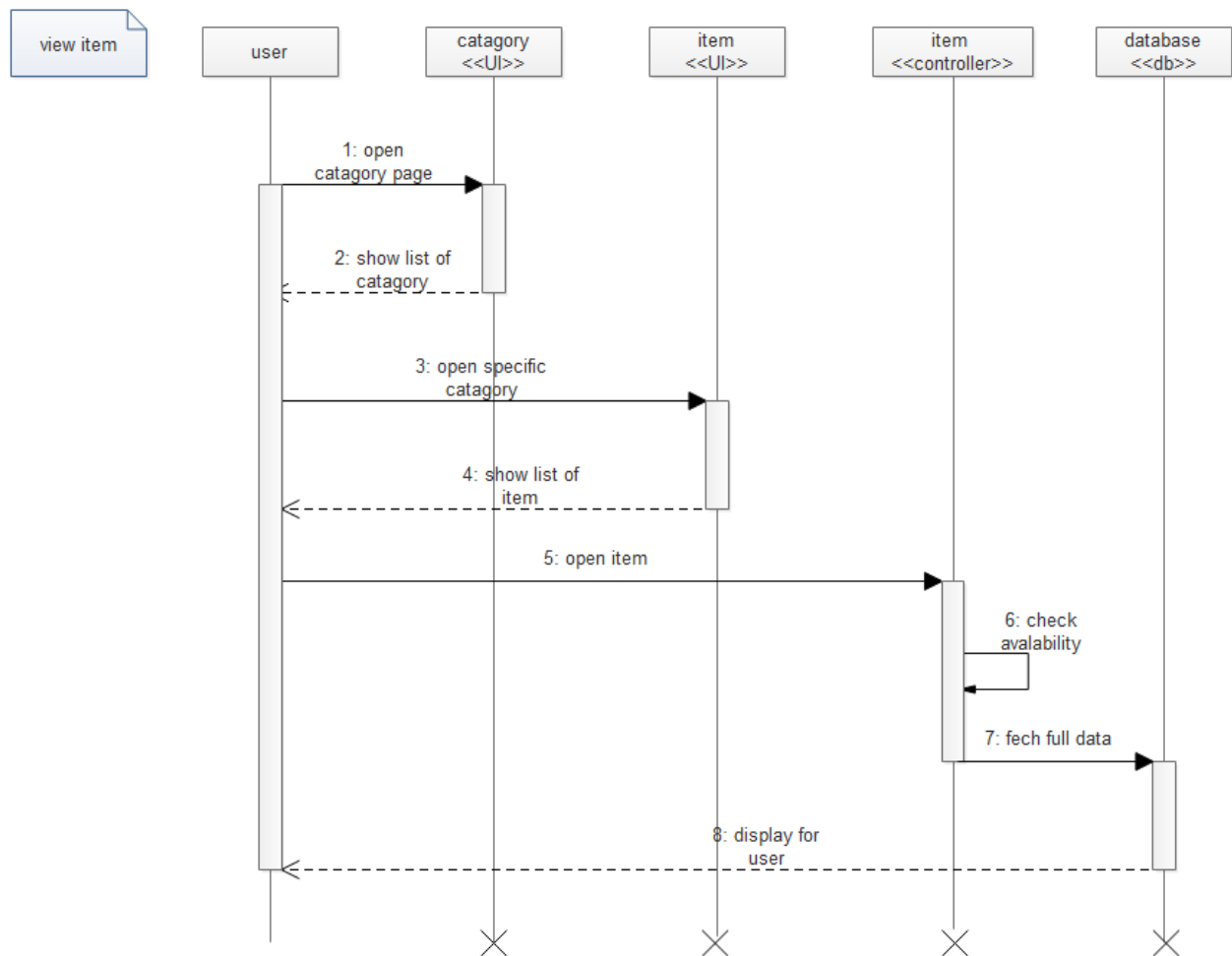


Figure 5 View item sequence diagram

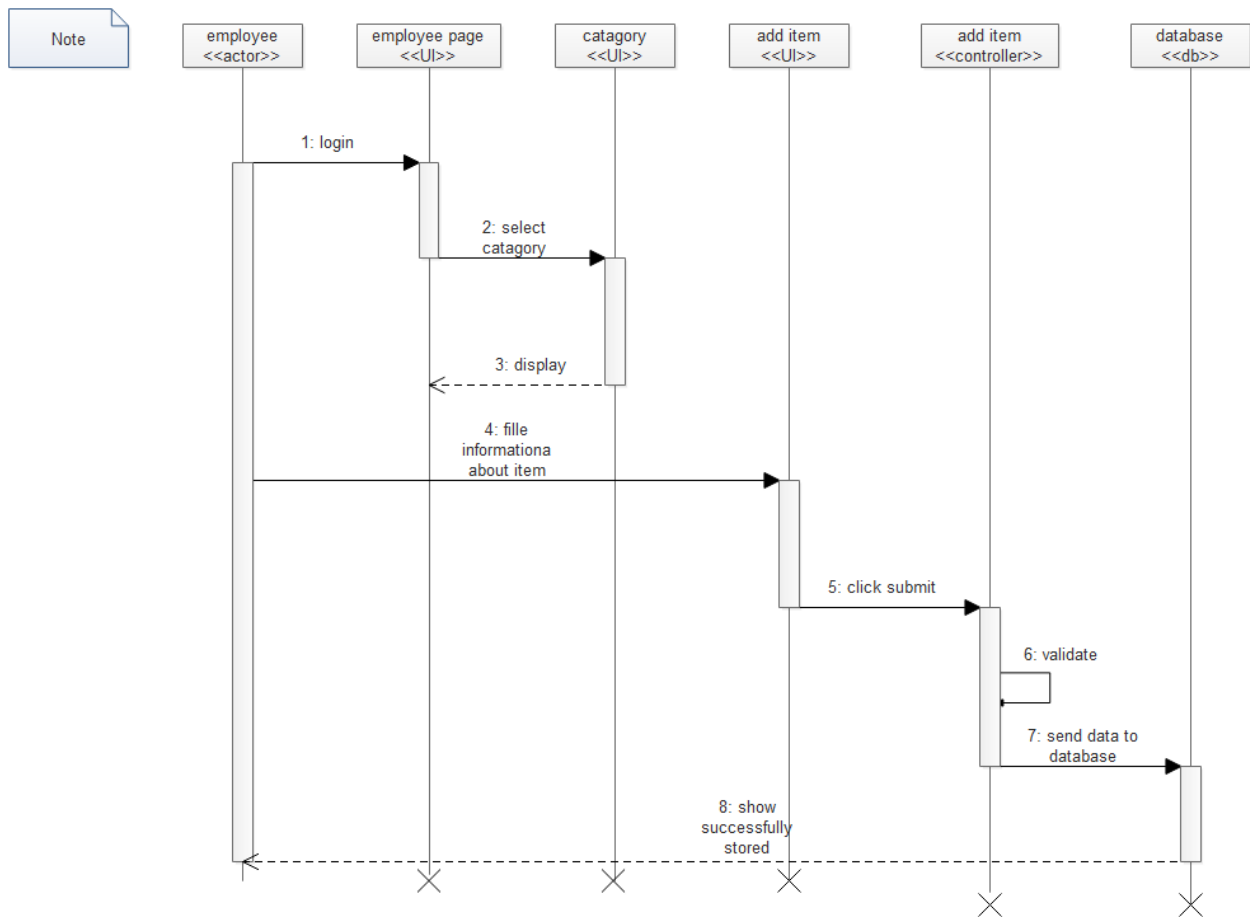


Figure 6 Add item sequence diagram

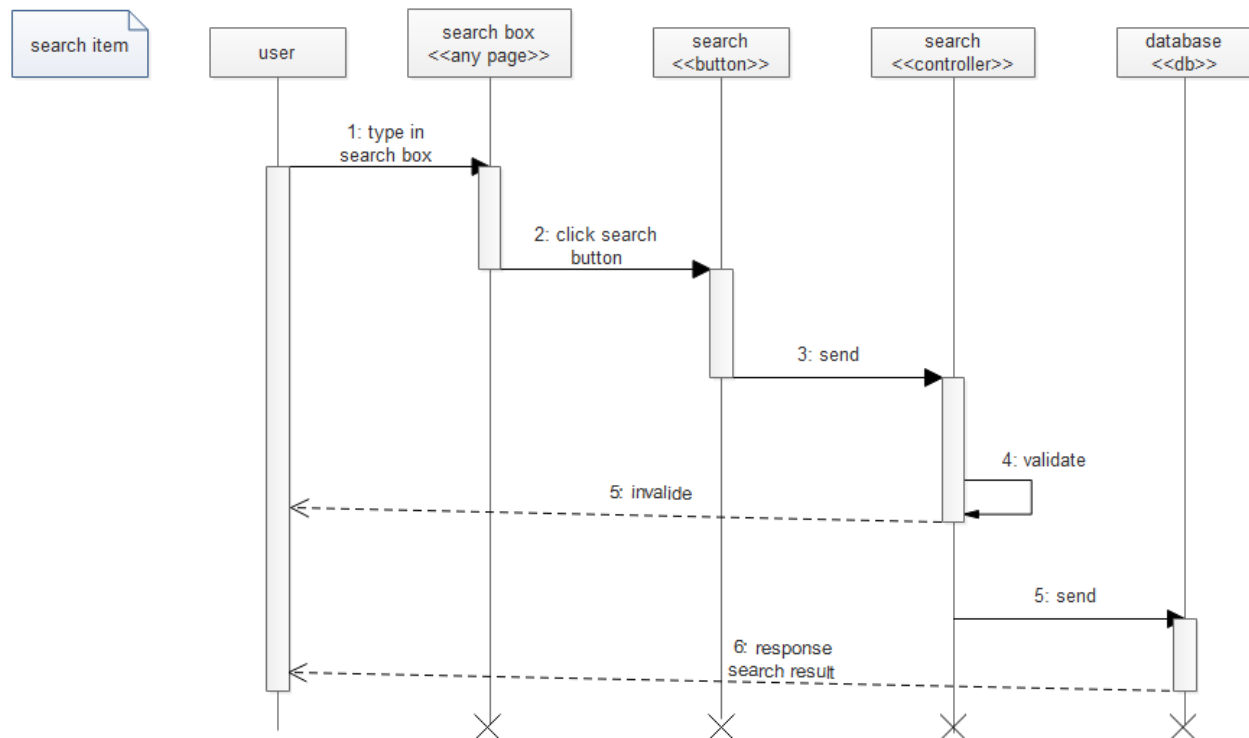


Figure 7 Search item sequence diagram

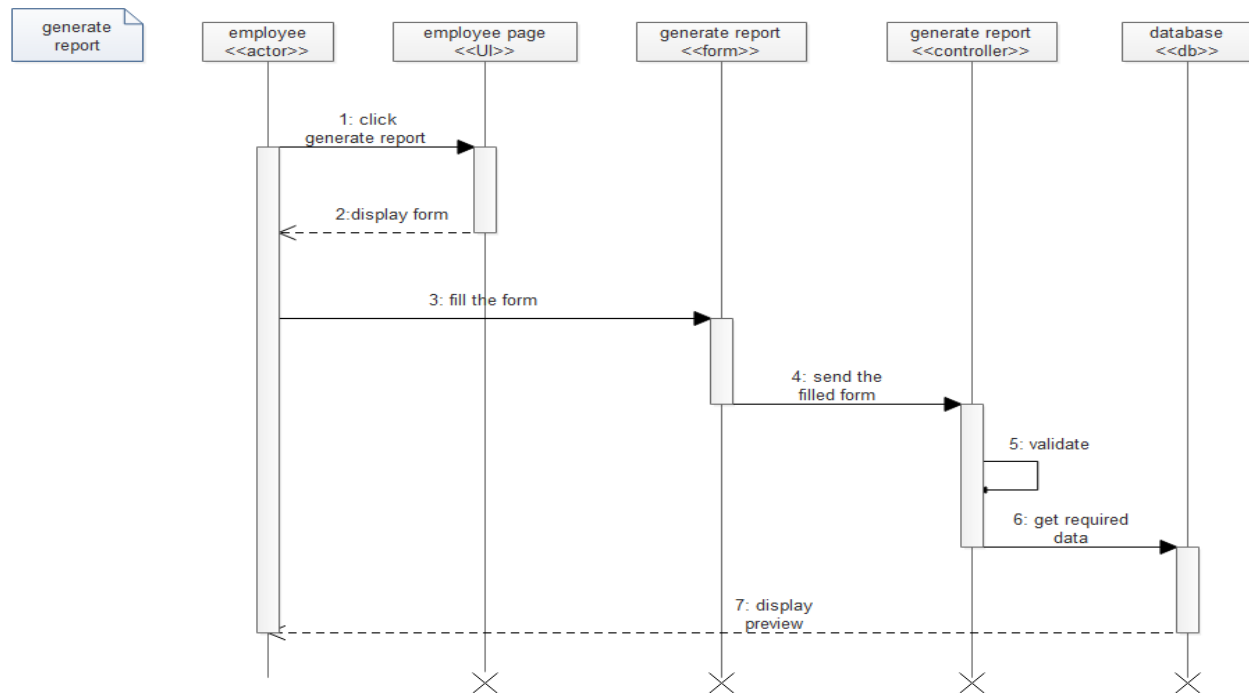


Figure 8 Generate report sequence diagram

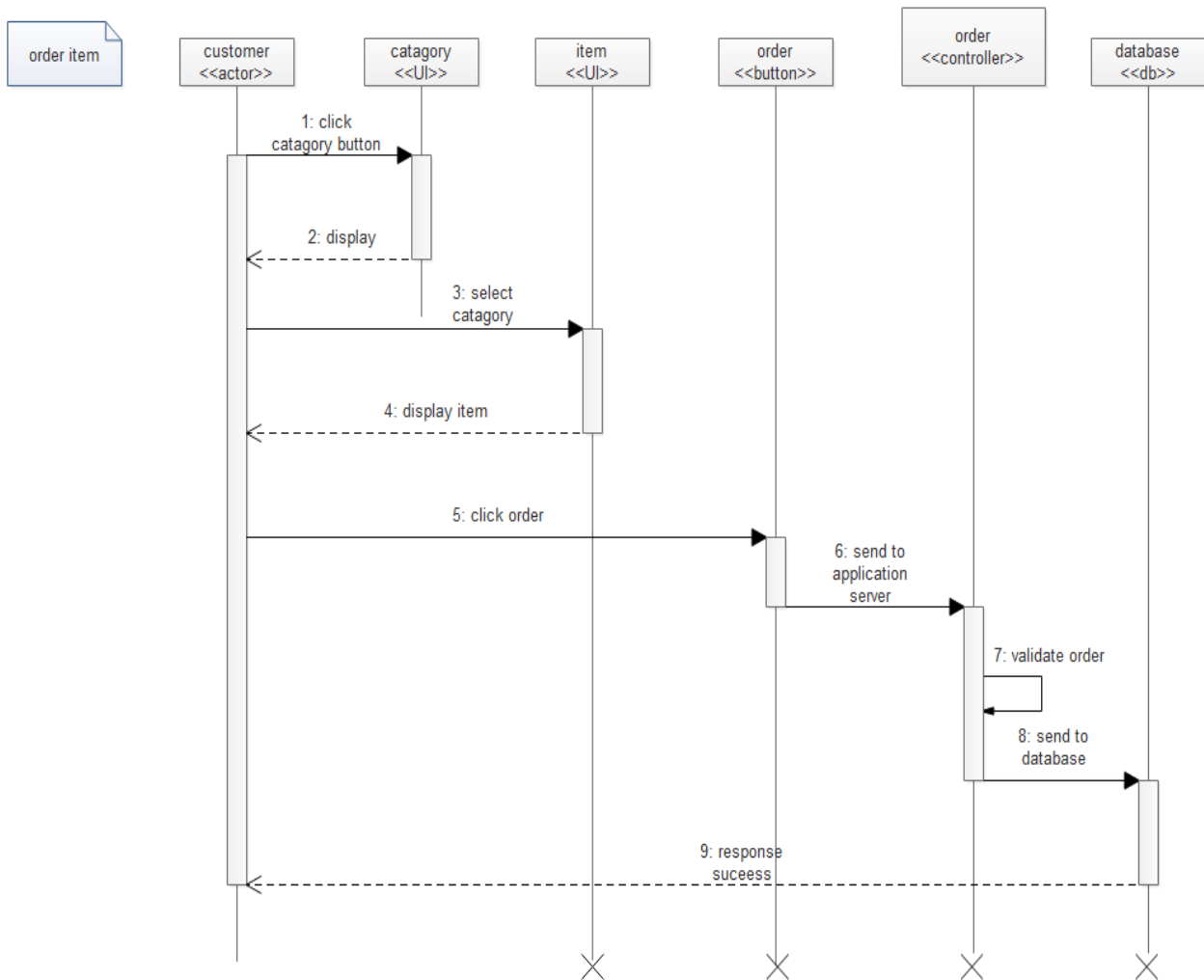


Figure 9 Item order sequence diagram

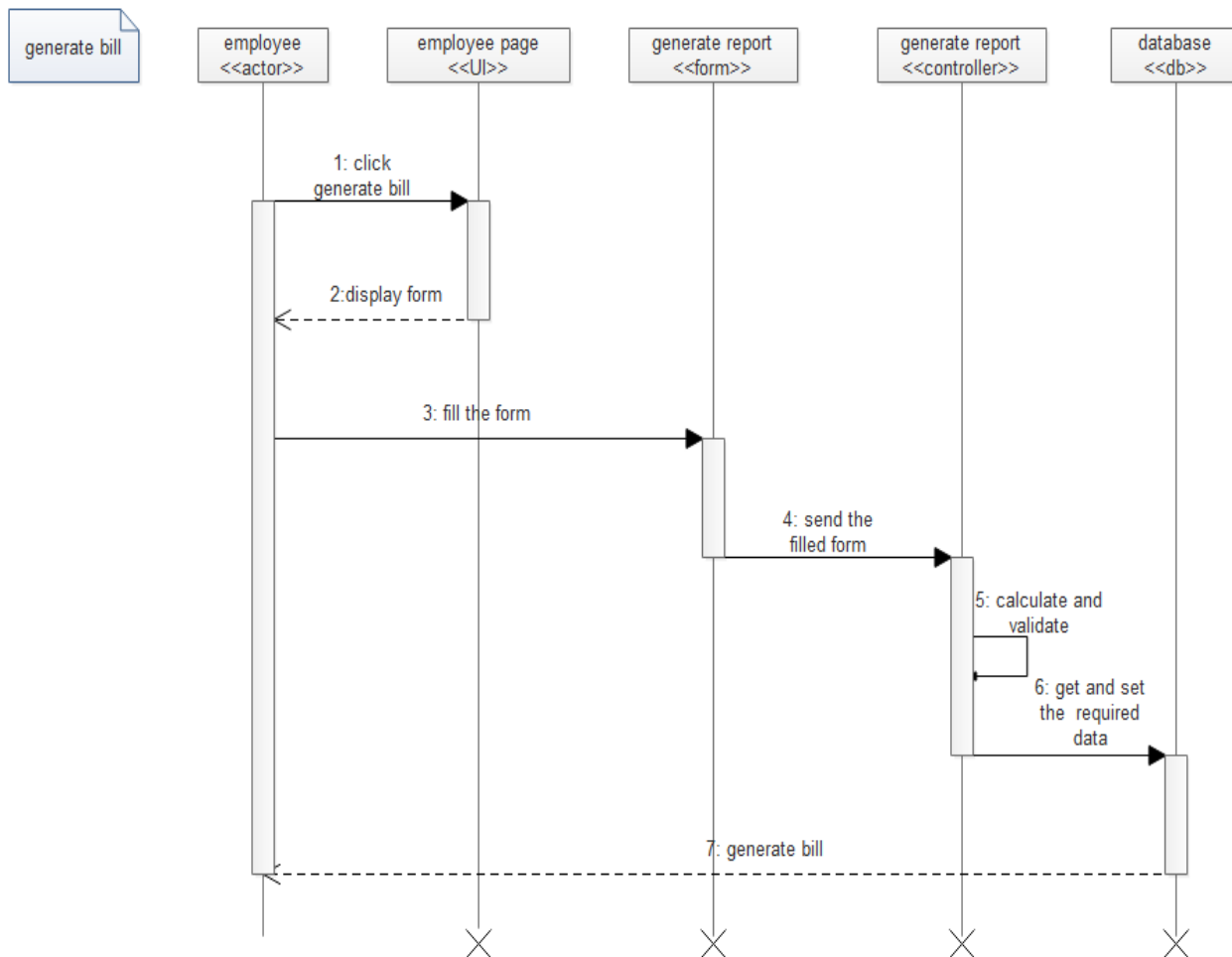


Figure 10 Bill generate sequence diagram

4.4.2. Activity diagram modeling

Activity diagram is the series of actions or flow control of activity. In activity diagram there must exist start and ending state.

- An Activity diagram is a type of flow chart with additional support for parallel behavior.
- Activity diagram is another important diagram in UML to describe dynamic aspects of the system.
- Activity diagram is basically a flow chart to represent the flow from one activity to another activity

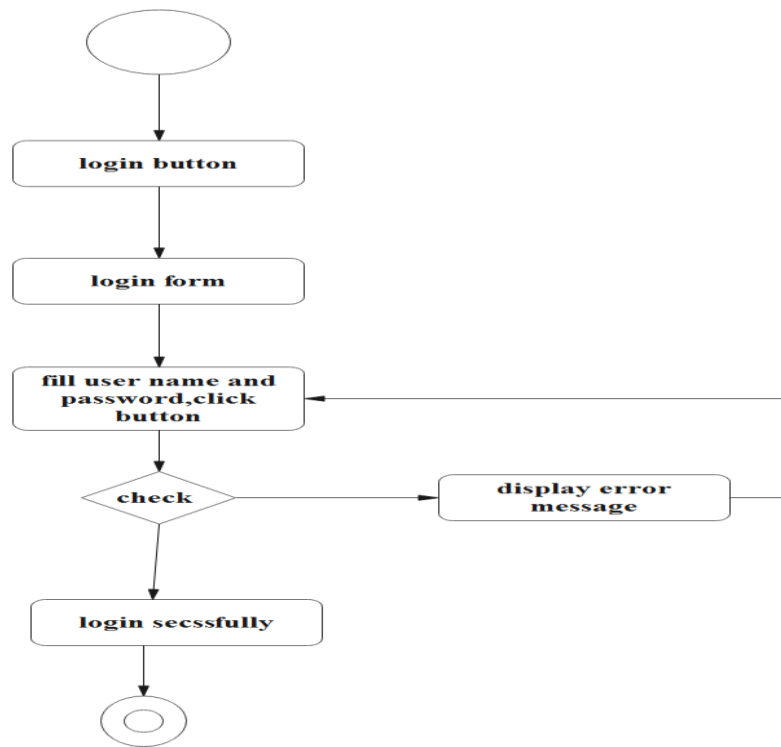


Figure 11 login activity diagram

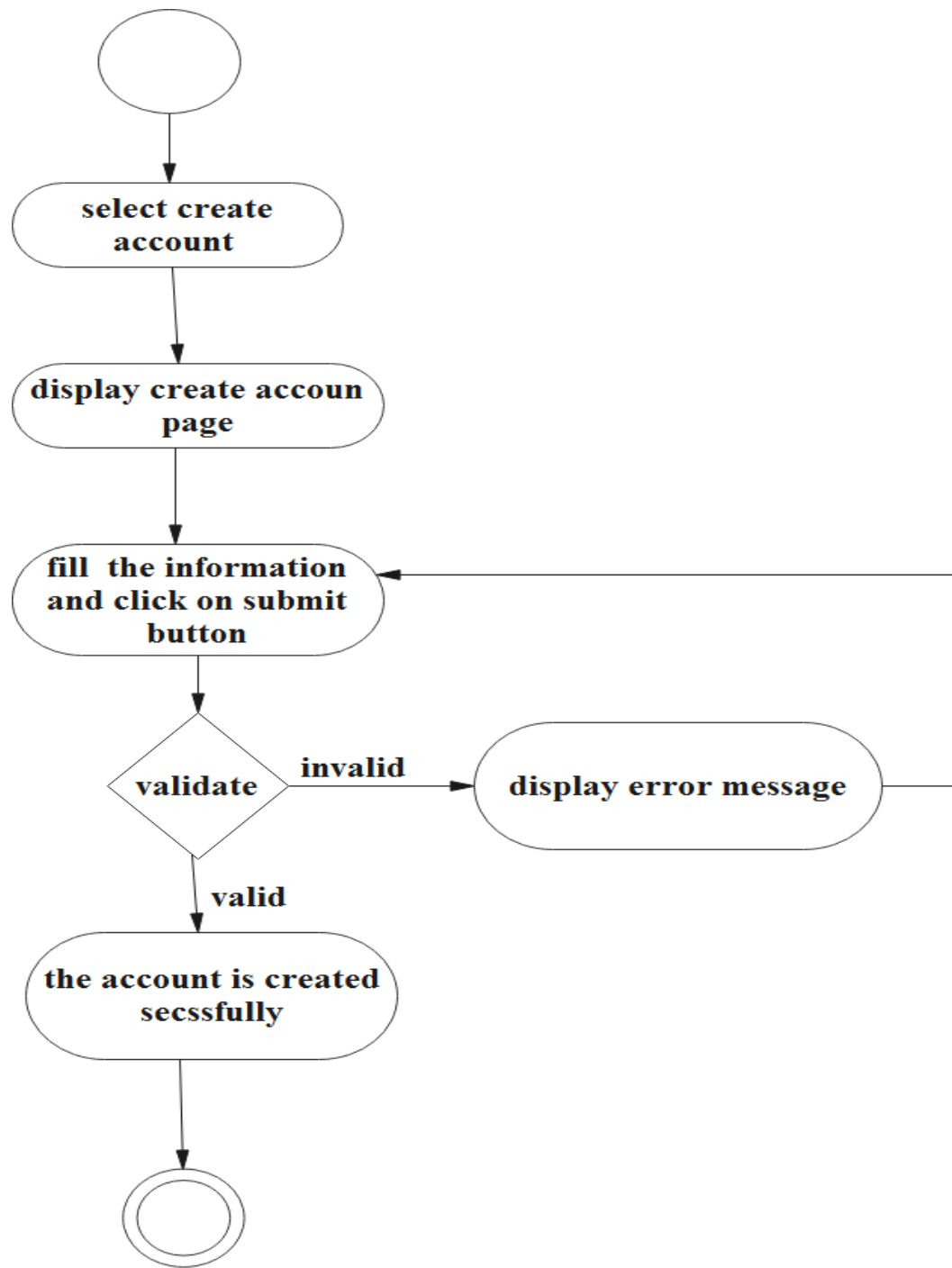


Figure 12 Create account activity diagram

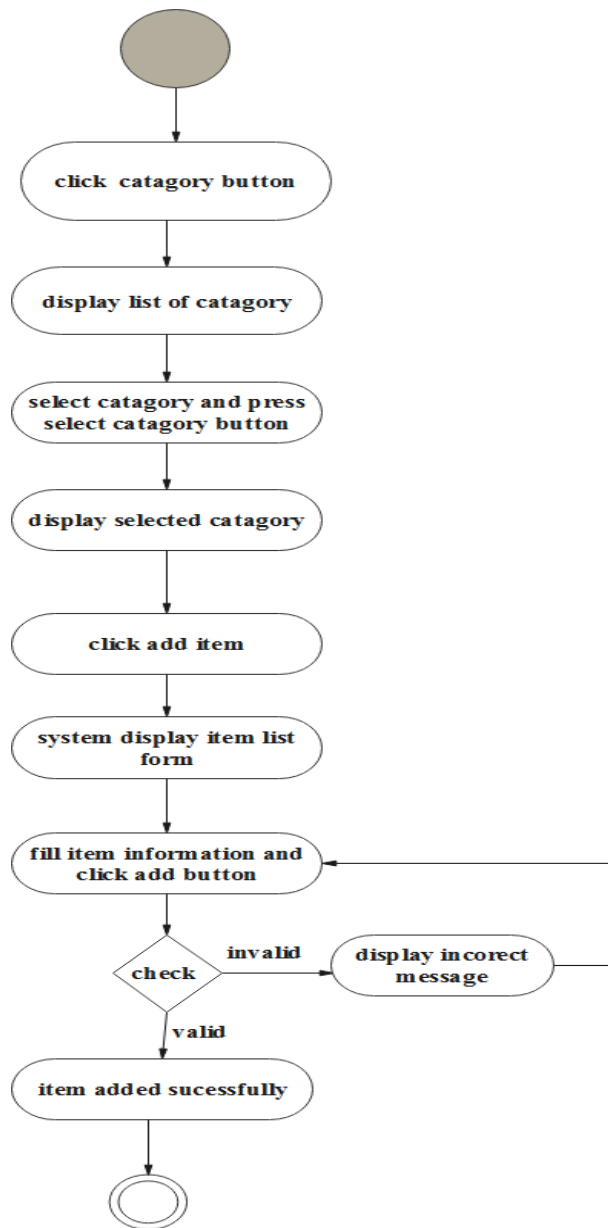


Figure 13 Add item activity diagram

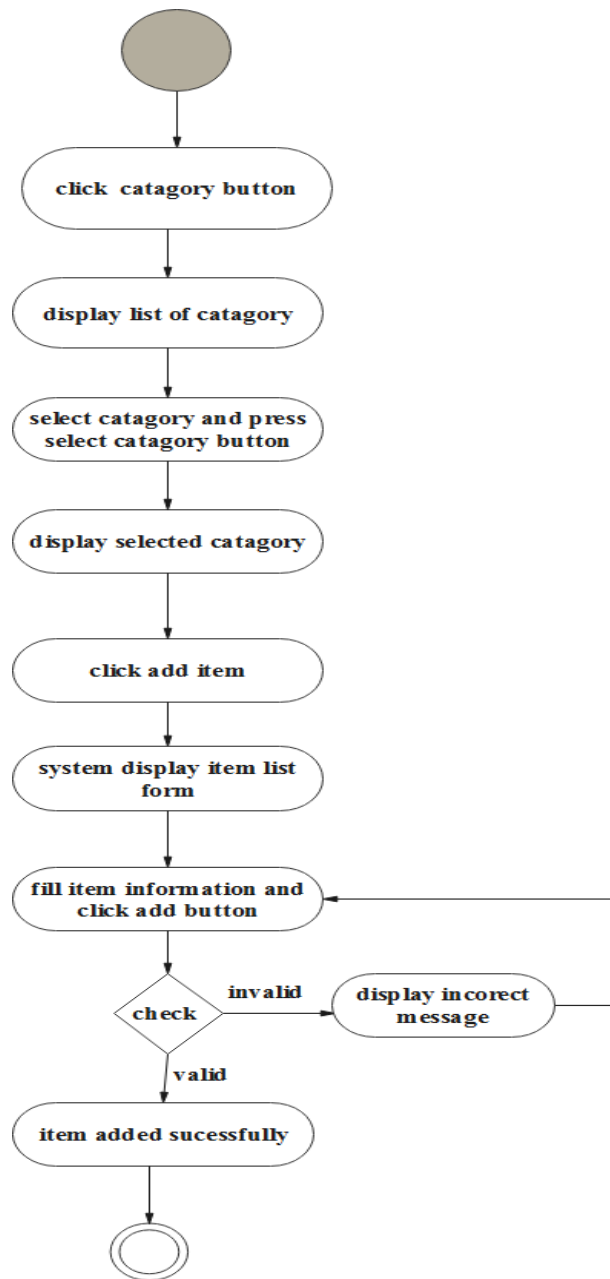


Figure 14 Update item activity diagram

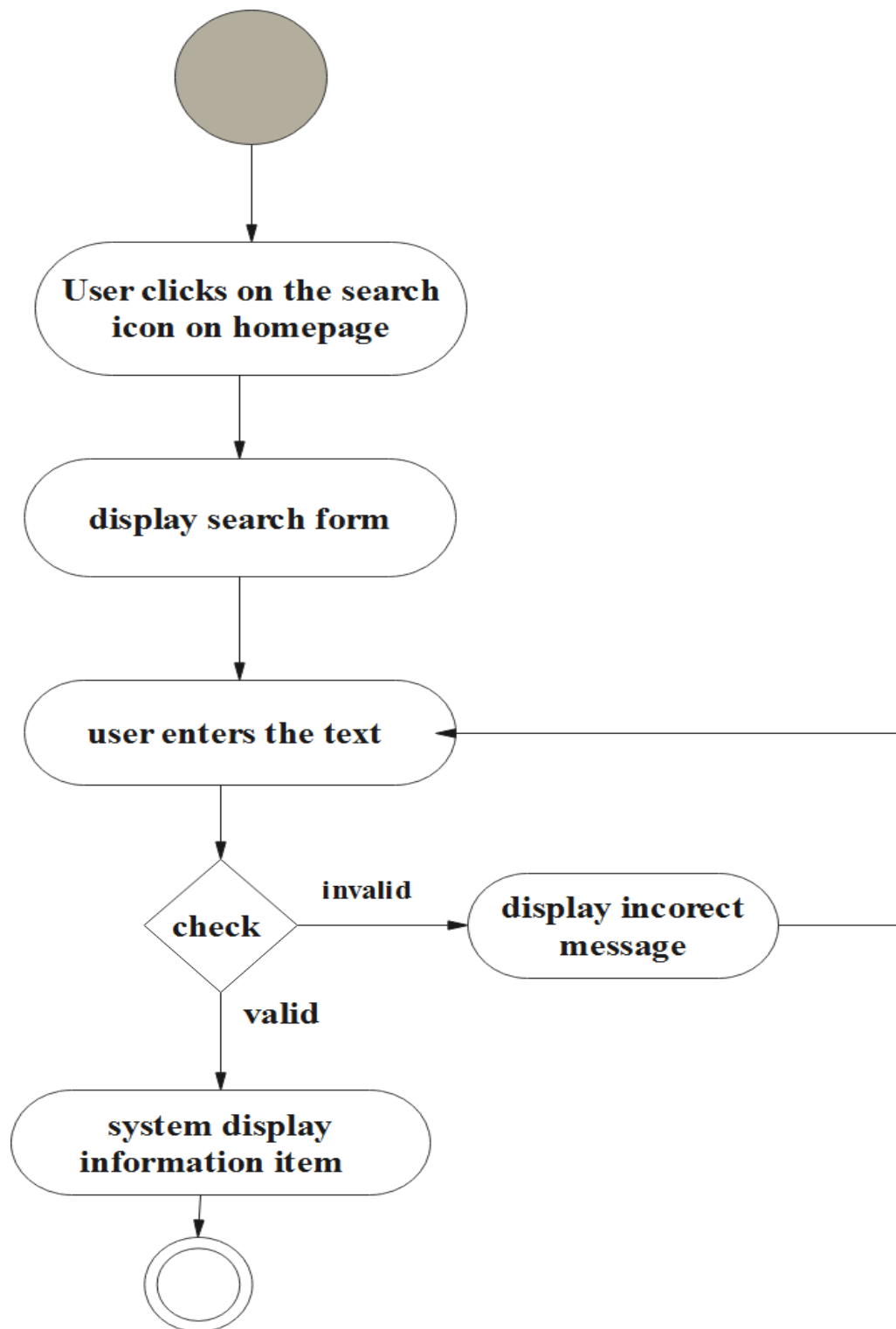


Figure 15 activity diagram search item

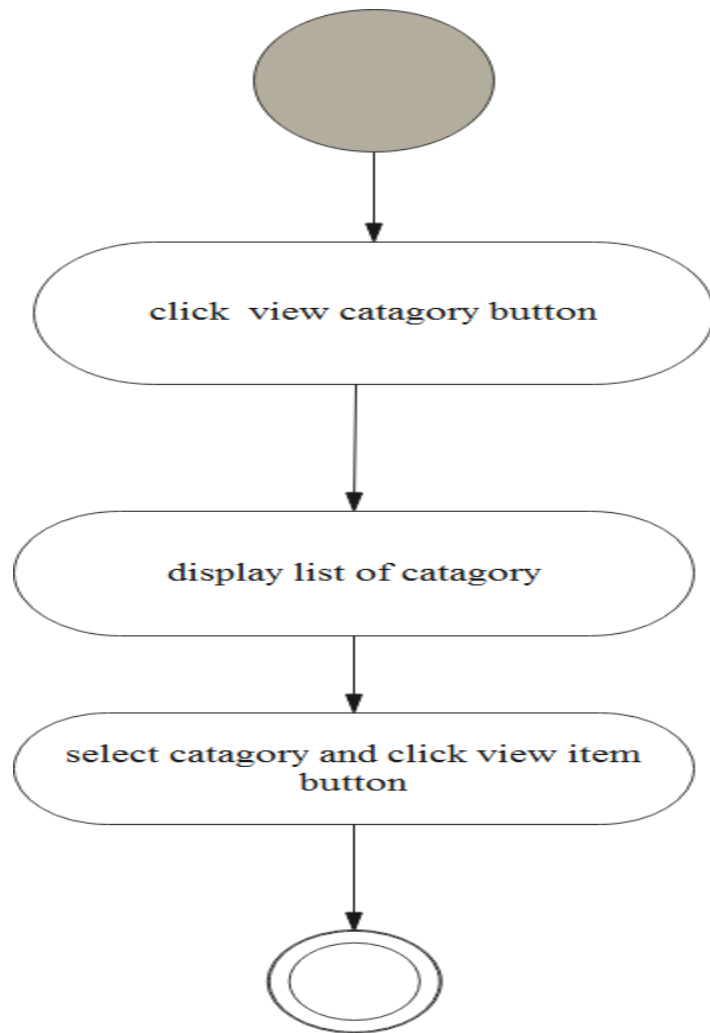


Figure 16 view item activity diagram

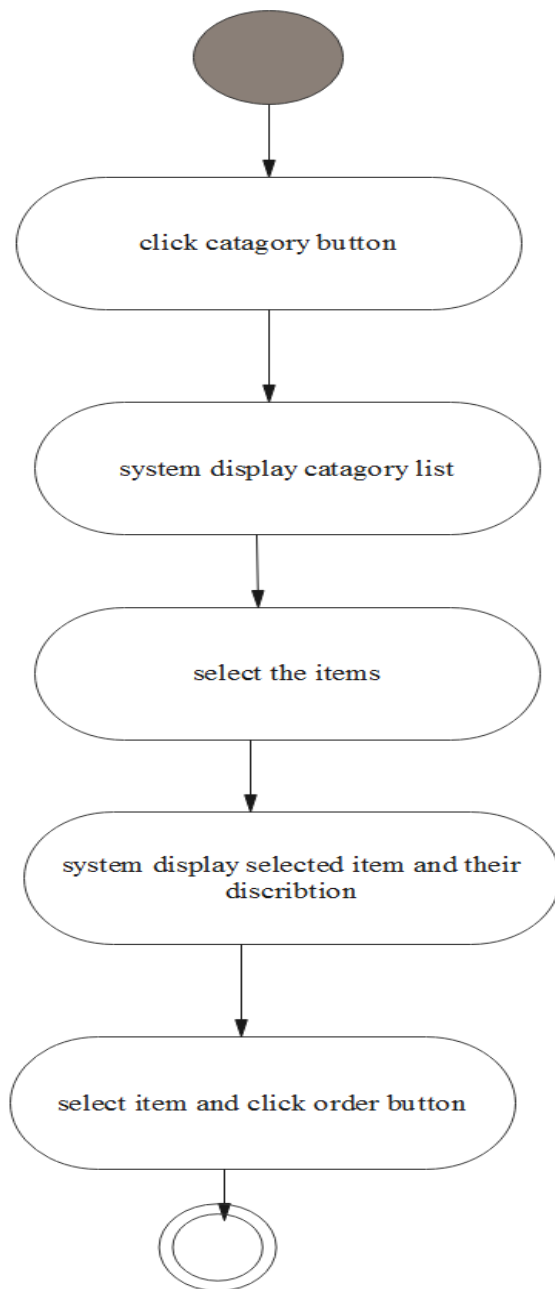


Figure 17 order item activity diagram

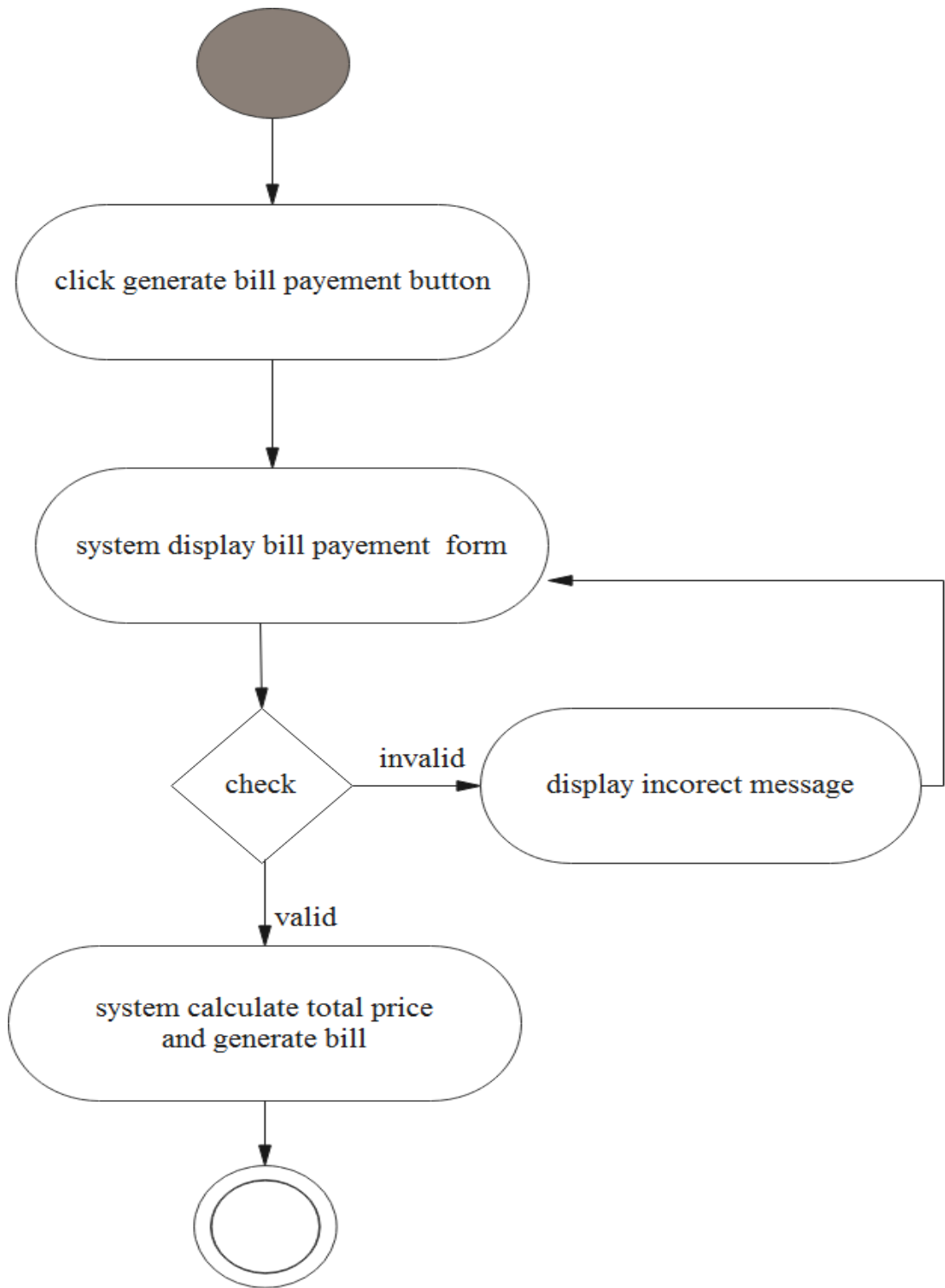


Figure 18 Bill generate activity diagram

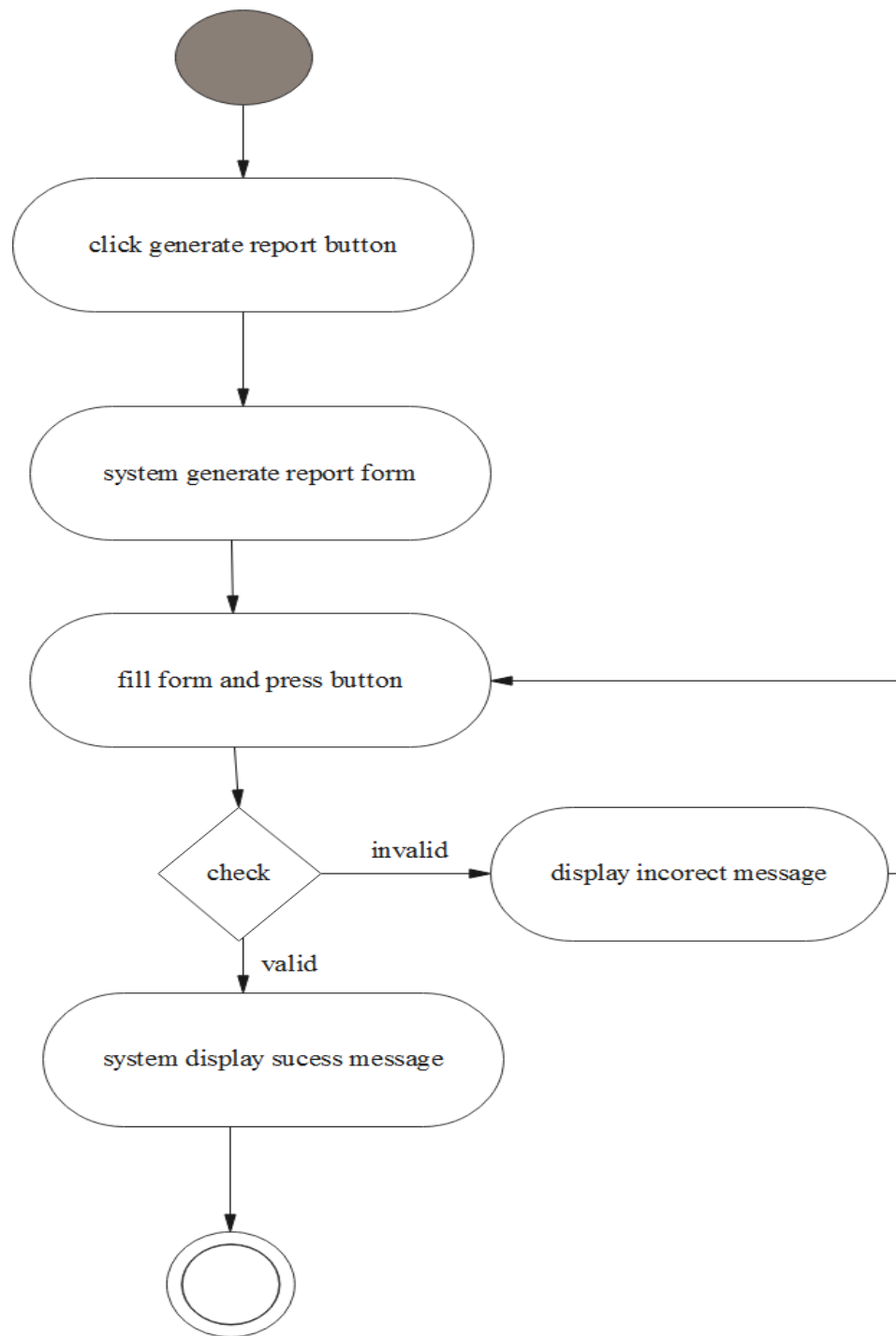


Figure 19 generate report activity diagram

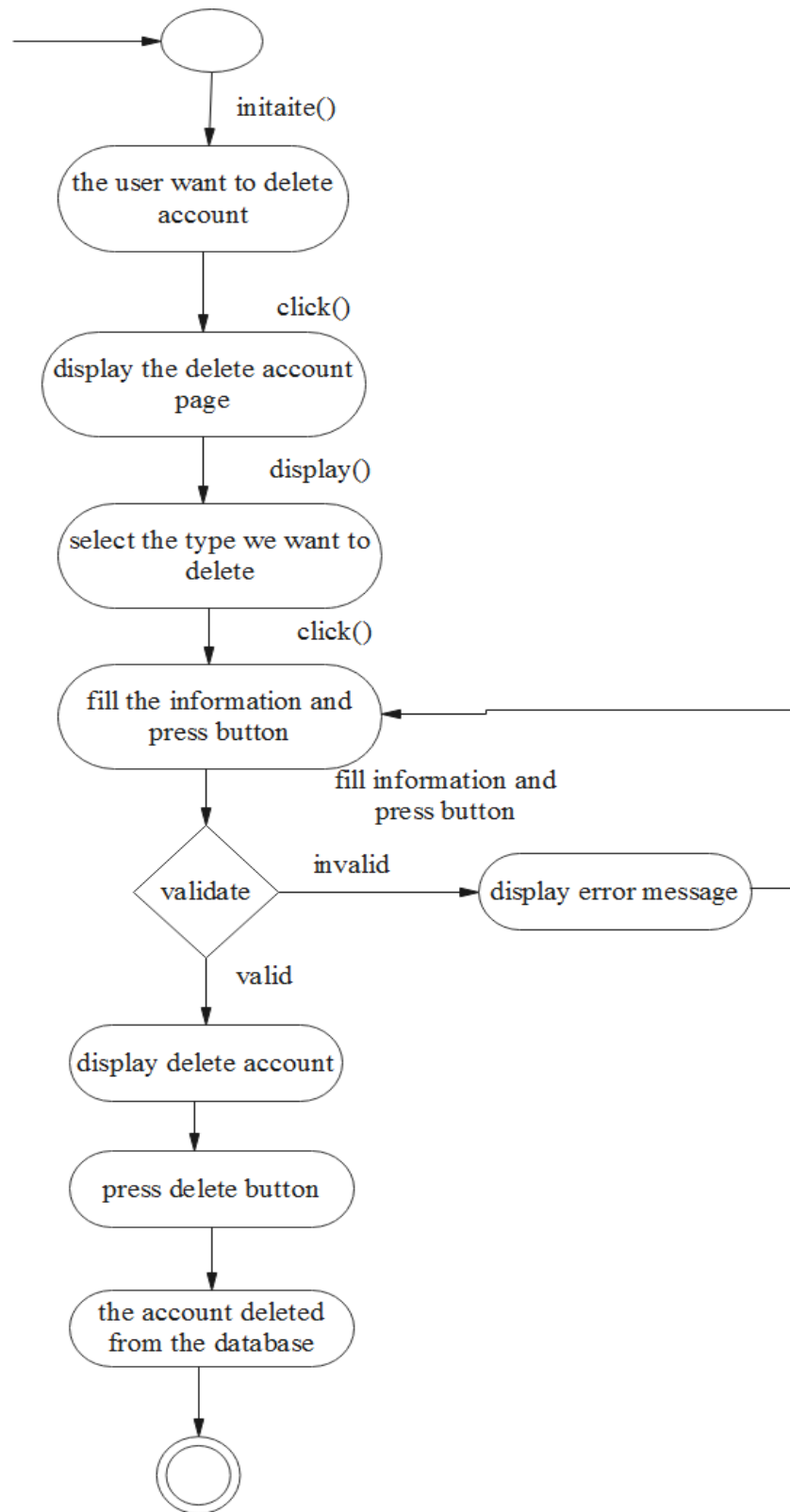


Figure 20 delete account activity diagram

4.4.3. State chart diagram

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So, the most important purpose of State chart diagram is to model life time of an object from creation to termination whenever performing different activities there will be some states that are included in activities so in our project ASMMS we have listed below some states that can be covered during the user interacts with the system.

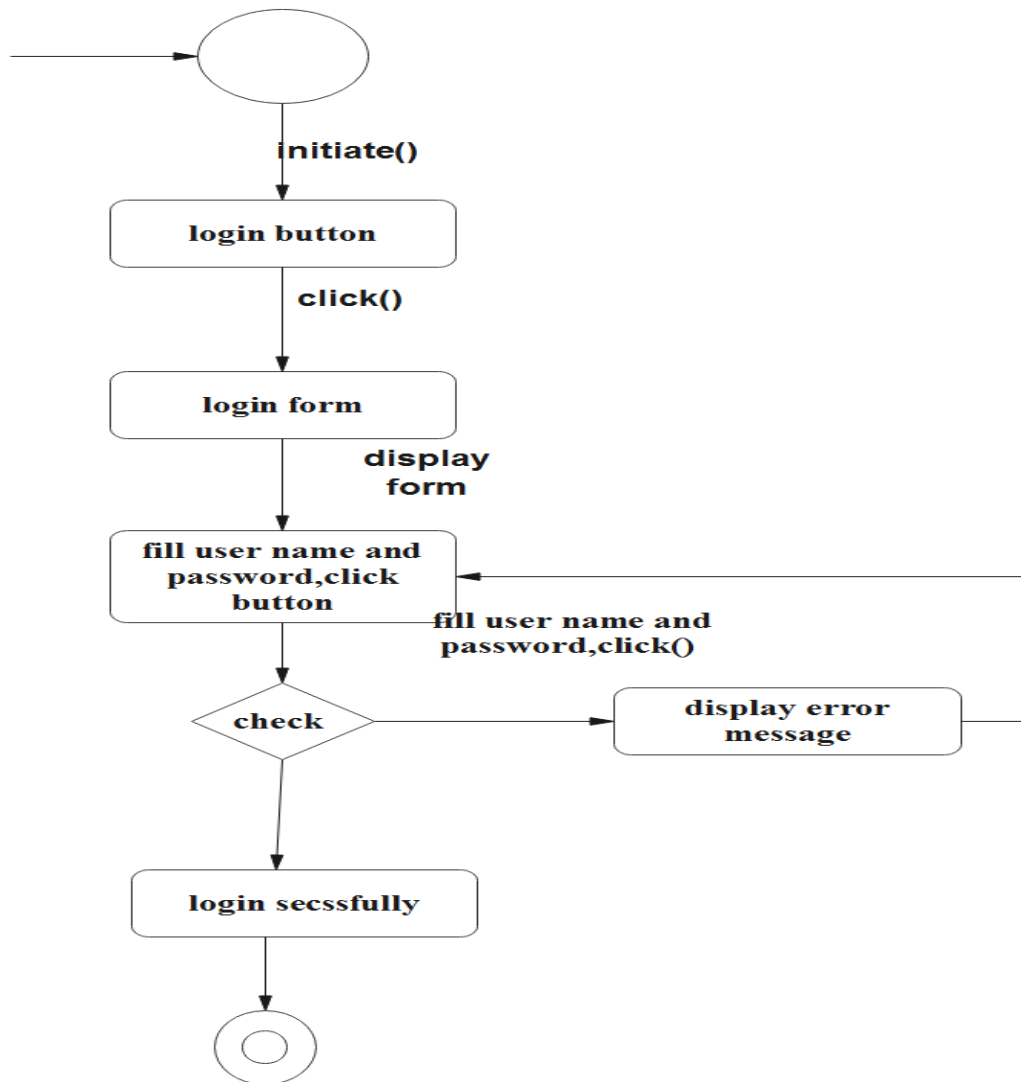


Figure 21 login state chart diagram

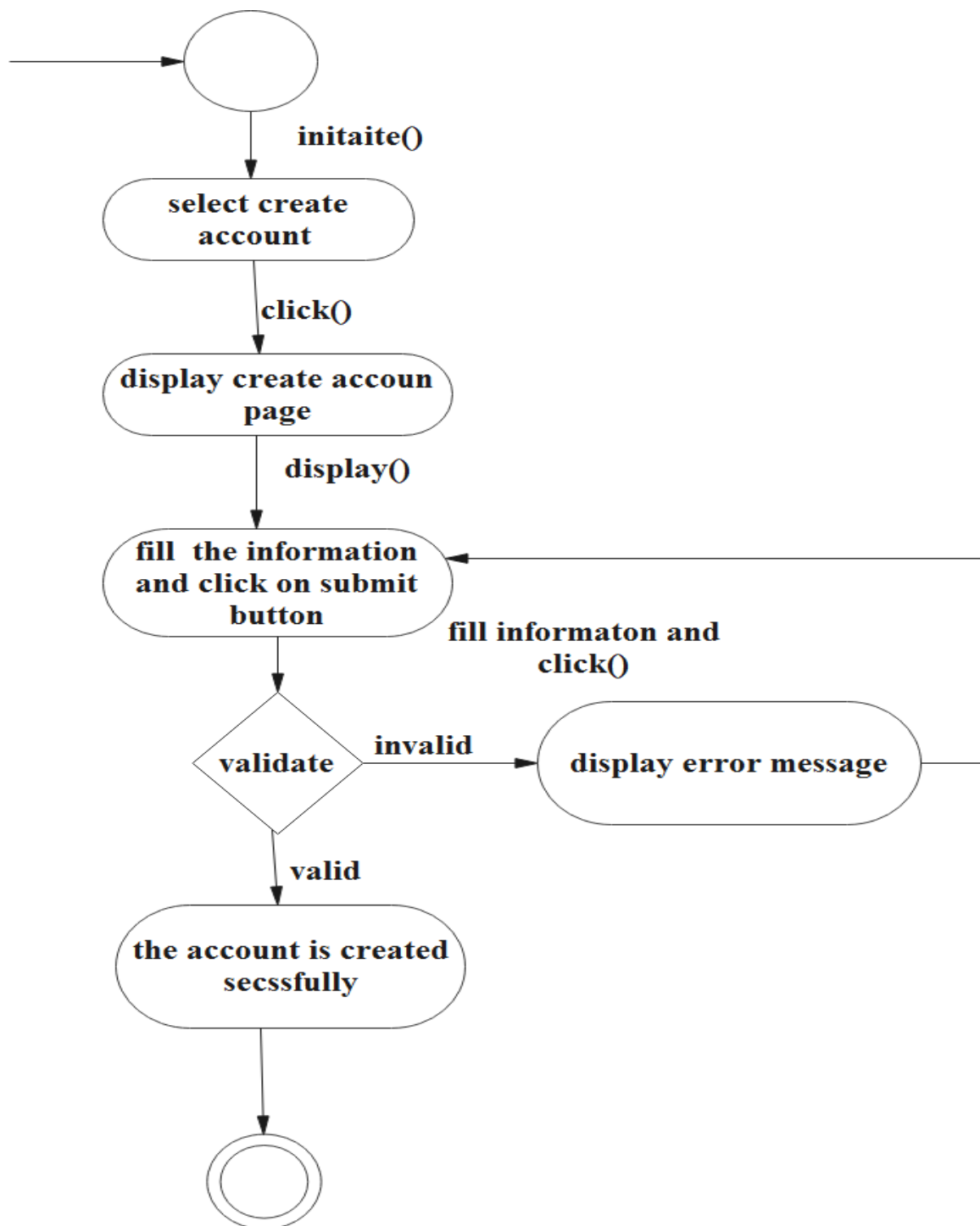


Figure 22 create account state chart diagram

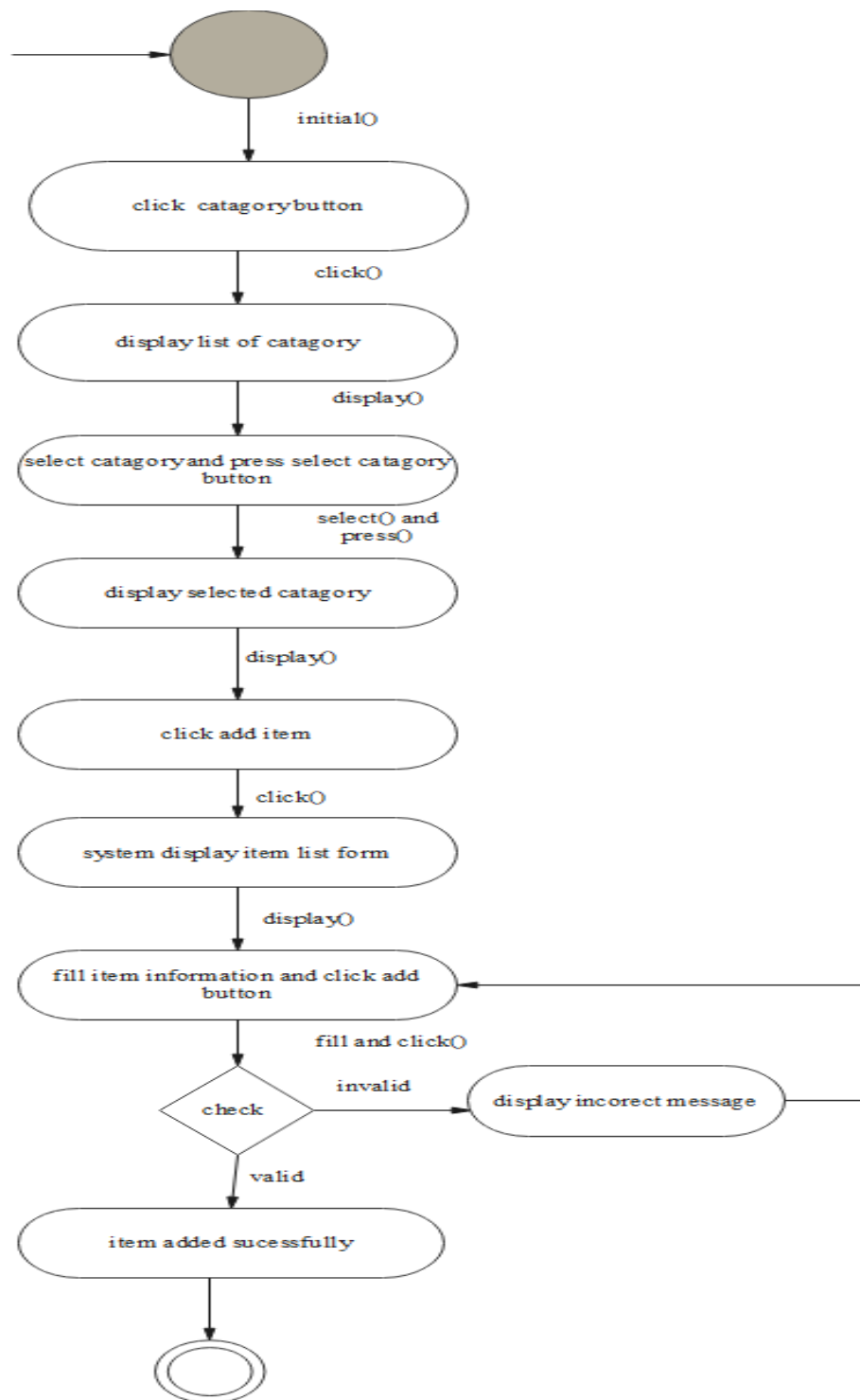


Figure 23 add item state chart diagram

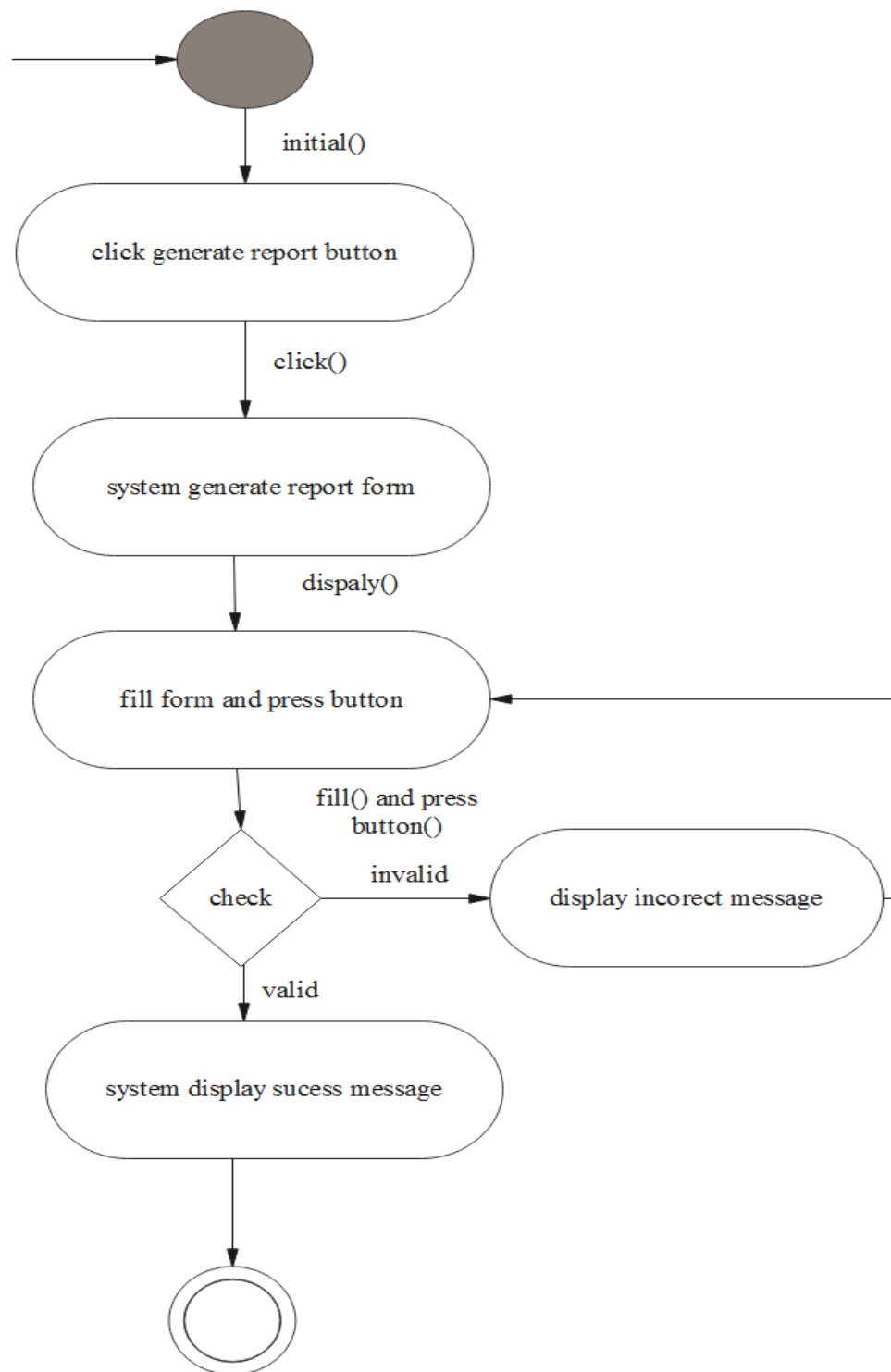


Figure 24 report generate state chart diagram

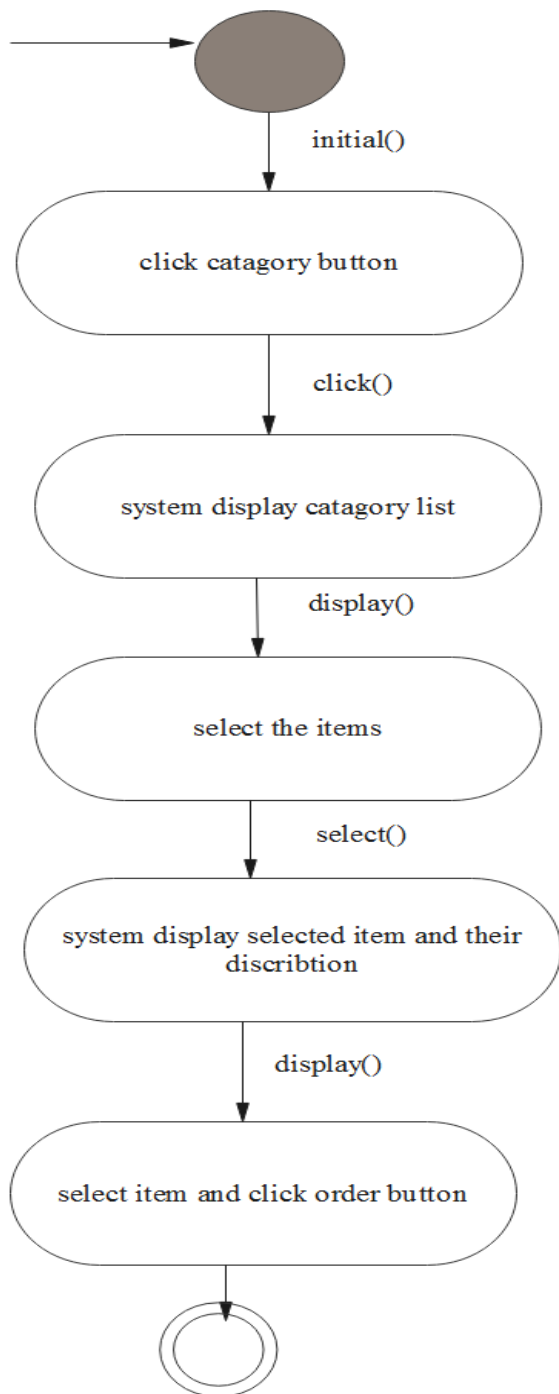


Figure 25 order item state chart diagram

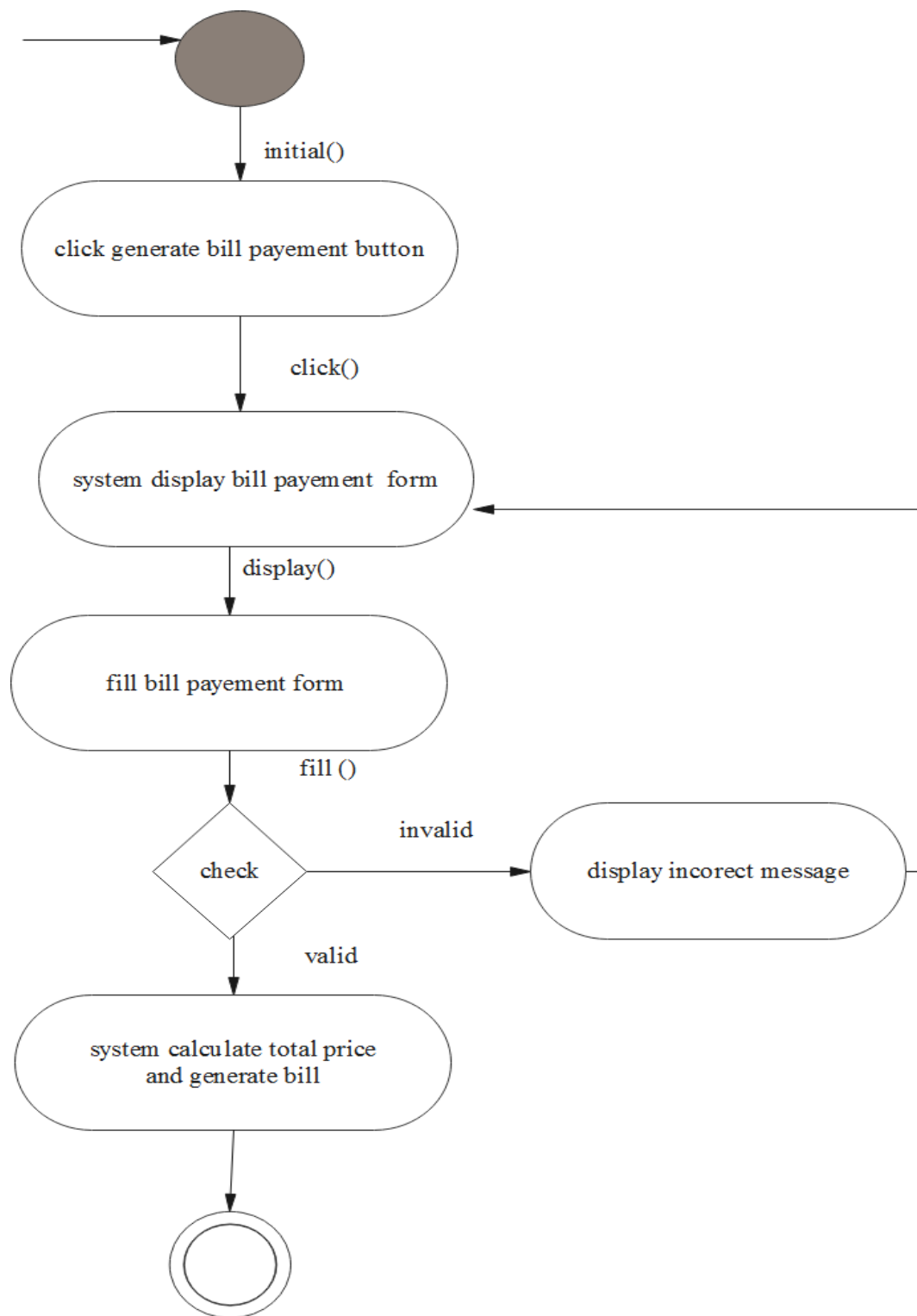


Figure 26 bill generate state chart diagram

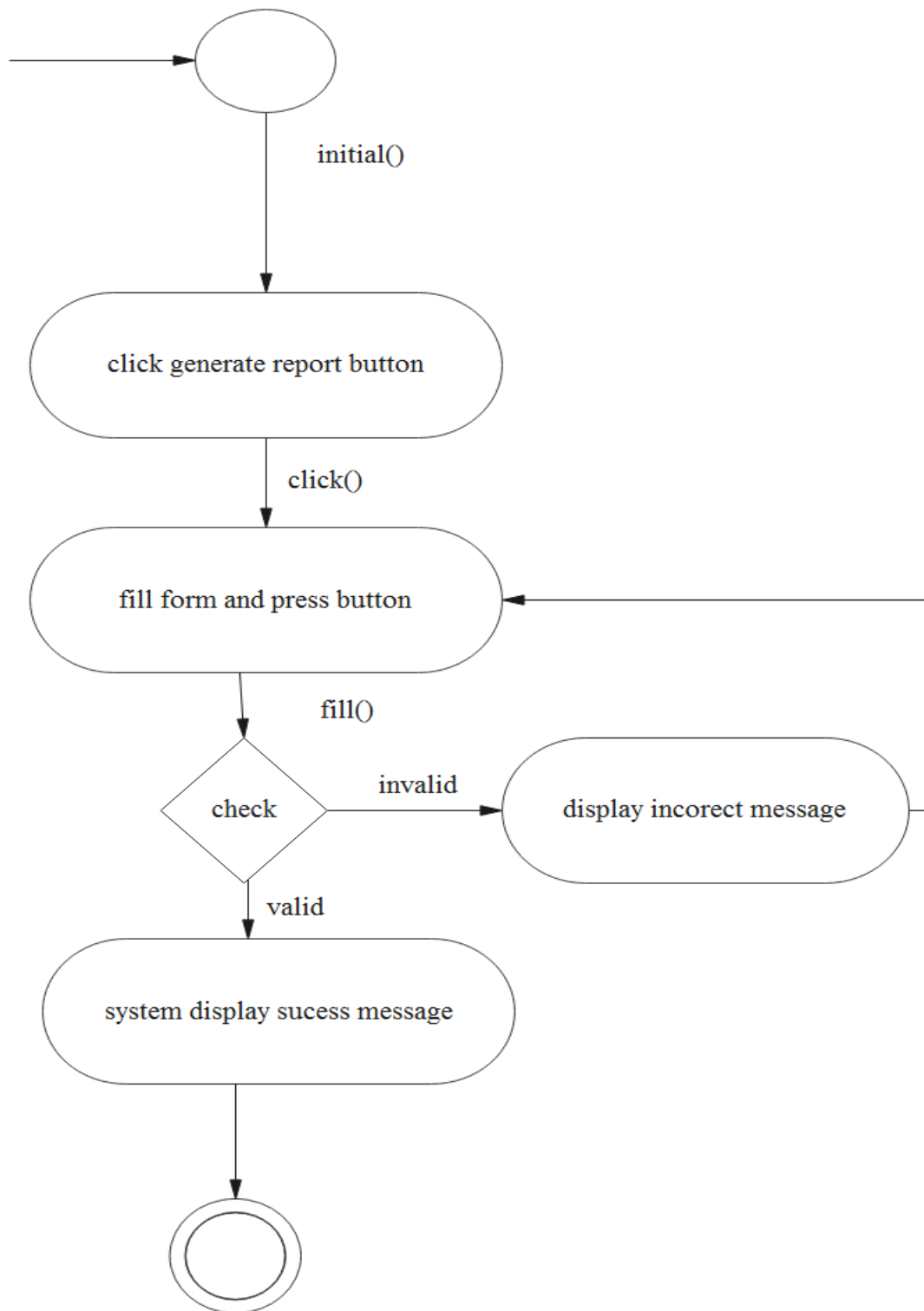


Figure 27 generate report state diagram

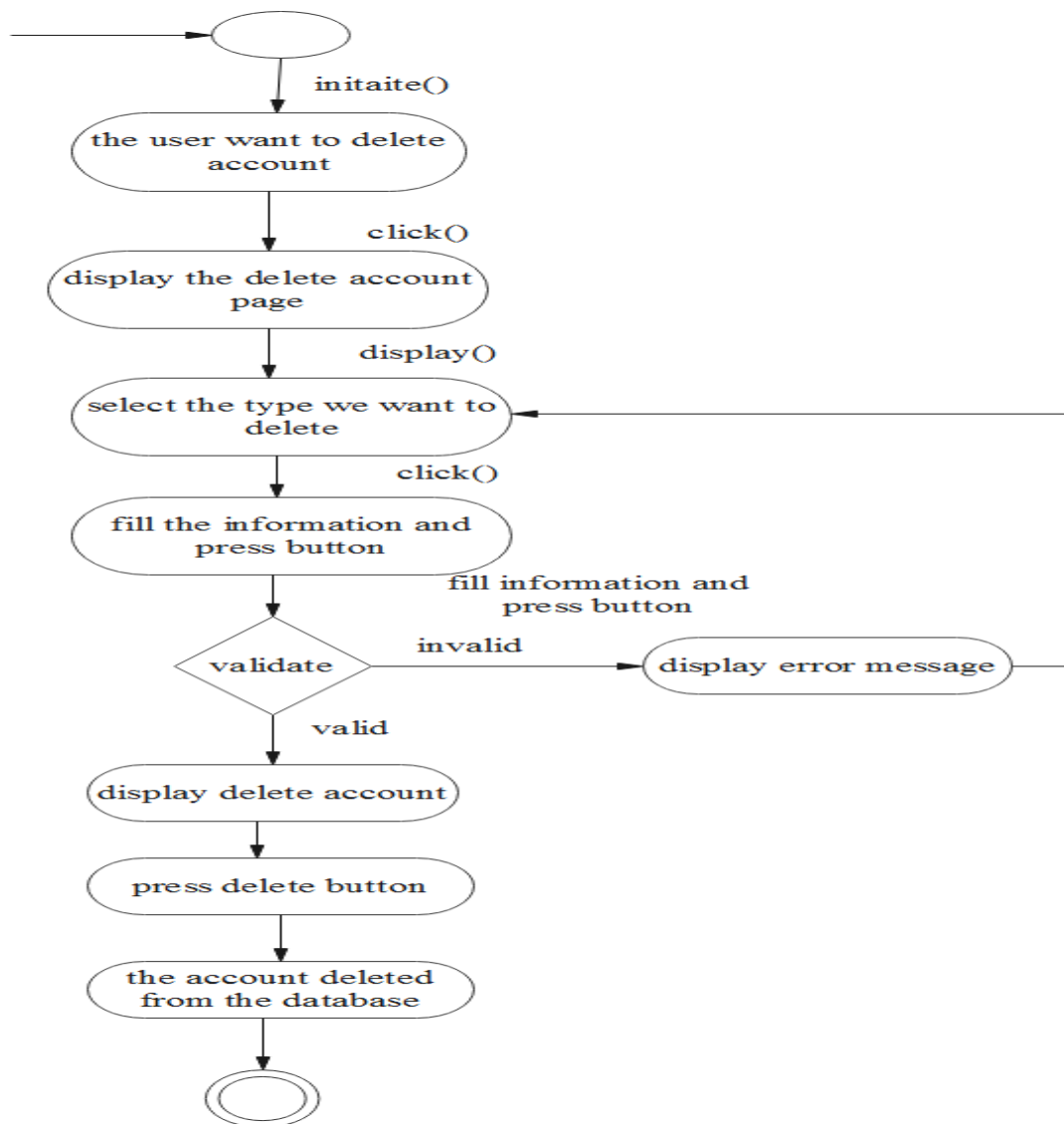


Figure 28 delete account state chart diagram

CHAPTER FIVE

5. SYSTEM DESIGN

5.1. Design Goal

The objective of designing is to model a system and to provide a high-level overview of the system architecture, and describes the data design associated with the system. In order to produce a good system, it is very significant to have a good design and the design must have a goal, which states the needed quality that must be incorporated (4).

5.1.1. Performance

Response time is the total amount of time it takes to respond to a request for service. So, in our system, the service time is the time it takes to do the work you requested generally the response time in our system is mostly short.

5.1.2. Dependability

Dependability is the ability to provide services that can defensibly be trusted within a time-period and also encompass mechanisms designed to increase and maintain the dependability of a system or software. So our system dependability is effectively high.

Robustness: Since the system is a web-based system that mainly uses a menu driven access there would not be an input problem by the user side. But for the server side there might be an error during the process of entering a data. In this time the system will provide an error page and the system will continue without failure or affection.

Availability: -as long as there is an internet connection the system will be available always.

Security: the system should be secured, i.e., not allow unauthorized users to access the database system.

5.1.3. Maintainability

The system must be maintainable. It facilitates the correction of errors and deficiencies and it can expand or be upgraded to satisfy new changes in requirement.

In time of failure or need modification the system needs to be maintained.

Modifiability: - If in the system, some functionality requires to be modified, this modification must be done specifically to that function or page without affecting the overall system organization.

5.1.4. End user

This is to mean that the user of the system after the completion of the system how the system is used friendly manner for both the experienced and inexperienced user using the user interface and this is clearly stated in the user's documentation.

5.2. Proposed system Architecture

Our proposed system is consisting of 3-tier architecture namely Presentation layer, Application layer, and Database layer. 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database.

Presentation layer: is client layer (User Interface) and topmost layer of the application. This is the layer we see when we use this system. It is the interface to our system which takes information from the user. The main functionality of this layer is to communicate with the application layer. This layer passes the information which given in terms of keyboard action and mouse click to the application layer. Example when user want to login into ASMS first see three text box and login button to enter user name, user type and password and click on login.

Application (Middle) layer: it is application layer which interacts with data layer and sends information retrieved from database toward to presentation layer.it act as the mediator between presentation layer and data layer. From above example once, the user clicks on login button application layer interact with the database and send information towards to presentation layer.

Database (Data) layer: which used to store data entered by the user? In general client of our system use browser to access the system using the internet. In this case when the user enters input and takes certain action application server process client request to interacting with the database server. Below figure is proposed system architecture

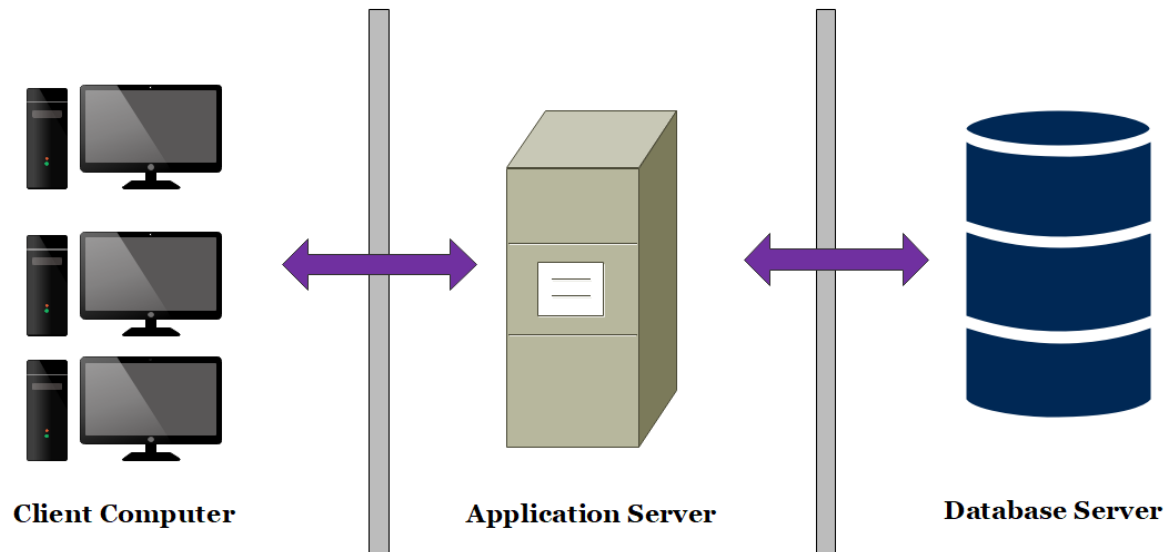


Figure 29 Proposed system architecture

5.2.1. Subsystem Decomposition and Description

Subsystem decomposition is the process of dividing the system into manageable subsystems from the analysis model of the proposed system. The goal of the system decomposition is to reduce the complexity of design model and to distribute the class of the system into large scale and cohesive components. Components are generally units of computation or data stores in the system. A component has a name, which is generally chosen to represent the role of component or the function it performs. The different components of a system are likely to interact while the system is in operation to provide the service expected of the system. In our system the following subcomponent is available.

- **system subsystem**-- responsible for perform the following task
 - generate notification
 - calculate profit
 - print receipt
 - display detail item info
- **manager subsystem**: responsible for perform the following task
 - Manage account
 - Mange category
 - View report
 - View notification
 - Calculate profit
 - Mange employee

- Change password
- **Customer subsystem:** perform the following task
 - order item
 - view Item category
 - Create account
 - check expire date
 - view item
 - receive receipt
 - change password
- **Employee subsystem**
 - Mange item
 - generate report
 - bill generate
 - view item
 - sales item info
 - view Item category
 - check expire date
 - Search item
 - Receive request
 - View notification
 - Change password
 - Send receipts.

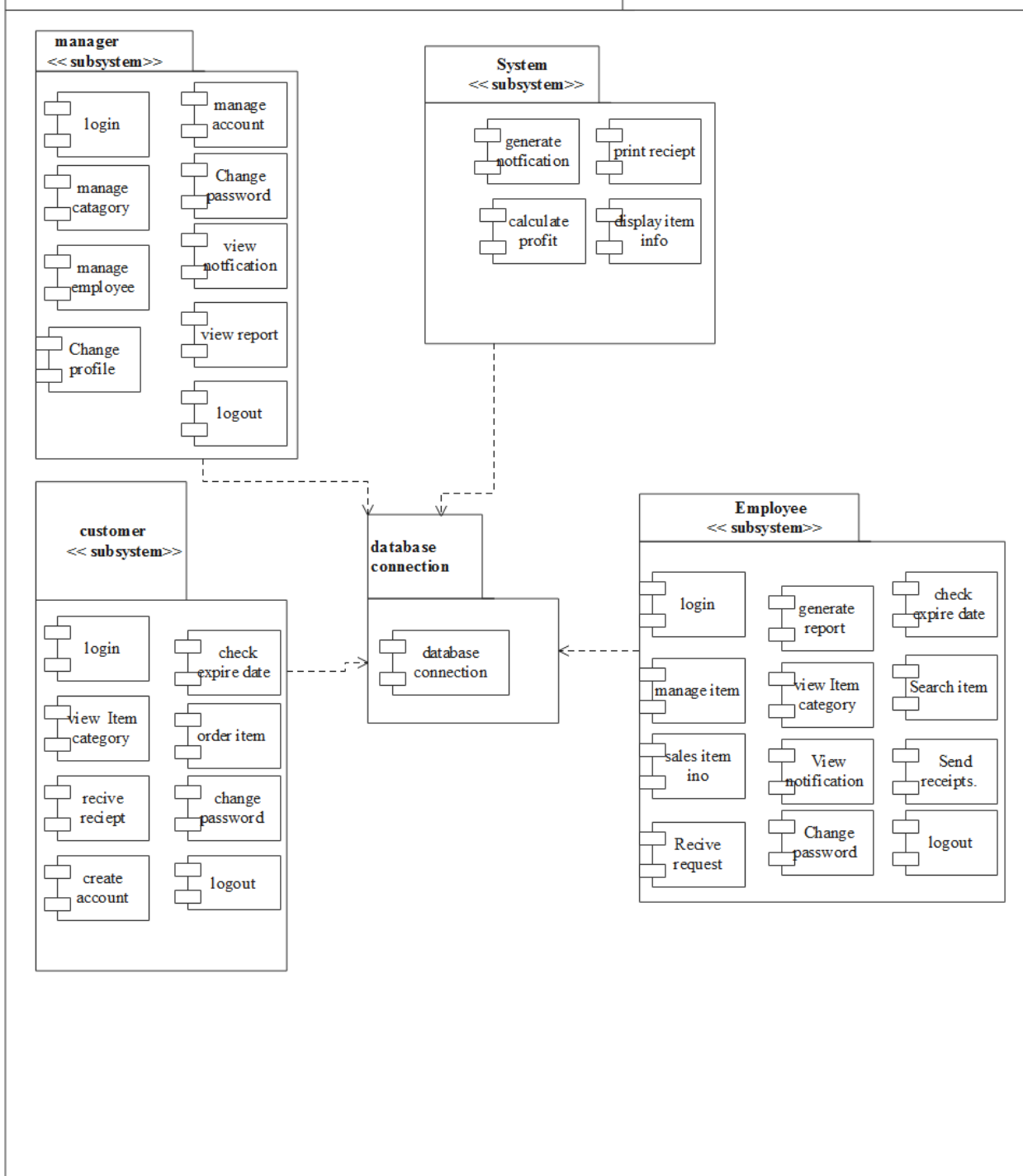


Figure 30 Subsystem Decomposition

5.2.2. Hardware/Software Mapping

Deployment diagram depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes. In other words, deployment diagrams show the hardware for your system, the software that is installed on that hardware, and the middleware used to connect the disparate machines to one another. You want to create a deployment diagram for applications that are deployed to several machines. It also shows how the software and the hardware components work together. Deployment diagram used to show the hardware of the system, the software that is installed in the hardware and also the middleware that is used to connect the disparate machines to one and other. It also shows how the software and hardware components of the system work together. The figure 5.4 below shows the deployment diagram of our system

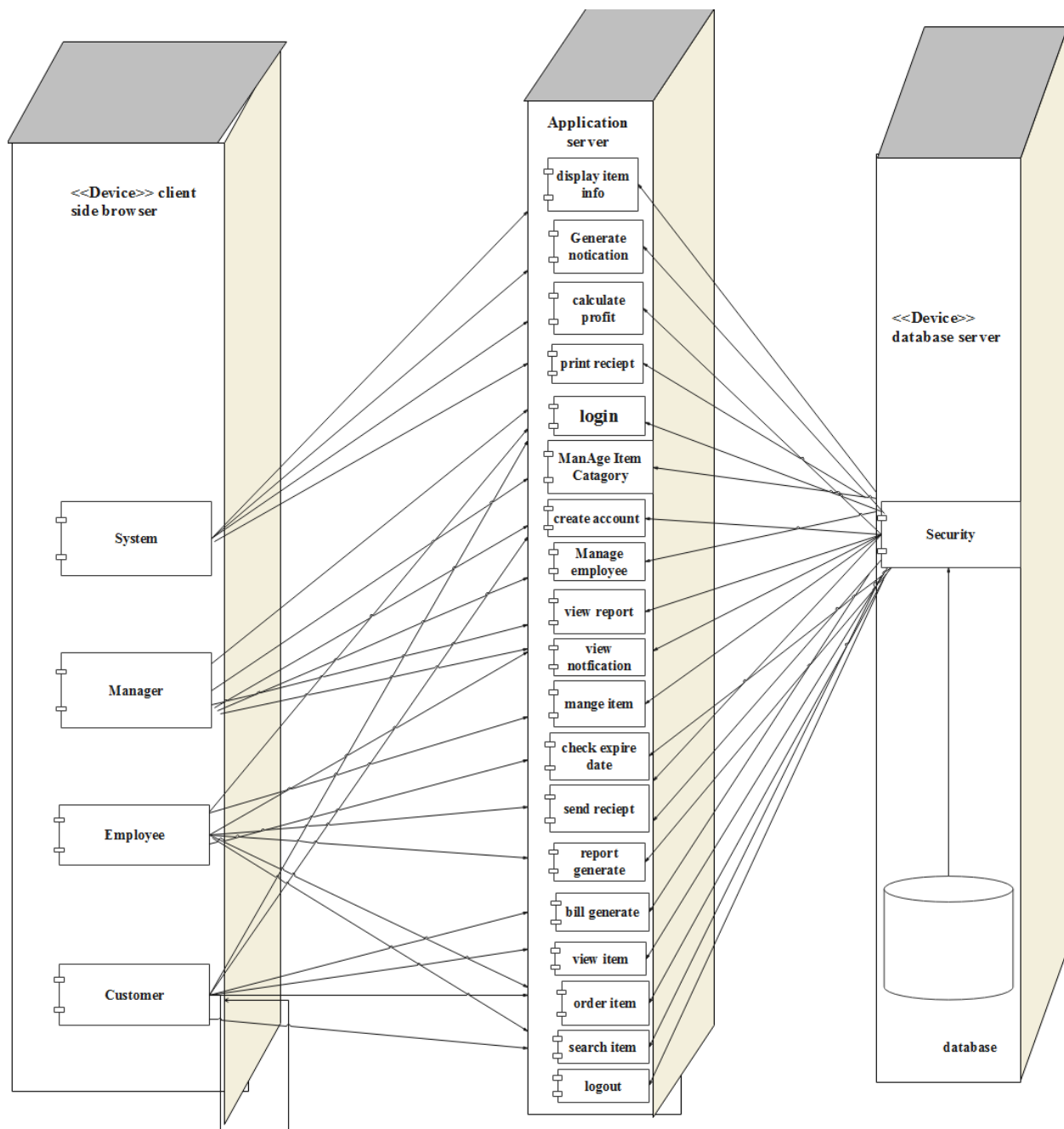


Figure 31 Deployment diagram

5.2.3. Detailed Class Diagram

Detail class diagram is class diagram that include attributes, methods, attribute data types, visibility of attributes and methods, inheritance, association, aggregation, composition, dependency, and municipality (cardinality and optimality). The following figure, uses the UML

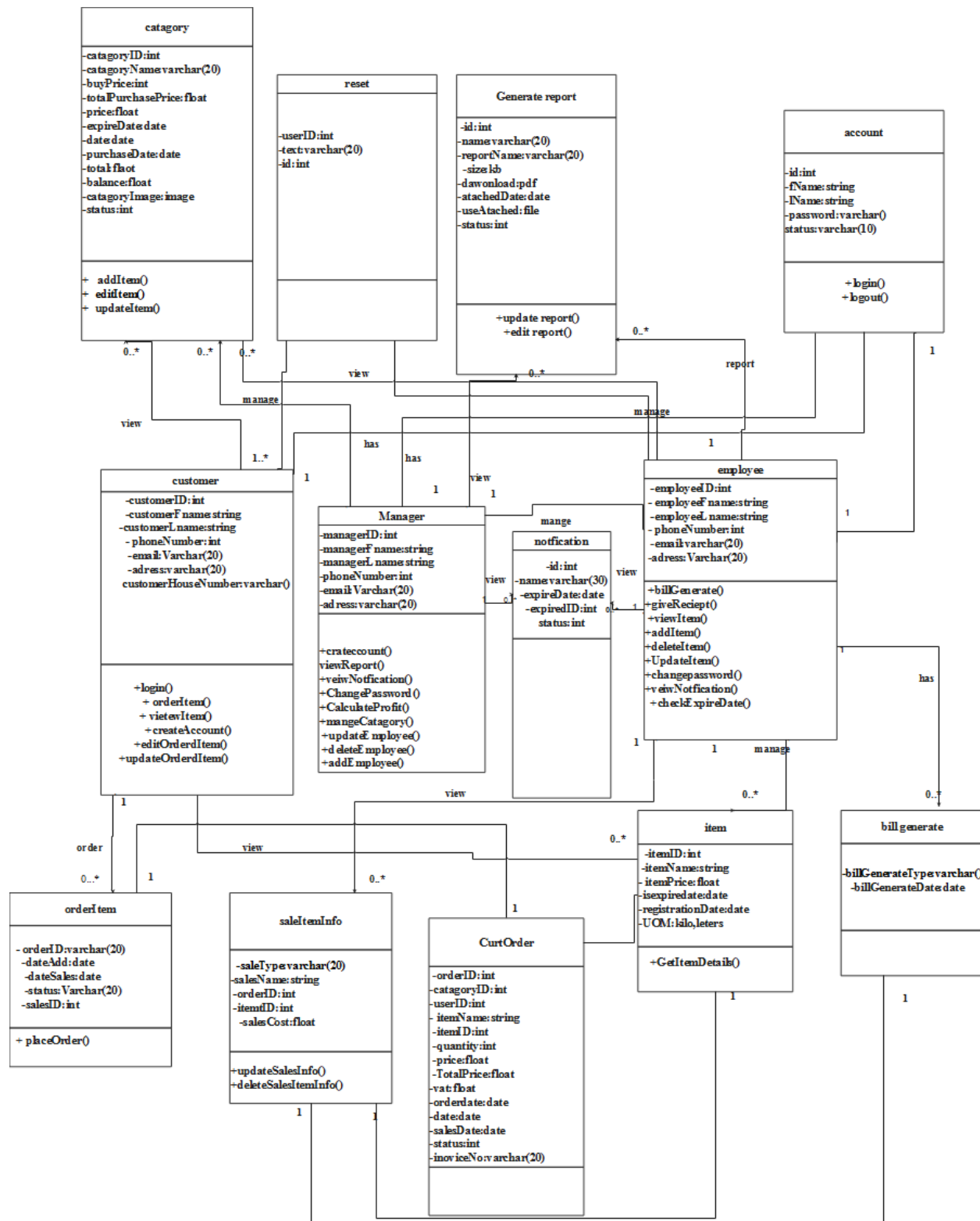


Figure 32 Detailed class diagram

5.2.4. Persistent Data Management

Persistence modeling is used to communicate the design of the database, usually the data base to both the users and the developers. It is also used to describe the persistence data aspect of the system. The following tables indicate the persistence data management of the system.

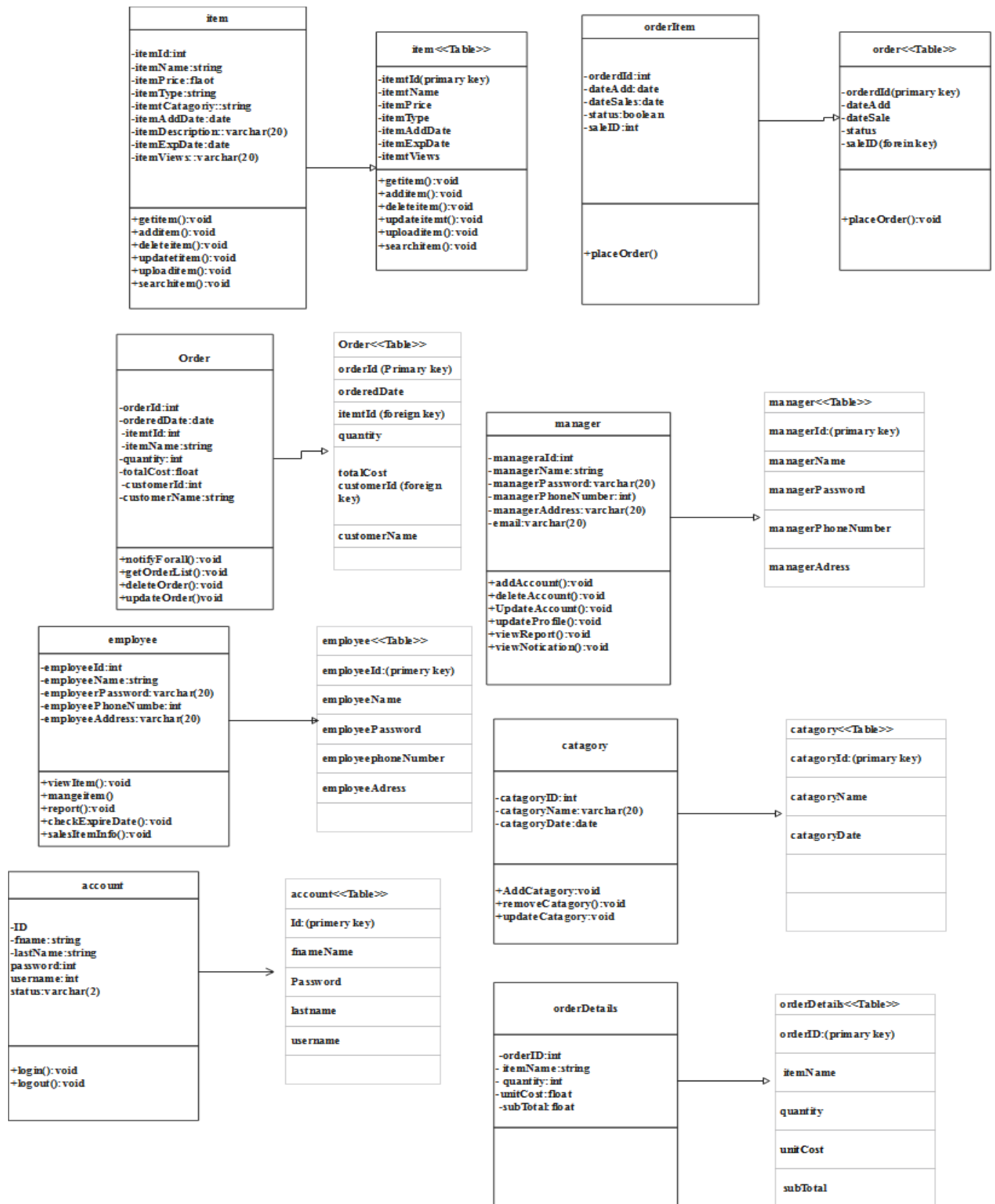


Figure 33 Persistent Data Management

5.2.5. Access Control and Security

In our system, different actors have access to different information and data. Access control and security specifies what the user can access or what cannot perform by some users. This access control is verified by username and password. System admin represents an authenticated user. The proposed system follows multi user system. In multi user system, different actors have access to different functionality and data. Then it must be having:

- **Confidentiality:** Only authorized person can see the information. Private data is kept private; personal privacy is respected.
- **Integrity:** There are limits on who can change the data in this system.
- **Availability:** The system is available at all times to authorized users.

ACTORS	OPRATIONS						
	Account	Report	Order	Manage item	Category	Manage employee	item
Manager	Manage account	view	-----	-----	view	Add Delete update	view
Employee	Change password	addReport	-----	Add Update delete	view	-----	view
Customer	Change password	-----	Order	-----	view	-----	view

Table 22 Access Control and Security

5.3. Packages

A package is a UML construct that enables to organize model elements and it enables to logically modularize a complex diagram in to package.

Package is an organized and functionality-based set of related interfaces and classes. Packages organize classes that belong to the same category or provide similar functionality.

In our system we category into account management package, notification management package, report management package, order management package, item management package, security management package.

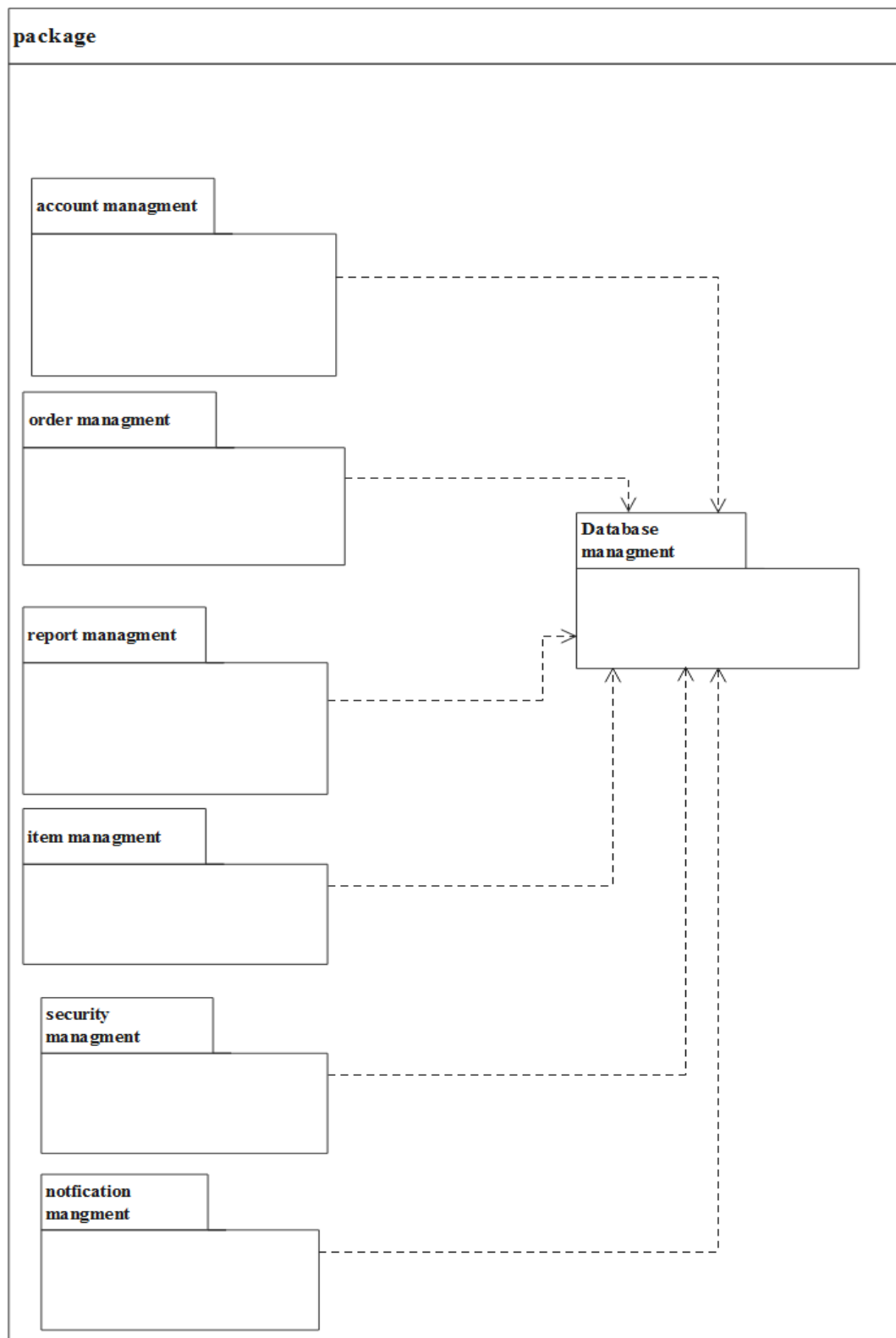


Figure 34 Package diagram

5.4. Algorithm Design

Algorithms are designed to show the flow of programs in the system. They are semantic driven rather than syntax driven. That means, the rule of syntax is not respected as other programming language but it has a complete meaning as that of syntax-based programming language. In addition, Algorithms show the flow and steps of logic in each function. This design part is important in the coding part of implementation. Some of the algorithms are listed below.

Example: -

➤ Login

- Home page displayed.
- User click on sign in link
- System display sign in page.
- User enters user name and password.
- **If** user name and password is correct, **then**
- **If** user is customer, **then** system display customer page.
- **Else if** user is employee, **then** system display employee page.
- **Else if** user is manager, **then** system display manager page.
- **Else** user name and password is not correct, then system display error message and redisplay login page.

5.5. User interface design

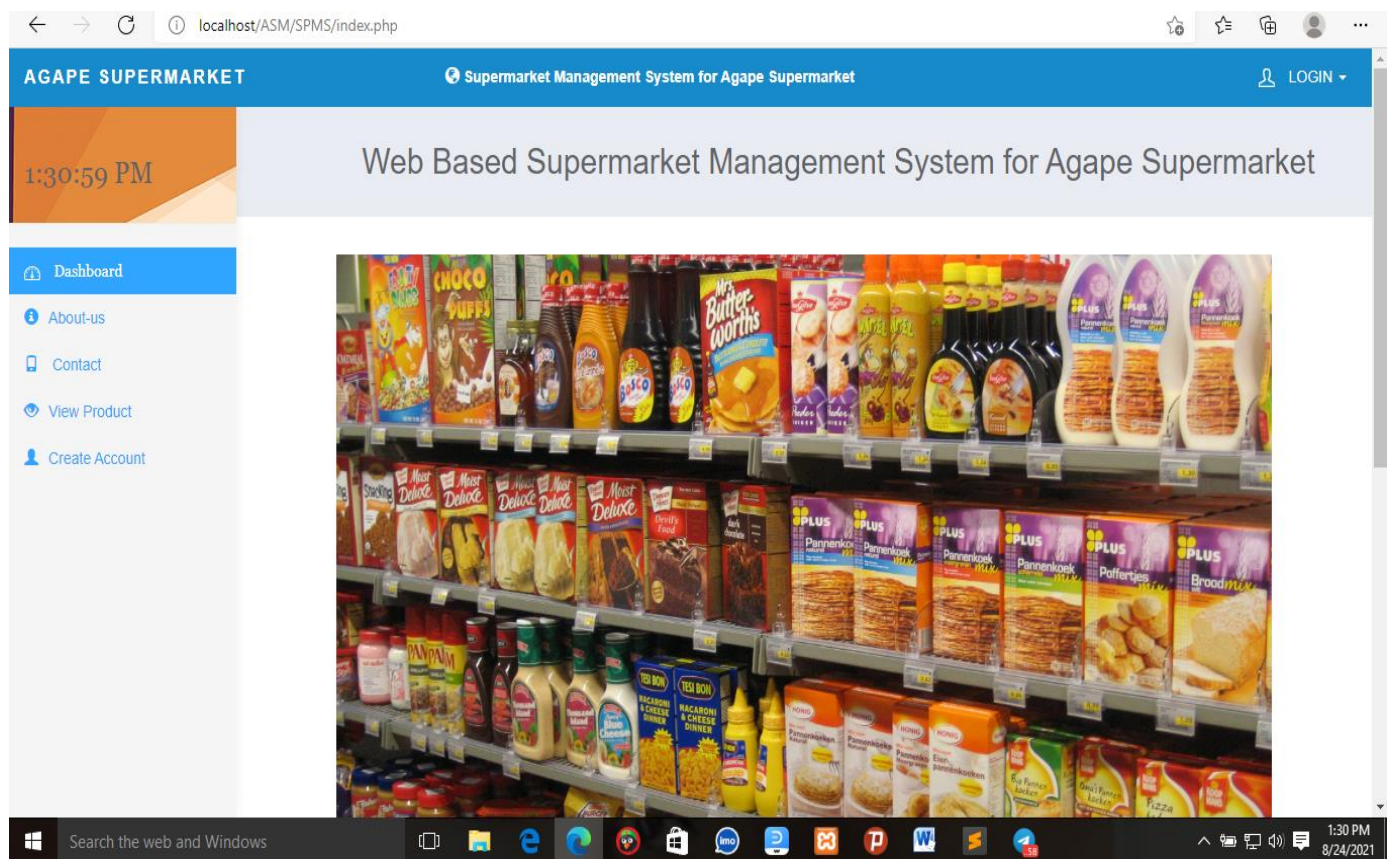


Figure 35 Home page user interface

Agape Supermarket

©ASM

 Please Enter Your Information





 Login

[back to home](#)

[forgot password?](#)

Figure 36 Login user interface

CHAPTER SIX

6. IMPLEMENTATION AND TESTING

Implementation is defined as a specified set of activities designed to put into practice an activity or program of known dimensions. It is the carrying out, execution or practice of a plan, a method, or any design, idea, model, specification, standard or policy for doing something. It is also a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment. It is the most crucial phase in which it transform the design and analysis of the system into a tangible system by writing the code to the system to be developed and make it operational and applicable by testing and debugging the functionalities that are done. This makes the implementation stage more essential step to develop the required system

6.1 Implementation of the Database

We use MYSQL database to manage our system which assists to insert, update, delete and view data which is stored in the database.

We use MySQL because it can be deployed and run on any operating system. MySQL is easy, fast and can be used for any type of database whether it is relational or simple, large or small.

-- phpMyAdmin SQL Dump

-- version 5.0.2

-- <https://www.phpmyadmin.net/>

--

-- Host: 127.0.0.1

-- Generation Time: Aug 24, 2021 at 08:49 PM

-- Server version: 10.4.14-MariaDB

-- PHP Version: 7.4.10

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

START TRANSACTION;

SET time_zone = "+00:00";

```

--
-- Database: `elm`
--
--
-- Table structure for table `category`
--
CREATE TABLE `category` (
  `category_id` int(200) NOT NULL,
  `ID` int(200) NOT NULL,
  `category_name` varchar(200) NOT NULL,
  `buy_price` varchar(200) NOT NULL,
  `total_purchase_price` varchar(200) NOT NULL,
  `price` double NOT NULL,
  `exp_date` varchar(100) NOT NULL,
  `y` int(10) NOT NULL,
  `m` int(10) NOT NULL,
  `d` int(10) NOT NULL,
  `Purchase_date` varchar(200) NOT NULL,
  `total` int(100) NOT NULL,
  `balance` int(200) NOT NULL,
  `cate_image` longblob NOT NULL,
  `status` int(2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
-- Table structure for table `curt_order`
--
CREATE TABLE `curt_order` (
  `id` int(200) NOT NULL,
  `order_id` int(100) NOT NULL,
  `category_id` int(200) NOT NULL,
  `user_id` varchar(200) NOT NULL,

```

```

`quantitiy` int(20) NOT NULL,
`price` varchar(50) NOT NULL,
`total_price` varchar(50) NOT NULL,
`vat` varchar(200) NOT NULL,
`orderd_date` varchar(100) NOT NULL,
`y` int(10) NOT NULL,
`m` int(10) NOT NULL,
`d` int(10) NOT NULL,
`sales_date` varchar(100) NOT NULL,
`status` int(20) NOT NULL,
`invoice_number` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Dumping data for table `category`
--
INSERT INTO `category` (`category_id`, `ID`, `category_name`, `buy_price`,
`total_purcase_price`, `price`, `exp_date`, `y`, `m`, `d`, `Purchase_date`, `total`, `balance`,
`cate_image`, `status`) VALUES
--
-- Dumping data for table `curt_order`
--
INSERT INTO `curt_order` (`id`, `order_id`, `category_id`, `user_id`, `quantitiy`, `price`,
`total_price`, `vat`, `orderd_date`, `y`, `m`, `d`, `sales_date`, `status`, `invoice_number`)
VALUES
(33, 99, 10, 'cus-1001', 2, '17.25', '34.5', '4.5', '2021-08-06', 2021, 8, 6, '2021-08-08', 2, 'F-1000'),
CREATE TABLE `customer` (
`ID` int(11) NOT NULL,
`user_id` varchar(200) NOT NULL,
`fname` varchar(10) NOT NULL,
`mname` varchar(50) NOT NULL,
`lname` varchar(50) NOT NULL,

```

```

`city` varchar(50) NOT NULL,
`kebele` varchar(50) NOT NULL,
`email` varchar(30) NOT NULL,
`phone` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
-- Dumping data for table `customer`
--
INSERT INTO `customer` (`ID`, `user_id`, `fname`, `mname`, `lname`, `city`, `kebele`, `email`,
`phone`) VALUES
(15, 'cus-1002', 'godie', 'mawun', 'mawunn', 'Wolkite', '09', '', '0966308312'),
--
-- Table structure for table `employee`
--
CREATE TABLE `employee` (
  `ID` int(200) NOT NULL,
  `user_id` varchar(200) NOT NULL,
  `fname` varchar(50) NOT NULL,
  `mname` varchar(50) NOT NULL,
  `lname` varchar(50) NOT NULL,
  `gender` varchar(50) NOT NULL,
  `email` varchar(30) NOT NULL,
  `bdate` varchar(50) NOT NULL,
  `address` varchar(50) NOT NULL,
  `phone` varchar(50) NOT NULL,
  `role` int(4) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
-- Dumping data for table `employee`
--

```



```

INSERT INTO `employee` (`ID`, `user_id`, `fname`, `mname`, `lname`, `gender`, `email`,
`bdate`, `address`, `phone`, `role`) VALUES
(13, 'emp-1003', 'zelalem', 'addis', 'workie', 'Male', '', '1998-11-21', 'shindy', '0946238305', 2),
--
--Table structure for table `report`
--
CREATE TABLE `report` (
  `id` int(100) NOT NULL,
  `name` varchar(200) NOT NULL,
  `report_name` varchar(200) NOT NULL,
  `size` int(11) NOT NULL,
  `downloads` int(11) NOT NULL,
  `attached_date` varchar(200) NOT NULL,
  `use_attached` varchar(200) NOT NULL,
  `status` int(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
--Dumping data for table `report`
--
INSERT INTO `report` (`id`, `name`, `report_name`, `size`, `downloads`, `attached_date`,
`use_attached`, `status`) VALUES
(2, 'PDF SPMS (10).pdf', 'daily', 22742, 2, '2021-08-20', 'zelalem addis', 1),

```

6.2 Implementation of the Class Diagram

We use EDRAW MAX to implement class diagram of the system and we implemented the class diagram in our project code.

```

<?php
require_once 'init.php';
class cust
{
    public $user;

```

```

public $userid;
public $start;
public $tex;
public $empid;
public $sub;
public $db;
public $dbt;
public $role;
public $old;
public $new;
public $renew;
public $rnew;
public $fname;
public $lname;
public $phone;
public $kebele;
public $city;
public $mname;
public $requestdata;
public function __construct()
{
    $this->db = new DB();
}
public function create( $user,$fname, $mname,$lname, $city,$kebele, $phone ) {
    $send    = "INSERT INTO customer (user_id,fname,mname,lname,city,kebele,phone)
VALUES( '$user','$fname' ,'$mname','$lname','$city','$kebele','$phone')";
    $resultre= $this->db->query($send);
    if ($resultre) {
        return true;
    }
else{

```

```

        return false;
    }
}

public function update($old,$new,$renew,$sempid)
{
    $this->old=sha1($old);
    $this->new=sha1($new);
    $this->rnew=sha1($renew);

    $update = "update user set pass='$this->new',conpass='$this->rnew' WHERE pass='$this->old' AND user_id='$sempid' ";
    $result=$this->db->query($update);
    if ($result) {
        return true;
    }
    else
    {
        return false;
    }
}

public function viewcatagory($requestdata)
{
    $sql="SELECT * FROM category WHERE ID = '$requestdata' AND balance > 0 AND status ='1' ORDER BY category_id DESC ";
    $res=$this->db->query($sql);
    return $res;
}

public function view( )
{
    $qu="SELECT * FROM conformation WHERE status='0'";
    $resultt = $this->db->query($qu);
    return $resultt;
}

public function viewpro()

```

```

{ $sql="SELECT * FROM product ORDER BY ID DESC ";
  $res=$this->db->query($sql);
  $this->db->num_rows($res);
  while ($row=$this->db->fetch_array($res))
  {
    $dep_tit=$row['product_name'];
    $dep_id = $row['ID'];
    echo "<option value='$dep_id'>$dep_tit</option>";
    //echo "<option value='$dep_tit'>$dep_id</option>";
  }
}
}

```

Register employee class implementation

```

<?php
require_once 'init.php';
class addemp
{
  public $requ;
  public $fname;
  public $mname;
  public $lname;
  public $itemid;
  public $gender;
  public $bd;
  public $pb;
  public $address;
  public $city;
  public $phone;
  public $sempid;
  public $old;
  public $new;

```

```

public $renew;
public $count;
public $datee;
public $kebele;
public $role;
public $price;
public $user_id;
public $date;
public $rnew;
public $user;
public $userid;
public function __construct()
{
    $this->db = new DB();
}
public function adddemp($user,$fname, $mname, $lname,$gender,$bd,$pb,$phone)
{
    $sql = "INSERT INTO employee
(user_id,fname,mname,lname,gender,bdate,address,phone,role)
VALUES('$user','$fname','$mname','$lname','$gender','$bd', '$pb','$phone','4')";
    $result=$this->db->query($sql);
    if(!$result)
    {
        return false;
    } else{
        return true;
    }
}
public function update($old,$new,$renew,$sempid)
{
    $this->old=sha1($old);

```

```

$this->new=sha1($new);
$this->rnew=sha1($renew);
$update = "update user set pass='$this->new',compass='$this->rnew' WHERE pass='$this-
>old' AND user_id='$empid'";
$result=$this->db->query($update);
if ($result) {
    return true;
}
else
{
    return false;
}
}
public function editaccount($level, $user_id)
{
    $sql="update user set role='$level' WHERE user_id='$user_id'";
    $result= $this->db->query($sql);
    if($result)
    {
        return true;
    }
    else
    {
        return false;
    }
}
public function updateitem($itemid, $price)
{
    $sql="update category set price ='$price' WHERE category_id='$itemid'";
    $result= $this->db->query($sql);
    if($result)

```

```

    {
        return true;
    }
    else
    {
        return false;
    }
}

public function updateemployee($user_id, $fname,$mname,$lname,$address,$phone,$role)
{
    $sql="update employee set user_id ='$user_id' WHERE user_id='$user_id'";
    $result= $this->db->query($sql);
    if($result)
    {
        return true;
    }
    else {
        return false;
    }
}

public function delete($key )
{
    $SQL = "DELETE FROM employee WHERE user_id = '$key'";
    $res = $this->db->query($SQL);
    if ($res) {
        return true;
    } else {
        return false;
    }
}

public function deleteitem($key )

```

```

{
    $SQL = "DELETE FROM category WHERE category_id = '$key'";
    $res= $this->db->query($SQL);
    if($res)
    {
        return true;
    }
    else{
        return false;
    }
}

public function deleteorder($key )
{
    $SQL = "DELETE FROM curt_order WHERE order_id = '$key'";
    $res= $this->db->query($SQL);
    if($res)
    {
        return true;
    }
    else{
        return false;
    }
}

public function approveorder($key )
{
    $sql="update curt_order set status='2' WHERE order_id='$key' AND status='1'";

    $result= $this->db->query($sql);
    if($result)
    {
        return true;
    }
}

```



```

    }
    else
    {
        return false;
    }
}

public function select()
{
    $sql = "SELECT * FROM user ";
    $result= $this->db->query($sql);
    // TO check that at least one row was returned
    $rowCheck = $this->db->num_rows($result);
    while ($row=$this->db->fetch_array($result))
    {
        $id=$row['role'];
        echo"<option value='$id'>$id</option>";
        //echo "<option value='$dep_tit'>$dep_id</option>";
    }
}

public function status($status)
{
    $select_status="select * from user where user_id='$status'";
    $result = $this->db->query($select_status);
    while($row=$this->db->fetch_array($result))
    {
        $st=$row['status'];

        if($st=='0')
        {
            $status2=1;
        }
    }
}

```

```

        else
        {
            $status2=0;
        }
        $update="update user set status='$status2' where user_id='$status' ";
        $updatee=$this->db->query($update);
        if($updatee)
        {
            header("Location:recreate.php");
        }
        else
        {
            echo mysql_error();
        }
    }
}

public function view()
{ $sql="SELECT * FROM employee ORDER BY user_id asc ";
    $res=$this->db->query($sql);
    return $res;
}

public function viewpro()
{ $sql="SELECT * FROM product ORDER BY ID asc ";
    $res=$this->db->query($sql);
    return $res;
}

public function viewcatagory()
{ $sql="SELECT * FROM category where balance >=0 ORDER BY category_id DESC ";
    $res=$this->db->query($sql);
    return $res;
}

```

```

public function vieworder()
{ $sql="SELECT * FROM curt_order where status ='1' ORDER BY id asc ";
  $res=$this->db->query($sql);
  return $res;
}

```

Manager class of implementation

```

<?php
require_once 'init.php';
class manager
{
    public $fname;
    public $lname;
    public $role;
    public $key;
    public $uom;
    public $datere;
    public $db;
    public $dbt;
    public $old;
    public $new;
    public $renew;
    public $rnew;
    public $level;
    public $user_id;
    public $requ;
    public $woreda;
    public $kebele;
    public $date;
    public $status;
    public $register;
    public $type;
}

```

```

public $pro;
public $sub;
public $procesid;
public function __construct()
{
    $this->db = new DB();
}
public function addpro($pro, $status, $register,$uom)
{
    $send = "INSERT INTO product (product_name ,is_exp,regis_date,uom) VALUES( '$pro'
,$status','$register','$uom')";
    $resultre= $this->db->query($send);
    if ($resultre) {
        return true;
    }
    else{
        return false;
    }
}
public function select()
{
    $sql = "SELECT * FROM product";
    $result= $this->db->query($sql);
    // TO check that at least one row was returned
    $rowCheck =$this->db->num_rows($result);
    while ($row=$this->db->fetch_array($result))
    {
        $product_name=$row['product_name'];
        $pro_id =$row['ID'];
        echo "<option value='$pro_id'>$product_name</option>";
        //echo "<option value='$dep_tit'>$dep_id</option>";
    }
}

```

```

    }
}
public function delete($key)
{
    $SQL = "DELETE FROM logfile WHERE user_id = '$key'";
    $res = $this->db->query($SQL);
    if ($res) {
        return true;
    } else {
        return false;
    }
}
public function update($old,$new,$renew,$sempid)
{
    $this->old=sha1($old);
    $this->new=sha1($new);
    $this->rnew=sha1($renew);
    $update = "update user set pass='$this->new',conpass='$this->rnew' WHERE pass='$this->old' AND user_id='$sempid' ";
    $result=$this->db->query($update);

    if ($result) {
        return true;
    }
    else
    {
        return false;
    }
}
public function view()
{ $sql="SELECT * FROM product ORDER BY ID asc ";

```

```

        $res=$this->db->query($sql);
        return $res;
    }

    public function viewcatagory()
    { $sql="SELECT * FROM category WHERE balance > 0 AND status ='1' ORDER BY
category_id DESC ";
        $res=$this->db->query($sql);
        return $res;
    }

    public function deletepro($key )
    {
        $SQL = "DELETE FROM product WHERE ID = '$key'";
        $res= $this->db->query($SQL);
        if($res)
        {
            return true;
        }
        else{
            return false;
        }
    }

    public function viewinfo($empid)
    {
        $sql = "SELECT * FROM employee where empid = '$empid' ";
        $res = $this->db->query($sql);
        return $res;
    }

    public function updateinfo($phone ,$empid)
    {
        $update = "update employee set phoneno='$phone' WHERE empid='$empid' ";
        $result=$this->db->query($update);
        if ($result) {
            return true;
        }
    }

```

```

    }
    else
    {
        return false;
    }
}
}
?>
}
?>

```

6.3 Configuration of the Application Server

We use Xampp application server because Xampp is simple, lightweight Apache distribution it is extremely easy to create a local web server for testing and deployment purposes. Everything you needed is to set up a web server – server application (Apache), database (Maria DB), and scripting language (PHP). XAMPP works equally well on Linux, Mac, and Windows. Since it is suitable and have the function, we listed above we use Xampp application server.

6.4 Configuration of Application Security

Our system validates all the input by returning error message and suggesting to try again when invalid input occur by using JavaScript in addition to this suggesting please fill out the field when jump the fill space of the field . We implement encryption for user password when the system Manager creates an Employee account or when the user changes their password the system encrypts the password by using sha1 encryption algorithm. The system has three role and the role are defined clearly. We implemented session for each user login page.

6.5 Implementation of User Interface

We implement the user interface by considering the users of our system and easy for use interface of our system is developed using English language. Because the customer can understand simple way by using bootstrap, Ajax, JavaScript and CSS.

6.6 Testing

6.6.1 Test case

Test case 1: Create account	
Test case objective: login into the system	
Test case description: user enters username and password, then click login button. Next to this Client side program contacts with server, server also contacts with the database and database checks for authentication and displays authorized page based on user role.	
Requirements Verified: Yes	
Test Environment: Apache MySQL server must be in running state, Database Should contain appropriate table and link must be established between server and client program	
Test Setup/Pre-Conditions: Apache server should be in running state and User name and Password fields should be filled correctly.	
Actions	Expected Results
The user should enter the correct user name and password to login.	Displays authorized page based on user role.
If user name and password are not filled correctly the system display to fill the user name and password again.	

Table 23 Tast case

6.6.2 Testing Tools and Environment

Test Environment consists of elements that support test execution with software, hardware and network configured. Test environment configuration must mimic the production environment in order to uncover any environment/configuration related issues.

A typical Environmental Configuration for a web-based application is given below: For web server we use Apache server, for Database MySQL, we use widow operating system, and we use Microsoft Edge browser.

6.6.3 Unit Testing

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

In this phase of testing we tried to eliminate faults in procedure and functions point of view by using black box and white box testing.

Tasks that we have performed under this phase

- Prepare unit test plans.
- Identify the unit test objectives.
- Prepare a test case that includes information such as set of test inputs, execution condition and expected output.
- Perform the tests according to our plan
- Analyses the test result

Tested unit	Input specification	Output specification	Environment	Result
Create account script	input is valid	Account is created	Create account form	sucsess
	Input is invalid	Display error message	Create account form	sucsess
Login script	Valid Uname and password	Authorized home page display	Login form	sucsess
	Invalid Uname & password	Error message display	Login form	sucsess

Table 24 Unit tast

6.6.4 System Testing

In this level of testing process, we have examined how the whole subsystems of department selection and student placement System work together to achieve the desired goal (user's requirements of the system). The goals of system testing are to detect faults that can only 95 be exposed by testing the entire integrated system or some major part of it. Generally, under this testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity of the System. But we will more focus only on function validation and performance.

Sample Tests

1. Evaluate the functionality of subsystem after combination of individual sub system weather it works correctly or not.
2. Check the coherence and coupling of each subsystem.
3. Check the overall functionality of the System that achieves the user's requirement.
4. Measure the system boundary which is beyond the goal or not.
5. Measure the weakness and the strength of the system using different metrics.
6. Check the interaction of each subsystem that performs the specified business process.
7. Verify the system completeness-based the requirement of end user.

6.6.5 Acceptance Testing

Our team members participate on the acceptance testing of our system. According to system requirements and other resources (documentation, source code, user manual) test cases are generated to determine (validation and verification) whether the system satisfies users need and expectation to maintain the reliability of our system and also meet the user's requirements.

CHAPTER SEVEN

7 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

Currently, Agape supermarket's day to day business operations such as product selling, product registration, daily item sales report are carried out using a traditional approach or manual based method. Due to this many problem are occurring such as information losing, data accessibility for privileged users, and also repetitive activities as recording same information, updating information/product, adding new product or information is not that easy. By analyzing this problem, the group team proposed, analyze, design, and implement this computerized system for Agape Supermarket. The system that will provide more efficiency, and accuracy than the manual based system. The new web based computerized and management system perform many operations performed. It prevents the loss of paper documents, time, and effort. This system solves many problems such as lose of information form, and data duplication. But now there is a clear and base idea how the system can be developed and integrated, this project can be seen as an initiation for customers or other users who are doing project on the same field.

7.2 Recommendation

While doing this system the team has faced different types of challenges. But by the cooperation of all the group members and the adviser, the team is now able to reach to the final result. All the group members strongly fought these challenges and take the turn to the front. We would like to recommend the next fourth year student of computing and informatics to add some of the extra features add to this System which we could not do due to time limitation and unavoidable reasons and challenges including the following features.

- ➡ Update employee(when the manager register the employee in wrong information)
- ➡ Update category(when the manager create category or insert wrong information of category)
- ➡ Update profile

APPENDICES

Sample source code

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> </title>
  <script>
    <!--
    var ScrollMsg= "Agape Supermarket - "
    var CharacterPosition=0;
    function StartScrolling() {
      document.title=ScrollMsg.substring(CharacterPosition,ScrollMsg.length)+
        ScrollMsg.substring(0, CharacterPosition);
      CharacterPosition++;
      if(CharacterPosition > ScrollMsg.length) CharacterPosition=0;
      window.setTimeout("StartScrolling()",150); }
    StartScrolling();
    // -->
  </script>
  <link rel="icon" type="img/jpg" href="img/cart.gif"/>
  <link href="css/bootstrap.min.css" rel="stylesheet">
  <link href="css/datepicker3.css" rel="stylesheet">
  <link href="css/styles.css" rel="stylesheet">
  <!--Icons-->
  <script src="js/lumino.glyphs.js"></script>
  <!--[if lt IE 9]>
  <script src="js/html5shiv.js"></script>
```

```

<script src="js/respond.min.js"></script>
<![endif]-->
<script type="text/javascript">
    function lock()
    {
        var date=new Date();var weekday=date.getDay();var year=date.getYear();var
month=date.getMonth();var day=date.getDate();var hour=date.getHours();var
minute=date.getMinutes();var second=date.getSeconds();var days=new Array
("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"
);var
months=new
Array("January","February","March","April","May","June","July","August","September","Octo
ber","November","december");var monthname=months[month];var
dayname=days[weekday];var ap;var sal;if(year<1000){year=year+1900} if(hour<12
&&minute<60 &&second<60){sal='Good Morning';} if(hour>=12 &&hour<17 &&minute<60
&&second<60 ){sal='Good Afternoon';} if(hour>=17 &&hour<24 &&minute<60 &&second<60
){sal='Good Evening';} if(hour>=12){hour=hour-12;ap='PM';}else{ap='AM';}
if(minute<10){minute="0"+minute} if(second<10){second="0"+second};
document.getElementById('time').innerHTML=sal;setTimeout("lock()",60000)}window.onload=
function(){lock();}
</script>
<script type="text/javascript">
    function clock(){
//Save the times in variables
        var today = new Date();
        var hours = today.getHours();
        var minutes = today.getMinutes();
        var seconds = today.getSeconds();
//Set the AM or PM time
        if (hours >= 12){
            meridiem = " PM";
        }

```

```

        else {
            meridiem = " AM";
        }
//convert hours to 12 hour format and put 0 in front
        if (hours>12){
            hours = hours - 12;
        }
        else if (hours===0){
            hours = 12;
        }
//Put 0 in front of single digit minutes and seconds
        if (minutes<10){
            minutes = "0" + minutes;
        }
        else {
            minutes = minutes;
        }
        if (seconds<10){
            seconds = "0" + seconds;
        }
        else {
            seconds = seconds;
        }
        document.getElementById("clock").innerHTML = (hours + ":" + minutes + ":" + seconds +
meridiem);
    }
    setInterval('clock()', 1000);

</script>
<style>
    #clock {

```

```

        margin-top: 5%;
        font-size: 25px;
        font-family: Baskerville, Georgia;
    }
</style>
</head>
<body style="background-color: white">
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
    <div class="container-fluid" style="background-color:#178acc;">
        <div class="navbar-header" style="background-color:#178acc; margin-left:auto;margin-
right:auto">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#sidebar-collapse">
                </button>
            <a class="navbar-brand" href="index.php">
                <span style="color:white"><b>Agape Supermarket</b></span></a>
            <ul class="user-menu">
                <li style="float:left; margin-right:400px;padding-left:20px"><a href="#"><i
class="glyphicon glyphicon-globe"></i><b> Supermarket Management System for Agape
Supermarket</b></a></li>
                <li class="dropdown pull-right">
                    <a href="#" class="dropdown-toggle" data-toggle="dropdown"><svg class="glyph
stroked male-user"><use xlink:href="#stroked-male-user"></use></svg> LOGIN <span
class="caret"></span></a>
                    <ul class="dropdown-menu" role="menu">
                        <li><a href="login.php"><svg class="glyph stroked male-user"><use
xlink:href="#stroked-male-user"></use></svg> login</a></li>

                    </ul>
                </li>
            </ul>
        </div>
    </div>

```

```

</div>

</div><!-- /.container-fluid -->
</nav>
<div id="sidebar-collapse" class="col-sm-3 col-lg-2 sidebar" style="background-color:#f5f5f5">
  <form role="search"style="background-image: url(img/maru.png); height: 100px; width: 400px; ">
    <div id="clockbox" class="form-group" >
      <div id="time" style="padding-bottom: 0px;"></div>
<div id="clock"></div>
    </div>
  </form>
  <ul class="nav menu">
    <li class="active" style="font-family:georgia;"><a href="index.php"><svg class="glyph
stroked      dashboard-dial"><use      xlink:href="#stroked-dashboard-dial"></use></svg>
Dashboard</a></li>
    <li><a href="about.php"><span class="glyphicon glyphicon-info-sign"><use
xlink:href="#stroked-calendar"></use></span>About-us </a></li>
    <li><a href="contact.php"><span class="glyphicon glyphicon-phone"><use
xlink:href="#stroked-line-graph"></use></span> Contact</a></li>
    <li><a href="whatisnew.php"><span class="glyphicon glyphicon-eye-open"><use
xlink:href="#stroked-line-graph"></use></span>View Product</a></li>
    <li><a href="custre.php"><span class="glyphicon glyphicon-user"><use
xlink:href="#stroked-line-graph"></use></span>Create Account</a></li>
  </ul>
</div><!--/.sidebar-->
<div class="col-sm-9 col-sm-offset-3 col-lg-10 col-lg-offset-2 main">
  <div class="row" >
    <ol class="breadcrumb">

```



```

        <li class="active" style="padding-left:10%;padding-top:0%"><h2>Web Based
Supermarket Management System for Agape Supermarket<h2></li>
    </ol>
</div>
<div style="padding-bottom:2%; margin-top:0%; padding-right:5%;padding-left:5%;padding-
top:3%" dclass="box-body chat" id="chat-box" >
    <div style="padding-bottom:2%; margin-bottom:3%; padding-right:3%;padding-left:3%"
dclass="box-body chat" id="chat-box" >
        <div id="carousel-example-generic" class="carousel slide" data-ride="carousel">
            <!-- Indicators -->
            <ol class="carousel-indicators">
                <li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>
                <li data-target="#carousel-example-generic" data-slide-to="1"></li>
                <li data-target="#carousel-example-generic" data-slide-to="2"></li>
                <li data-target="#carousel-example-generic" data-slide-to="3"></li>
            </ol>
            <!-- Wrapper for slides -->
            <div class="carousel-inner" role="listbox" >
                <div class="item active">
                    
                    <div class="carousel-caption">
                        </div>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="row" style="margin-top:20%;">
    <ol class="breadcrumb">

```

```
<h10 style="text-align: center;font-family: Georgia;font-size: small;">&copy; <?php  
echo date("Y");?> @AgapeSupermarket.All rights reserved 2021</h10>
```

```
</ol>  
</div><!--/.row-->  
<script src="js/jquery-1.11.1.min.js"></script>  
<script src="js/bootstrap.min.js"></script>  
<script src="js/chart.min.js"></script>  
<script src="js/chart-data.js"></script>  
<script src="js/easypiechart.js"></script>  
<script src="js/easypiechart-data.js"></script>  
<script src="js/bootstrap-datepicker.js"></script>  
</body>  
  
</html>
```

Reference

1. **H.Gao.** *the analyis and design of supermarket management system.* 2012.
2. **Lonnie.D.Bantlely.** *system analysis and design for global enterprise.*
3. **A.Hoffer, Jeffrey.** *modern system analysis and design six edition.*
4. **Booch, Grady.** *object oriented analysis and design with application third edition.*