Food Management System for Debre Markos University Cafeteria

project report

submitted by:

1 Habtamu kebede……………………………………TER_1201_10

2 Melaku wubetu………………………………………TER_1209_10

3 Metasebiya Getaneh………………………………. TER_1211_10

In partial fulfilment of the requirement for the award of degree of

BACHELOR OF SCIENCE DEGREE IN SOFTWARE ENGINEERING

Under the guidance of

Addisu Mesfin (MSC)

____

Advisor's signature

DEBREMARKOS UNIIVERSITY

Institute of technology

SCHOOL OF COMPUTING

ACADAMIC PROGRAM OF SOFTWARE Engineering

Debre Markos Ethiopia

July,2022

Declaration

We hereby to declare that this project is entitled Food management system for Debre Markos university is our own work.it is being submitted in partial fulfilment of the Academic requirements for the bachelor degree in software engineering the result of our project is carried out under supervision of M.R Addisu Mesfin

Candidate’s name signature Date

Habtamu kebede __________ _____________

Melaku wubetu __________ _____________

Metasebiya Getaneh ____________ ____________

This has been submitted for examination with our approval

Mr. Addisu Mesfin _____________ ________________

Advisor signature Date

Examiner ____________ signature ___________ Date___________

Acknowledgement

First of all, we would like to thank God for helping through the overall process this project.

Next, we would like to thank Debre Markos university school of computing for giving us support that would involve students in such kinds of problem-solving solutions.

Our special thanks go to our advisor Mr. Addisu Mesfin for his useful and valuable support, encouragement and hints in every step of the way.

Finally, we would like to thank the stake holders who provide us with necessary information.

Table of Contents

List of tables

List of figures

List of acronyms

GHz	giga hertz
CSS	cascading style sheet
UML	unified modelling language
GB	Gigabyte
HTML	Hypertext markup language
MYSQL	My structured query language
RAM	Random access memory
EID	employee identification
URL	uniform resource location
SQL	structured query language
Stud_id	student identification

Abstract

Debre Markos University student’s food service gives a food service for the students. Addressing of food service for the student’s gets difficult when workers use manual system to conduct the service; student service directorate get student’s information from registrar office using either soft copy or hard copy, and he/she distribute this information for head café, item store manager and other remaining stakeholders. in order to perform what they do for students after head café get student’s information, he/she assign daily student’s food menu based on their numbers and send it to item store manager and item store manager see the ordered item food then food items are send to chef after that item store calculate how much food items are left and how much are left through manual way.

This project emphasizes on the manual system that is described above. After studying this manual system by using different types of methodologies we have developed this project that decrease the workload of workers and increasing information exchange rate by automating some of the activities that are done manually. Generally, this system has major mechanisms to get current student’s information fast, enables item store manager to view bin balance, to register incoming food items and material, and head café to assign student’s daily menu, to register non cafe students, to assign meal card for students, student directorate to register punish students, to scale-out food scale.

CHAPTER ONE

1 INTRODUCTION

Food management system is the process of addressing and managing food for a target user. when we talk about managing food, the main task we take into consideration is the difficulty if the supplier does not know the number of consumers. Because of this reason most organizations might get loss in their services. from those organizations' universities are places that provide food service for large number of students. And the main focus of this project is automating the food management system for Debre Markos University.

1.1 Background of the project

Debre Markos University is one of the governmental higher educational institutions that gives services for the students and the community. Addressing food service for students is one of the services from it serves. To do this task the university build up many divisions of responsibilities, among those: student service directorate, food store, head café, ticker head, chef, student union and other different participants are the stake holders. Even though different participants are assigned at different places, their main goal is addressing food for students. To do this they have to send number of students to federal and the federal funds for food services. After the university gets its budget, the stakeholders plan annual food menu and make a bid to get food materials from outside body. Those suppliers may be merchants or enterprises. After they get the raw materials from the suppliers, they communicate each other to serve food for students. When they do these activities, tasks are done through manual way, which makes it difficult to manage the food service. so, we are initiated to do a web-based food management system for Debre Markos University cafeteria.

1.2 Statement of the problem

Even if the technology is arising from years to years the cafeteria of Debre Markos university addressing system of food for students and its management of food items have some problems and we stated them as follows:

- ✓ The student service directorate receives number of students from registrar in manual way. This is time consuming to get necessary information fast.
- ✓ The head café does not get current status of student's information fast so it is difficult to order daily menu because it might lead to wastage of food.
- ✓ Item store manager gets it hard to make bin balance; all mathematical operation is done using calculators and papers which makes them prone to errors.
- ✓ Daily food menu from the head cafe goes to store manager in manual way
- ✓ It is difficult to know the number of students who left the university's compound due to vacation, internship or withdrawal

- ✓ When the workers in the cafeteria make monthly and annual reports they work in a manual way.

1.3 Objective of the project

1.3.1 General objective

The general objective of this project is developing web-based food management system of students in Debre Markos university cafeteria.

1.3.2 Specific objective

Specific objectives are designed to give answers for the question what advantage will the system provide after it is accomplished; what kind of situation is it going to create for users. Specific objectives of the system are listed as follows:

- ✓ To develop a mechanism that enables registrar office, student service directorate, student food service, head café, item store manager, enterprise, ticker head and the forth to communicate each other online.
- ✓ To make daily food menu of the students using the system.
- ✓ To generate annual and monthly report by the system
- ✓ To make back up of the system and restore it when it is necessary.
- ✓ To give meal card for students.
- ✓ To Create a mechanism for calculating bin balance

1.4 Scope of the project

The scope of this project is concentrated on how to facilitate student’s food management system activities and application of cafeteria in an easy, efficient and functional way. The following lists are the scope of the project:

- ✓ Registering and displaying detail information about the employees of cafeteria and the students
- ✓ Allow employees of the cafeteria and the administrator to login with their own password
- ✓ Keep students and cafeteria file secured
- ✓ Information sharing among employees
- ✓ Update record of correctional program attained by employees.
- ✓ Update record of correctional program attained by students

1.5 significance of the project

The proposed system delivers different function for the university:

- ✓ Reduce work load of student’s food service workers like head café, head ticker, item store manager

- ✓ Since the number of students who are going for internship or trip is known by the system as soon as possible, it will reduce the wastage of food.
- ✓ Reduce the time needed to accomplish some tasks for example registration of students for food service
- ✓ Faster request response between stake holders
- ✓ The possibility of fraud on universities property will be reduced since everything found in the item store will be registered on the system
- ✓ It is profitable according to saving properties like paper and pen since the system is going to replace the paper work.

1.6 Tools and methodology

1.6.1 Data collection methodology

Throughout the work of this project, we used different methods for collecting information and as well as gathering data those are:

- ✓ Interview: we went to the university's different stake holders specially those who work around the cafeteria and interviewed them.
- ✓ Discussion: this is one of the techniques in which we have seated together and discussed on the project how we can perform the system.
- ✓ Observation: assessing and analysing the overall system has been carried out by observing the current working system we have gone to DMU cafeteria office and seen how their system works and we have conducted physical observation how data are Handled and information of the system kept in the system
- ✓ Referring documents: we have collected some documents through the internet.

1.6.2 Programming language to be used

We will use the following programming language in our system

Back-end design method will be:

PHP- for server-side scripting.

MYSQL (XAMPP) -used to as a database tool for our system.

Front end design method:

The user interface will be developed using HTML, CSS and Java script.

HTML- used to create a web page for the system.

CSS- we will use CSS to apply a set of style characteristics to our system

JavaScript- we will use java script for client-side scripting such as form validation and different pop-up messages.

Since we are going to use the above programming language, we will have a cost advantage. The other reason is they are license free and there is no royalty fee involved.

1.6.3 System requirements

There are hardware and software requirements of the system.

The main Hardware requirement we need is computer, which is used to develop our web application on food management system and the minimum computer hardware specifications are:

- ✓ Processor- core i3 @2Ghz
- ✓ Ram-4GB
- ✓ Hard drive- 200 GB
- ✓ Operating system-windows 10

Other hardware requirements beside computer are:

- ✓ Backup device- USB flash disk
- ✓ Other materials- paper and pen

The software requirements are as follows:

- XAMPP server: we use XAMPP server to manage our server settings.

XAMPP server is an open-source software which is used to run and test application before being uploaded on the actual server.

The advantage of XAMPP software is, it is easily configurable with the built-in tools.

- Visual studio code (vs code) - we use visual studio code text editor to write our code.
- Microsoft word 2019- we use Microsoft word 2019 to write the documentation about the proposed system from the beginning up to the end of the project.
- Microsoft power point 2019- we will use Microsoft power point for presentation purpose.
- Browsers
  - Mozilla Firefox
  - Google chrome
  - 0Brave browser

1.6.4 System modelling tools

- Draw.io- we used draw.io software to draw different uniform modelling language (UML) which is necessary to structure the system.

1.7 Feasibility study

A feasibility is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and [1].

- ✓ A feasibility study should management with enough information to decide
- ✓ Whether the project can be done
- ✓ Whether the intended project will benefit its intended users and organization
- ✓ What are alternatives among the solutions chosen.

1.7.1 Technical feasibility

At the implementation stage we should use the latest technology tools. And we should check if the technology exists or not. Or if it exists within given resource constraints.

During testing technical feasibility should consider the following essential questions:

- Does the technology exist?
- Is it possible within given resource constraints?
- Are the current technical resources sufficient for the new system?

1.7.2 Operational feasibility

Operational feasibility is the measure of how well our proposed system solves the existing manual systems problem. After automating the system, it addresses basic problems of the organization, particularly students' information can be handled easily, which reduces the workload of the stuff.

The system will fit the system and user requirement with regard to development schedule, delivery date, and existing business process due to our system will be operationally feasible.

1.7.3 Economic feasibility

Economic feasibility is determined by identifying by costs and benefits associated with the system, assigning values to them, calculating future cash flows and measuring the financial value of the project. Since we are going to develop web-based system, information will be gained fast, and it saves time and cost and facilitates work process for the university.

1.7.4 Legal feasibility

The legal feasibility determines whether the system conflicts with the legal requirement of the institution or not. We will build a system that will not conflict with the rules and regulations of Debre Markos University, rather it gives benefit to the university by making part of its work easy.

CHAPTER 2

2 System analysis

System analysis is the process by which individual studies a system such that an information system can be analysed, modelled, and logical alternative can be chosen.

2.1 Overview of the existing system

Now a day's most of the student's food services is performed through manual system. The current food management system in Debre Markos University looks like as described as follows:

The university accepts budget from federal according to the number of students. Then the student's food directorate make annual plan of student's food menu based on the budget.

Student's daily food consumption is scale up by considering fifteen birr per day for each student. The incoming materials from suppliers placed at food store and the registration of the items held on paper. When new materials are imported the student service directorate, student union, head café, store worker and merchant have to check the materials. Then the café head order food items based on daily menu to the store manager. After the store manager receives report from the head café, he/she gives the item based on the daily menu received from the head café. After they provide the necessary items to the cafeteria the store manager then calculates how much he/she gives and how much left in the store.

Head café and student union are responsible for checking the quality of items received from the merchant and provide it to chef. If the chef feels there is a problem with the items, he/she reports it to the head café. And the head café reports the situation to the store manager and the store manager deals with the suppliers and tell them to provide quality materials for future or even might return materials.

2.2 System requirement specification

System requirement specification is a document that describes a nature of a project, software or application. System requirement specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should [2]

2.3 Functional requirements

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it is important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behaviour under [3]

Under functional requirements there are user requirements and system requirements.

User requirements are requirements in which the system must fulfill for users. Some of these are listed below:

- ✓ The system should accept user name and password of a stakeholder.
- ✓ The system should give an exclusive privilege to the administrator to register eligible institutions' users.
- ✓ There system shall validate and authenticate the user's user name and password.
- ✓ The system shall allow item store manager to check bin balance, to register incoming items, to view remaining items and to replay to head café for daily menu request.
- ✓ The system shall allow the student service directorate to register students and to punish them and to report different situations to head café.
- ✓ The system shall allow the head café to give meal card, to assign students daily food menu, to send requests to the item store manager, to view students' current information, to view registered food and to make report.
- ✓ The system shall allow the president or vice president to view annual report.

System requirements are statements that identify the functionality that is needed by a system in order to satisfy customer's requirements. System requirements are a broad and also narrow subject that could be implemented to many items.

Some of the system requirements of our project are listed below: System requirements are the most effective way of meeting the users' needs and reducing the cost of implementation.

- ✓ The minimum requirement that the system needs is operating system should be windows ten.
- ✓ The users of the system should be a little more familiar with computer to use the system properly.
- ✓ Users should login to the system using their username and password only
- ✓ Users shall update their account.
- ✓ Users should be members of Debre Markos University.
- ✓ Users should insert valid data.
- ✓ After using the system, the user should logout from the system.

2.4 Non-functional requirements

Non-functional requirements define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

Non-functional requirements of the system are as described as below:

- ✓ **Performance:** web-based student food management system allows users of the system to access, send and calculate necessary tasks with a high speed.
- ✓ **Reliability:** since the system needs to have backup servers it shall be accessible at any time with the exception of infrastructure failure. This requirement shall be provided for and by Debre Markos university technology systems.
- ✓ **Security:** since the system provides security for database an authorized users should not access the systems database.
- ✓ **Integrity:** Debre Markos university food management system is only accessible by an authorized user of the system to update, delete, modify or access the system.
- ✓ **User access:** The system provides easy user interface for users by avoiding complex buttons and links. And it works in user understandable format.
- ✓ **Reusability:** The system shall be well documented in order for new administrators to change content as needed. Also, the system shall be designed in such a way that the administrators may modify content without the need to modify the code.
- ✓ **Compatibility:** The system shall be compatible with different internet browser like internet explorer, Mozilla Firefox, Brave and google chrome.
- ✓ **Resource utilization:** The system shall be accessible from any time of computer with an active internet connection, adequate hard drive and memory.
- ✓ **Authentication and authorization:** The system shall authenticate the user and provide authorization facilities by asking username and password to check whether the user is authorized to access resource or to perform operation.
- ✓ **Availability:** The system should be available 24/7.
- ✓ **Maintainability:** there should be a backup server for the system when failure happens and when it is needed to be recovered. The system should be easily maintainable by administrator easily by using user interface and also the system can be maintained by eligible programmers when it is needed.
- ✓ **Easy to use:** The system is simple to use and can be easily learned.

Generally,

- ✓ The system has well organized information storage and mechanism
- ✓ The language used to operate the system is English which is internationally understandable language and which is able to increase readability.
- ✓ The system provides quick and easy and quick information analysis which maximize work efficiency.
- ✓ The system is able to manage all incoming information from the data.

2.5 overview of the proposed system

The system is proposed to computerize the working environment of existing system and to overcome the problem regarded with manual work and to benefit the community of cafeteria in common.

The following lists are overview of the proposed system:

- ✓ Saving time which is spent on searching records by hand.
- ✓ Reducing space that is occupied by manual document.
- ✓ Helping the administrator to have any easy access and control over the system.
- ✓ Generating report when it is needed by the stake holders.
- ✓ Online feedback between the community of the cafeteria.
- ✓ Registering, searching, updating and keep all the file in safe way.

2.6 Business rule identification

A business rule is an organizational entity involved in the delivery of goods and services to consumers.

Business rule describes system policy and procedure. They are usually expressed at atomic level. That is, they cannot be broken down any further.

Business rules of food management system for Debre Markos cafeteria are listed below.

Name: -determine quality of materials and foods

ID: -BR-1:

Description: -this business rule states that supplier’s goods and materials are received if and only if and only if the quality of goods and materials have approved by the curators.

Name: -determine daily food menu change

ID: -BR-2

Description: -This business rule describes that daily food menu will change if and only if there is shortage of supply for that specific day specially interruption of electricity.

Name: -determine price of good and materials

ID: -BR-3

Description: -This business rule describes that the price of goods and materials should be constant till the contract has finished.

Name: -determine suppliers

ID: -BR-4

Description: -This business rule describes that supplier who supply goods and materials with minimum price are selected.

Name: -determine daily expenditure

ID: -BR-5

Description: -This business rule describes that the daily food expenditure is determined by number of students.

Name: -punishment of undisciplined workers

ID: -BR-6

Description: -This business rule describes that a person who break the rule and regulation of Debre Markos University Student food service system will punish.

Name: -determine replace meal card

ID: -BR-7

Description: -This business rule describes that cafeteria user students are on the system and when they lose their meal card they will be charged more

Name: -determine validating daily expenditure

ID: -BR-8

Description: -This business rule describes that the daily food expenditure is leave from Item store manager if and only if student union, Item store manager, and head café put their signature at the paper.

Name: -determine assigning meal card

ID: -BR-9

Description: -This business rule describes that student who are cafeteria users allowed to have only one meal card

2.7 System requirement analysis

System requirement analysis is the process of obtaining a through and detailed understanding of business need as defined in project origination and captured in a business case and to break it down in to discrete requirements which are then clearly defined, reviewed and agreed up on the customer decision.

2.8 Actors and use case identification

Actor: in a use case diagram is an entity that performs role in one given system. An actor could be a person, an organization or an external system that can be usually drawn as a skeleton. Actors of the system are:

- ✓ Web admin
- ✓ Student service directorate
- ✓ Department Head
- ✓ Student union
- ✓ Registrar
- ✓ President
- ✓ Vice-president
- ✓ Merchant
- ✓ Enterprise
- ✓ Chef
- ✓ Ticker head
- ✓ Student
- ✓ Head cafe
- ✓ Item store manager
- ✓ Nurse
- ✓ Proctor
- ✓ Purchase and procurement
- ✓ Finance

Use case: represents a function or an action within the system it is drawn oval and named with function. Our use cases are listed below:

ID	Use Case Name	Include
UC1	Login	
UC2	Register incoming food	UC1
UC3	View remaining food item	UC1
UC4	View daily menu	UC1
UC5	Register incoming materials	UC1
UC6	View Request	UC1
UC7	Register non café students	UC1
UC8	Assign daily menu	UC1
UC9	Make Report	UC1
UC10	Generate meal card	UC1
UC11	Update internship request	UC1
UC12	Request material lack	UC1
UC13	Assign special food	UC1
UC14	Upload students list	UC1
UC15	Request to finance	UC1
UC16	View report	UC1
UC17	Request material lack	UC1
UC18	Pay for non-cafe	UC1
UC19	Register Special food user	UC1
UC20	Check daily food menu	UC1
UC21	Register students	UC1
UC22	Approve finance request	UC1
UC23	Post notice	UC1
UC24	Register leave students	UC1
UC25	Register punished students	UC1
UC26	Check out food scale	UC1
UC27	Internship request	UC1
UC28	View log event	UC1
UC29	Use backup	UC1
UC30	Create account	UC1

UC31	Manage profiles	UC1
UC32	Send feedback	UC1
UC33	Register irregular food user	UC1
UC34	View shortage of item	UC1
UC35	View daily eaten food	UC1
UC36	Approve material lack request	UC1
UC37	View material lack request	UC1
UC38	View import item	UC1
UC39	Update student status	UC1
UC40	Reject request	UC1
UC41	Accept request	UC1
UC42	View feedback	UC1

Table 2.1 use case table

2.9 use case diagram

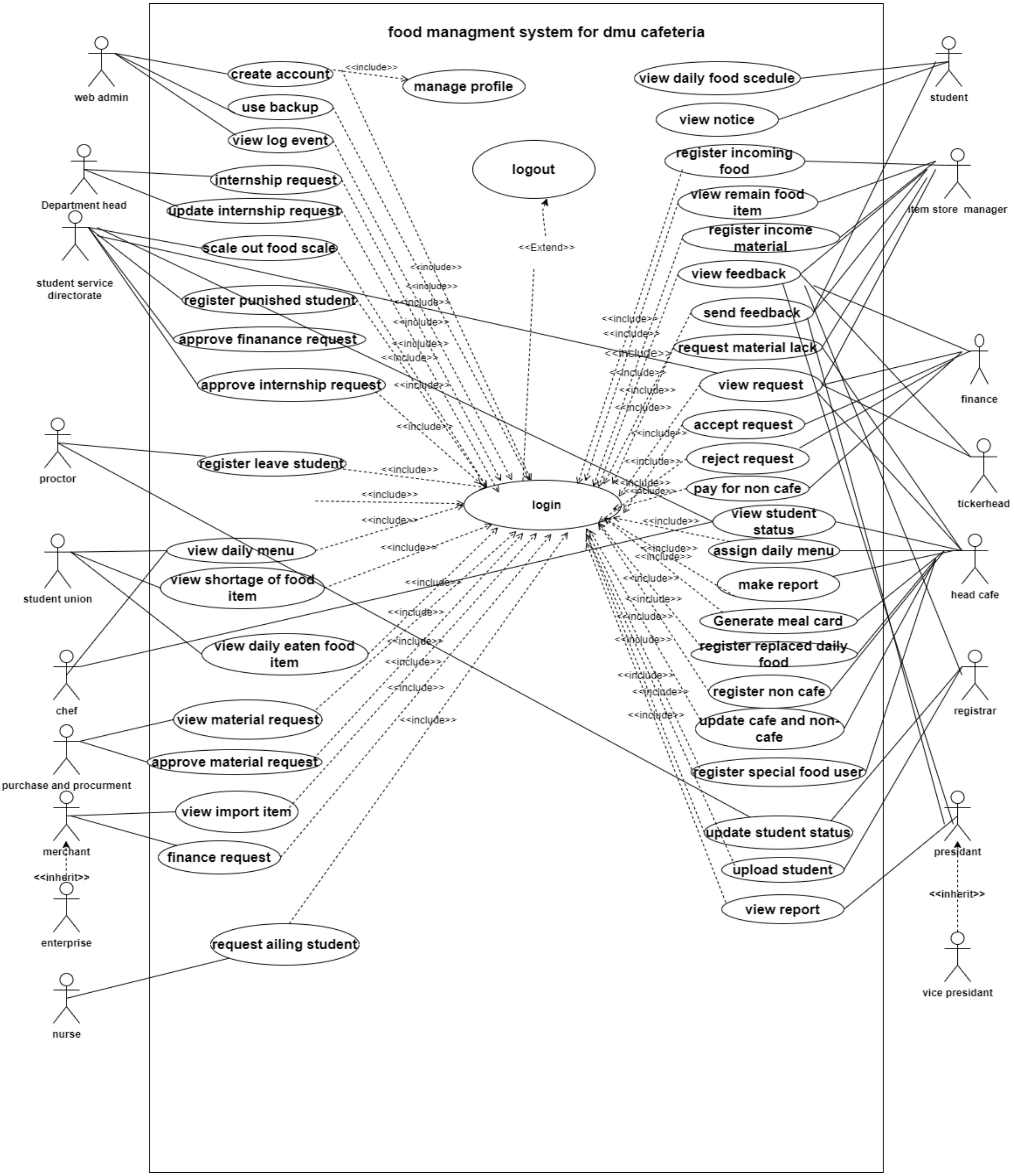


Figure2. 1 Use case diagram

2.10 Use case diagram description

Use case	View request
ID	UC6
Include	UC1
Actor	ticker head, student service directorate, finance, item store manager
Description	All the above listed actors view request to do their specified task
Precondition	Login to the system using his/her username and password
Post condition	Get full information
Basic course of action	1. Look at the site on the address bar of the browser. 2. Enter user name and password 3. Click on view request tab on the menu and select message from stakeholder. 4. End of the use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.

Table 2.2 use case description for view request

Use case	Register non café student
ID	UC7
Actor	Head cafe
Description	Head café register non café students.
Precondition	Login to the system using his/her username and password
Post condition	Non café students are registered.
Basic course of action	1. Look at the site on the address bar of the browser. 2. Enter user name and password 3. Click on Register non café students ‘tab on the menu. 4. Fill all the information at the form and click submit. 5. The system displays success message 6. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.3 register non-café students

Use case	Assign daily menu
ID	UC8
Actor	Head cafe
Description	Head café assign daily food menu based on the calendar
Precondition	Login to the system using his/her username and password
Post condition	Daily foods are assigned.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on Assign daily menu tab on the menu. 4. Select the day, food's good type and their amount. 5. Click submit button. 6. The system displays success message 7. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.4 Assign daily food menu

Use case	Give meal card
ID	UC10
Actor	Head cafe
Description	Head café register students for meal card
Precondition	Login to the system using his/her username and password
Post condition	Students get meal card.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on give meal card tab on the menu. 4. Fill all the required information and click submit button. 5. The system displays success message 6. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.5 Give meal card

Use case	Request materials lack
ID	UC12
Actor	Head café
Description	Head café request material lacks for food store.
Precondition	Login to the system using his/her username and password
Post condition	Material's request is sent.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on request material lack tab on the menu. 4. Fill all the required information and click submit button. 5. The system displays success message 6. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.6 request materials lack

Use case	Assign special food
ID	UC13
Actor	Head café
Description	Head café assign special food for students who have a special case on his or her health.
Precondition	Login to the system using his/her username and password
Post condition	Special foods are assigned.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on assign special food tab on the menu. 4. Fill all the required information and click submit button. 5. The system displays success message 6. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.7 assign special food

Use case	Request to finance.
ID	UC15
Actor	Merchant, enterprise
Description	Merchant and enterprise request to finance to pay for the goods which they import.
Precondition	Login to the system using his/her username and password
Post condition	Requests are sent.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on Request to finance tab from the menu. 4. Fill all the required information and submit. 5. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.8 Request to finance

Use case	View report
ID	UC16
Actor	President, vice-president, student's service directorate
Description	President and vice-president view annual report; student's service directorate view monthly and annual report send from head café.
Precondition	Login to the system using his/her username and password
Post condition	Reports are viewed.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on view reports. 4. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.

Table 2.9 view report

Use case	Request ailing student
ID	UC17
Actor	Nurse
Description	The nurse checks the student health status and then attach the information to the head cafe.
Precondition	Login to the system using his/her username and password
Post condition	Get message as he Send successful information to the head café.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on request ailment tab on the menu. 4. Fill full information on the form and click submit button. 5. The system displays success message 6. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.10 request ailing students

Use case	Register special food
ID	UC19
Actor	Student union
Description	Student unions register students who do not eat regular food .
Precondition	Login to the system using his/her username and password
Post condition	Students are registered.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on register special food tab on the menu. 4. Fill all the required information and click submit button. 5. The system displays success message 6. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.11 register special food

Use case	Register punished students
ID	UC25
Actor	Student service directorate
Description	Student service directorate announce students' who is dismiss from the university increase of discipline.
Precondition	Login to the system using his/her username and password
Post condition	Get message as he Send successful information to the head café.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on request ailment tab on the menu. 4. Fill full information on the form and click submit button. 5. The system displays success message 6. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.12 register punished students

Use case	Internship Request
ID	UC27
Actor	department
Description	department announces number of students those who are gone to internship to the student service directorate.
Precondition	Login to the system using his/her username and password
Post condition	Get successful message that shows weather he was sent correct student information or not.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on send request button 4. Fill all the required information on the form then send to student service directorate. 5. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.13 internship request

Use case	Create account
ID	UC30
Actor	Web admin
Description	Web admin create account for an eligible user of the system
Precondition	Login to the system using his/her username and password
Post condition	Users get account.
Basic course of action	1. Look the site on the address bar of the browser. 2. Enter user name and password 3. Click on create account button 4. Fill all the required information on the form then click submit it. 5. End of use case.
Alternative course of action	A: If login detail is incomplete or mismatch at step 2.
	A1: the system displays error message. A2: actors must look and fix those errors. A3: The use case continues at step 2.
	B: if the information is incomplete at step 4.
	B1: the system displays error message. B2: actors must look and fix those errors. B3: The use case continues at step 4.

Table 2.14 create account

2.11 Sequence diagram

It is an interaction diagram that details how operations are carried out: what messages are sent and when. Sequence diagrams are organized according to time. The time progress as we go down to the [4].

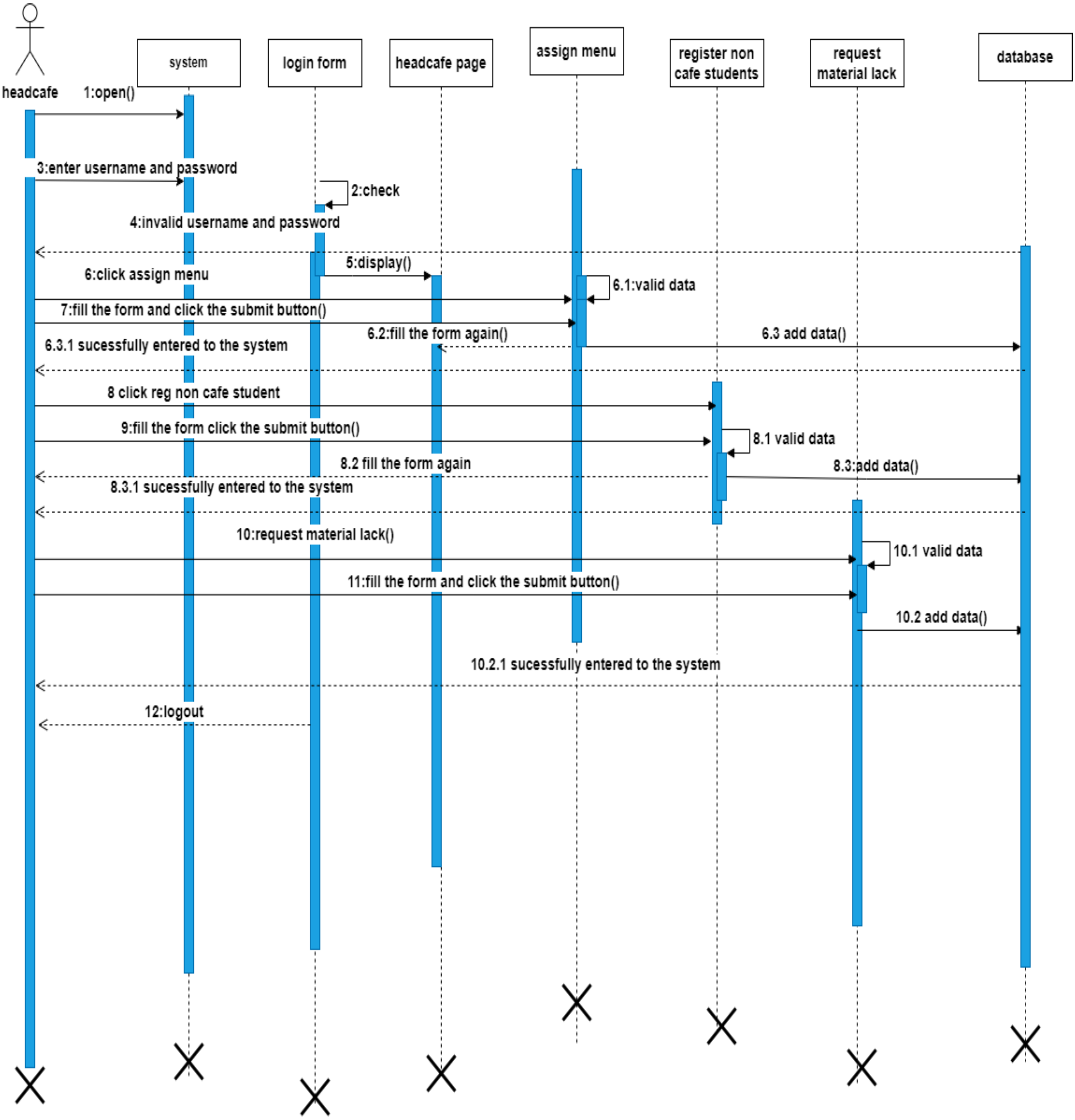

headcafe
system
login form
headcafe page
assign menu
register non cafe students
request material lack
database
1:open()
3:enter username and password
2:check
4:invalid username and password
5:display()
6:click assign menu
6.1:valid data
7:fill the form and click the submit button()
6.2:fill the form again()
6.3 add data()
6.3.1 sucessfully entered to the system
8 click reg non cafe student
9:fill the form click the submit button()
8.1 valid data
8.2 fill the form again
8.3:add data()
8.3.1 sucessfully entered to the system
10:request material lack()
10.1 valid data
11:fill the form and click the submit button()
10.2 add data()
10.2.1 sucessfully entered to the system
12:logout

Figure2. 2 Sequence diagram for head cafe

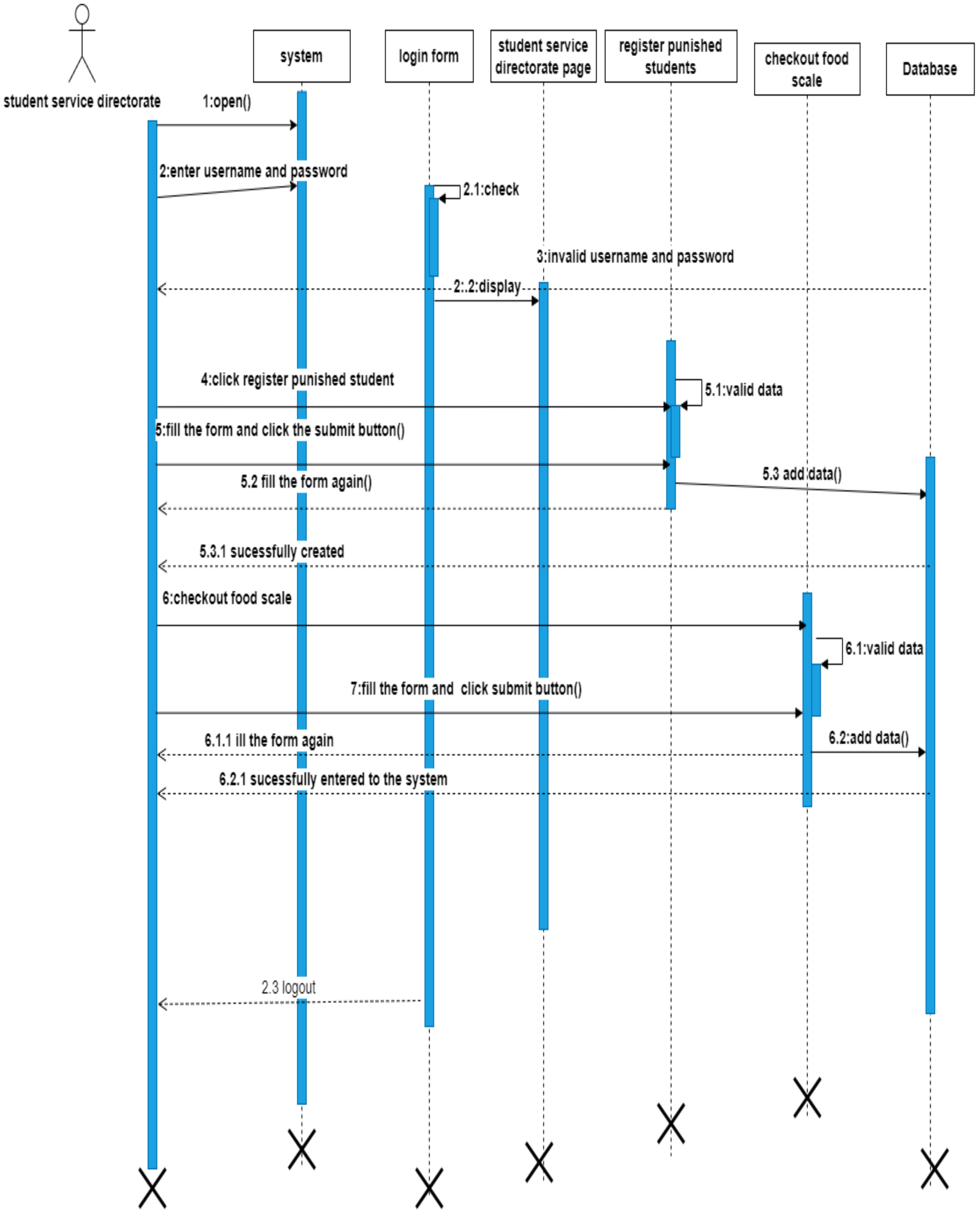

student service directorate
system
login form
student service directorate page
register punished students
checkout food scale
Database
1:open()
2:enter username and password
2.1:check
3:invalid username and password
2:.2:display
4:click register punished student
5.1:valid data
5:fill the form and click the submit button()
5.2 fill the form again()
5.3 add data()
5.3.1 sucessfully created
6:checkout food scale
6.1:valid data
7:fill the form and click submit button()
6.1.1 ill the form again
6.2:add data()
6.2.1 sucessfully entered to the system
2.3 logout

Figure2. 3 sequence diagram for student service directorate

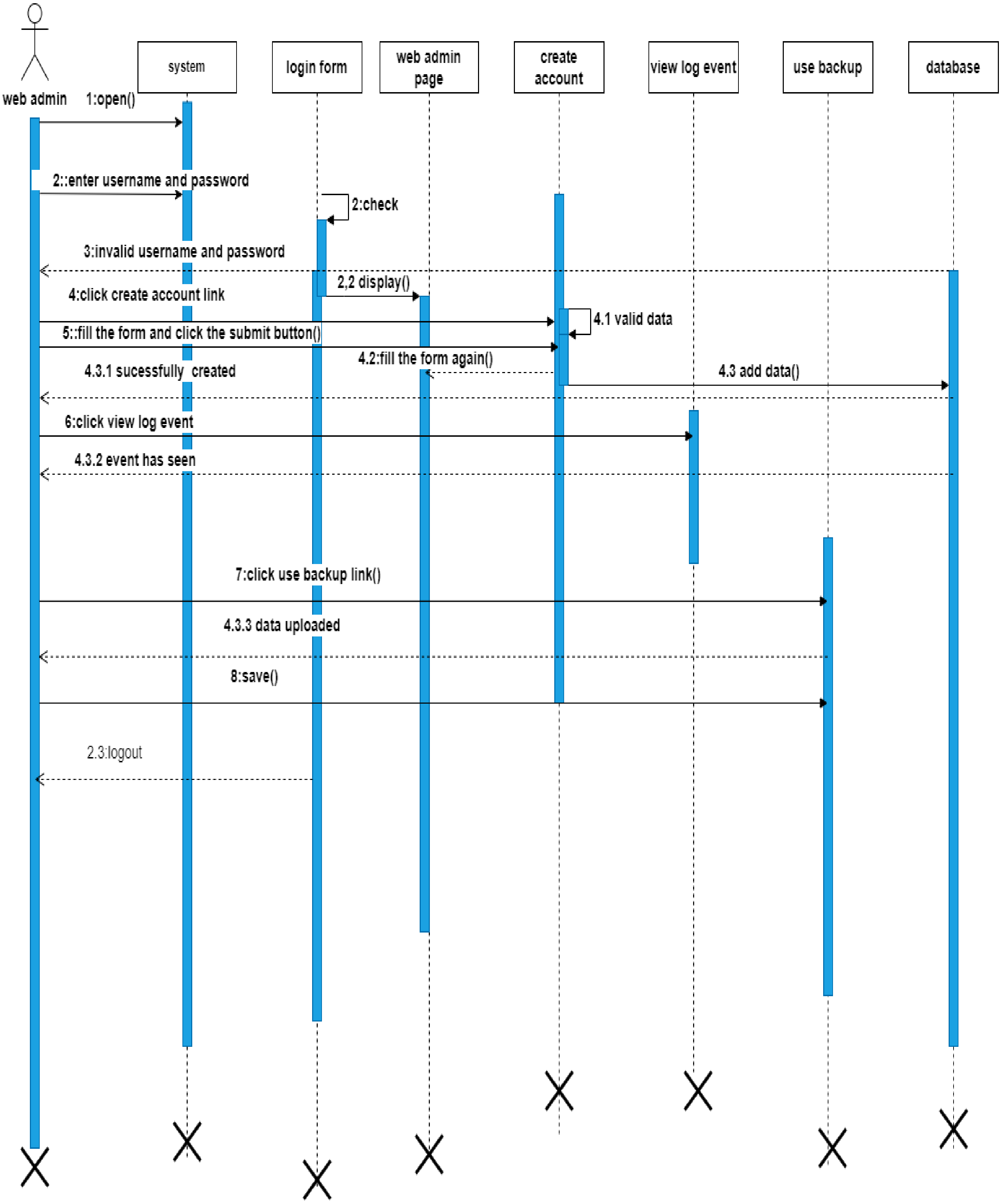


Figure2. 4 Sequence diagram for web Admin

2.12 Activity diagram

An activity diagram is a behavioural diagram that depicts the behaviour of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is [5]

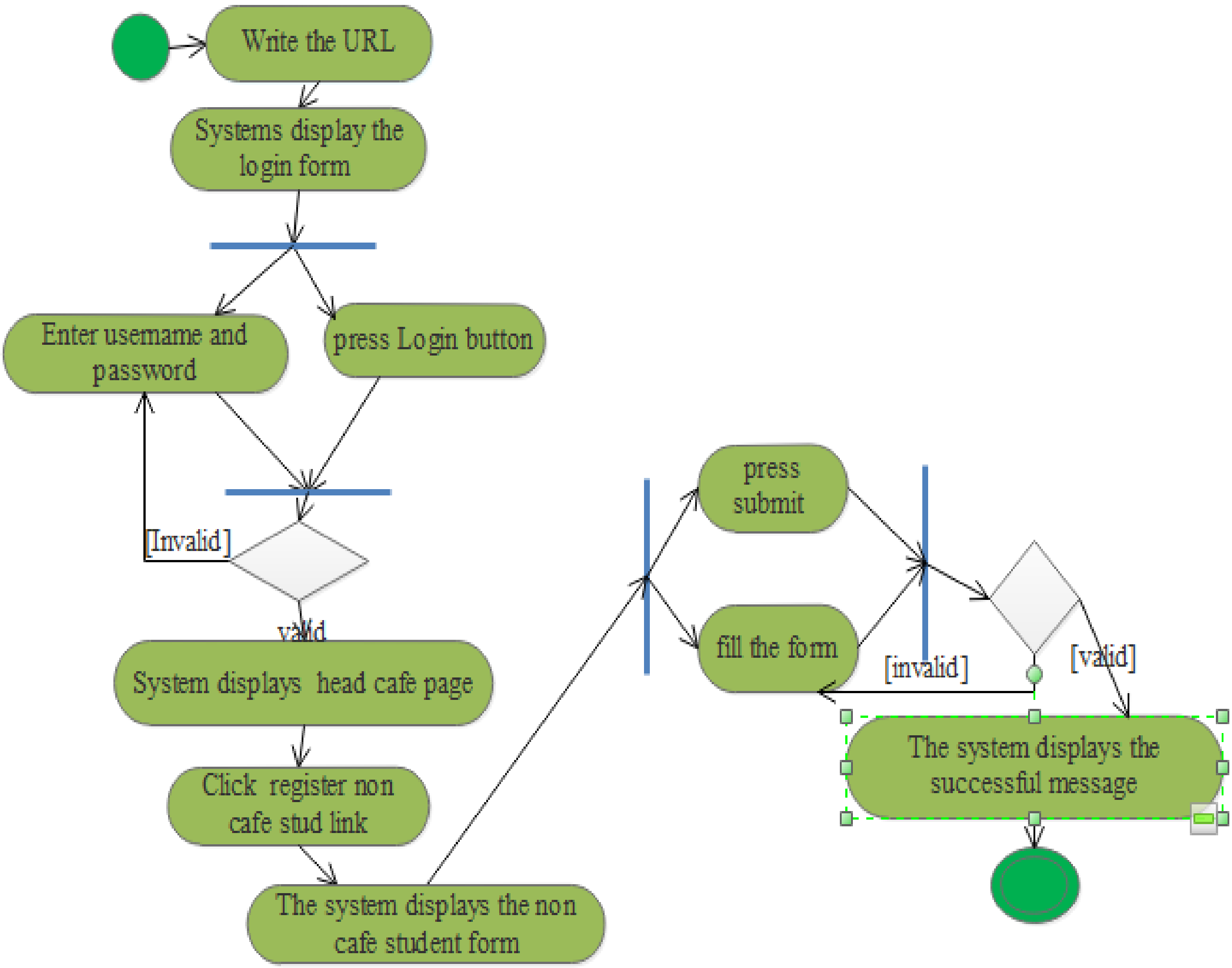
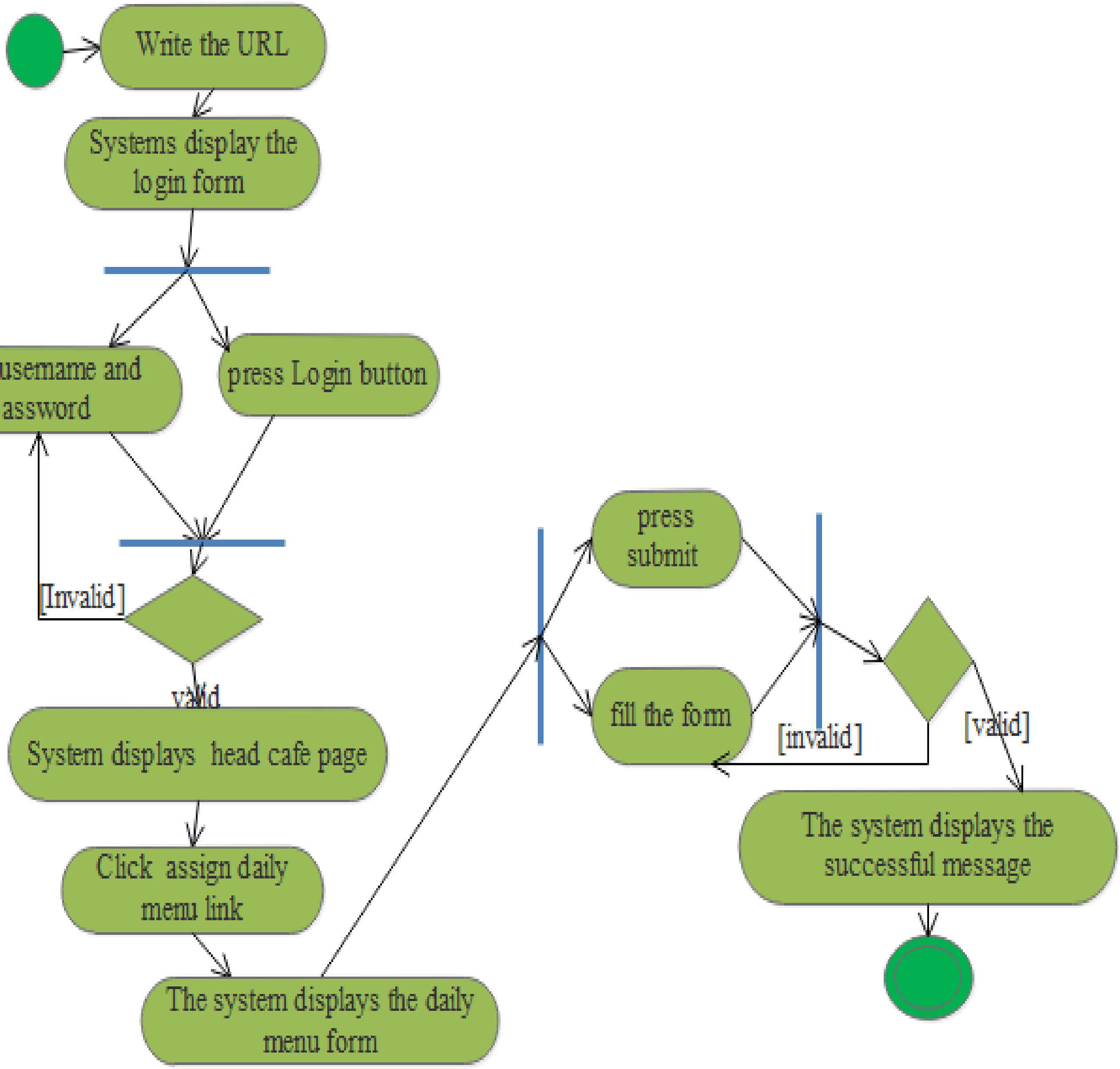

Write the URL
Systems display the login form
Enter username and password
press Login button
[Invalid]
valid
System displays head cafe page
Click register non cafe stud link
The system displays the non cafe student form
press submit
fill the form
[invalid]
[valid]
The system displays the successful message

Figure2. 5 activity diagram for register non-café

Figure2. 6 activity diagram for register non-café


Write the URL
Systems display the login form
Enter username and password
press Login button
[Invalid]
valid
System displays head cafe page
Click assign daily menu link
The system displays the daily menu form
press submit
fill the form
[invalid]
[valid]
The system displays the successful message

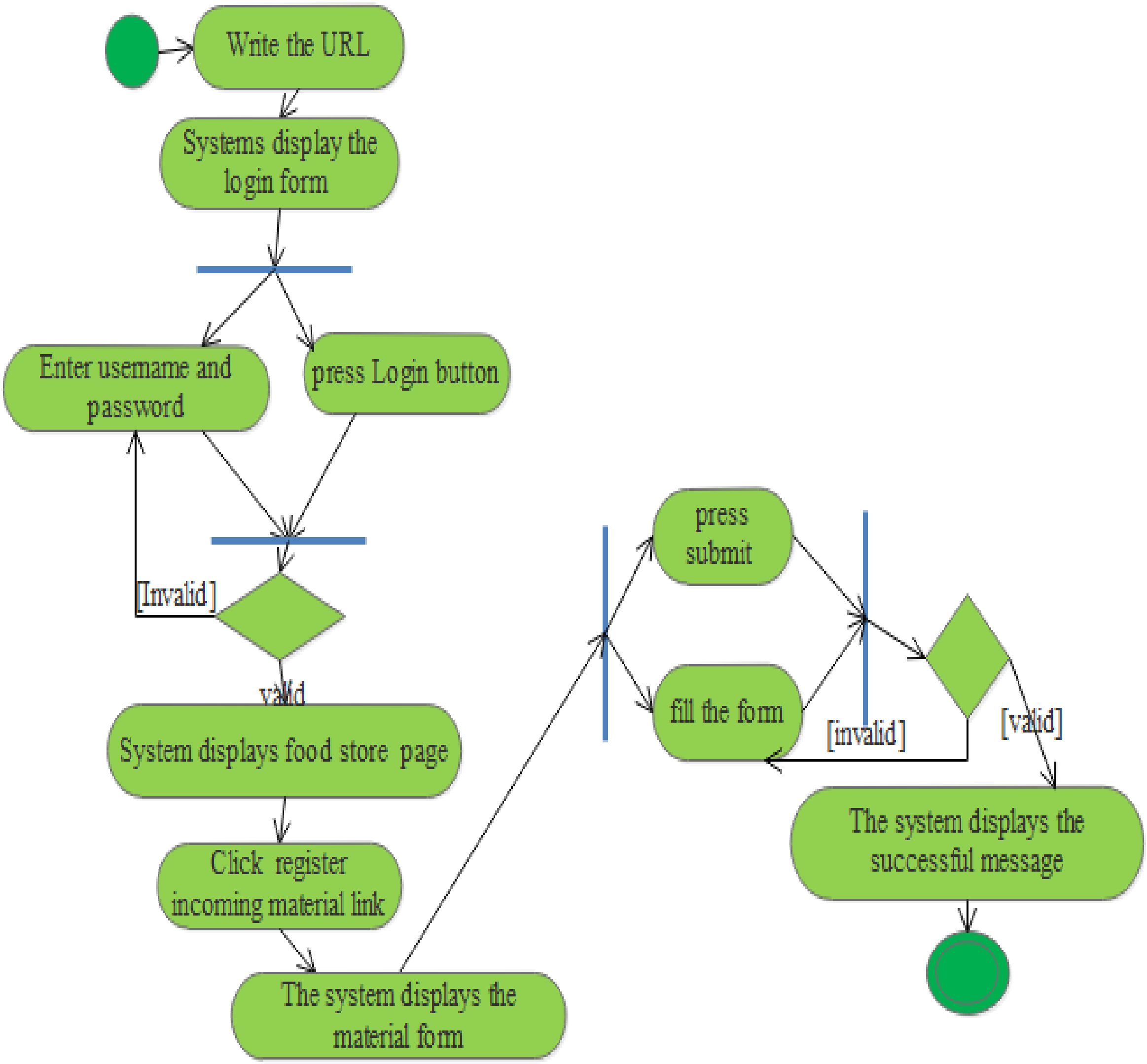

Write the URL
Systems display the login form
Enter username and password
press Login button
[Invalid]
valid
System displays food store page
Click register incoming material link
The system displays the material form
press submit
fill the form
[invalid]
[valid]
The system displays the successful message

Figure2. 7 activity diagram for register incoming materials

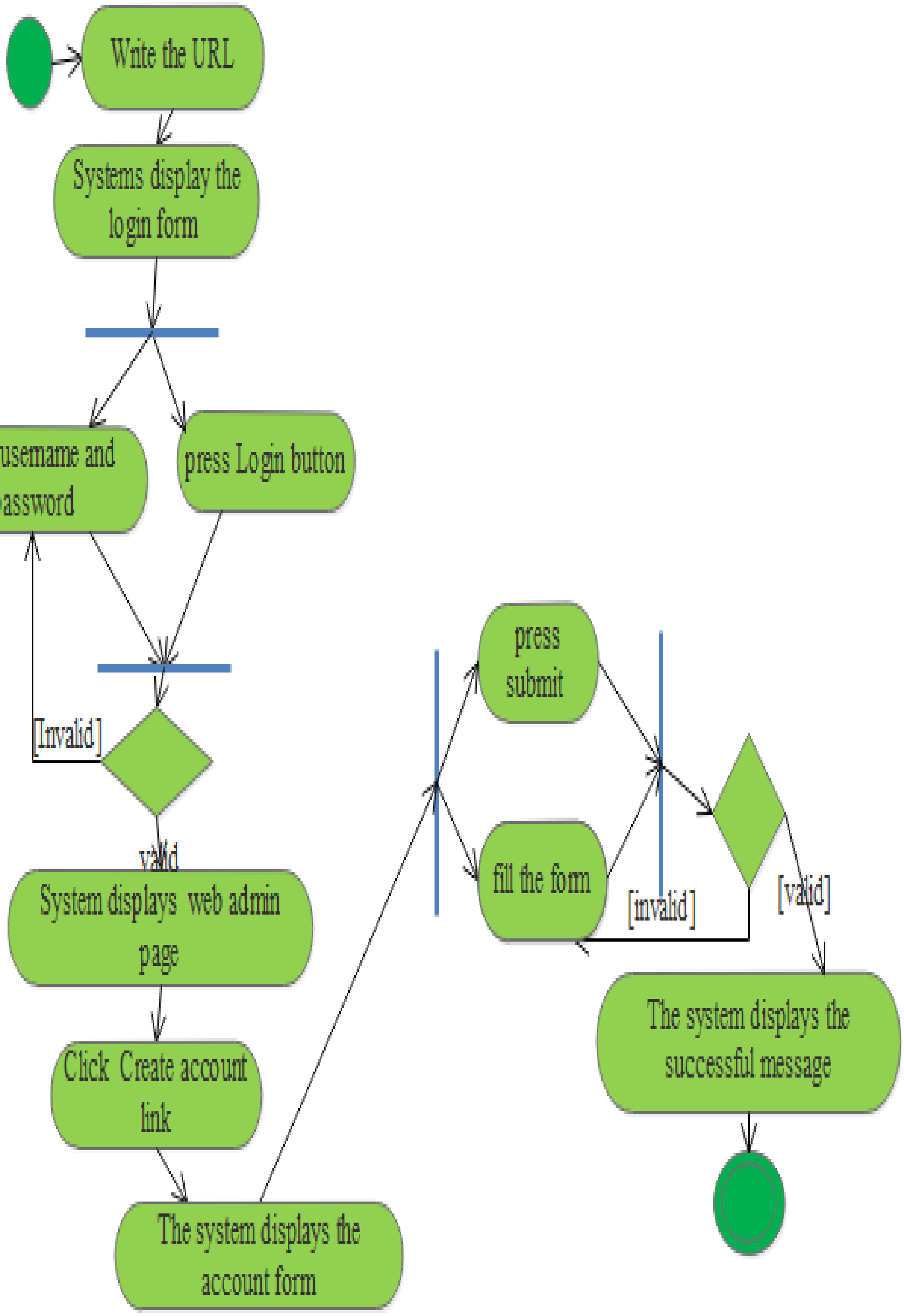

Write the URL
Systems display the login form
Enter username and password
press Login button
[Invalid]
valid
System displays web admin page
Click Create account link
The system displays the account form
press submit
fill the form
[invalid]
[valid]
The system displays the successful message

Figure2. 8 Activity diagram for create account

2.13 Analysis class diagram

Class diagrams are useful in many stages of the system design. In the analysis stage, a class diagram can help to understand the requirements of a problem domain and to identify its component.

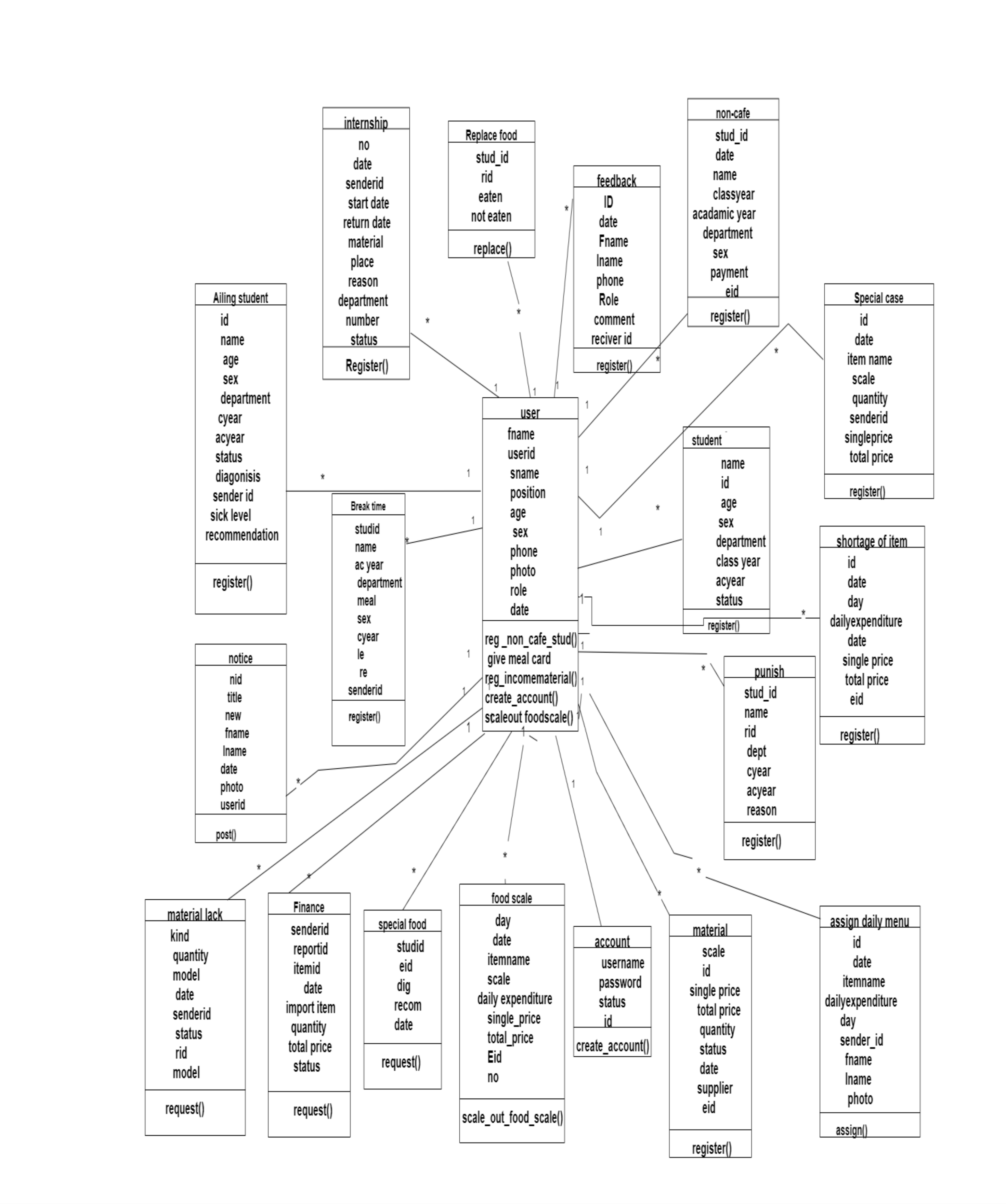


Figure 2. 9 Analysis class Diagram

2.14 State machine diagram

A state machine diagram typically is used to describe a state dependent behaviour of an object. State machine diagrams usually applied to objects but can be applied to any element that has behaviour such as: actors, use cases, methods, subsystems systems and etc. and they are typically used in conjunction with interaction diagrams (usually sequence diagrams).

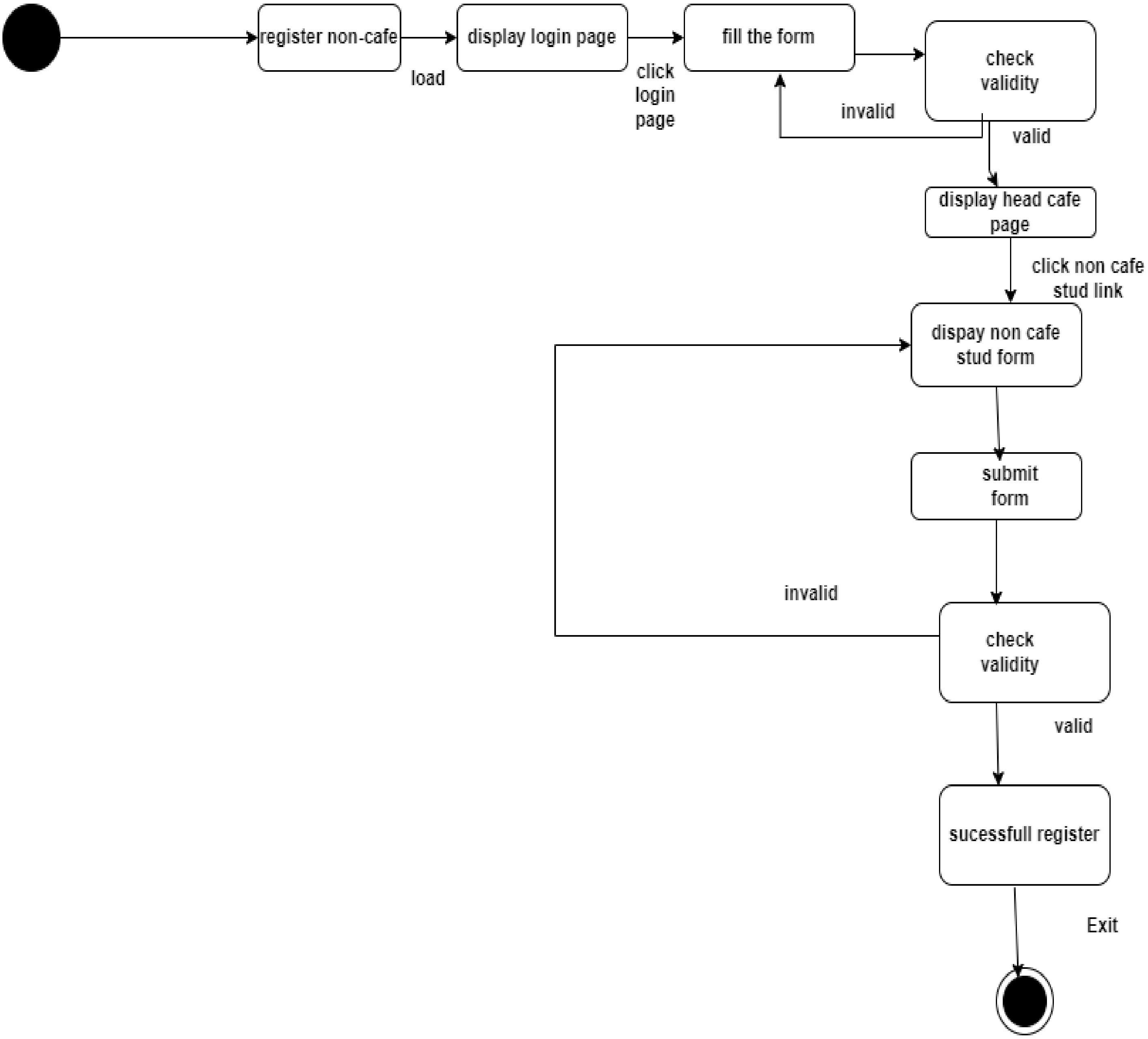


Figure2. 10 state machine diagram for register non café

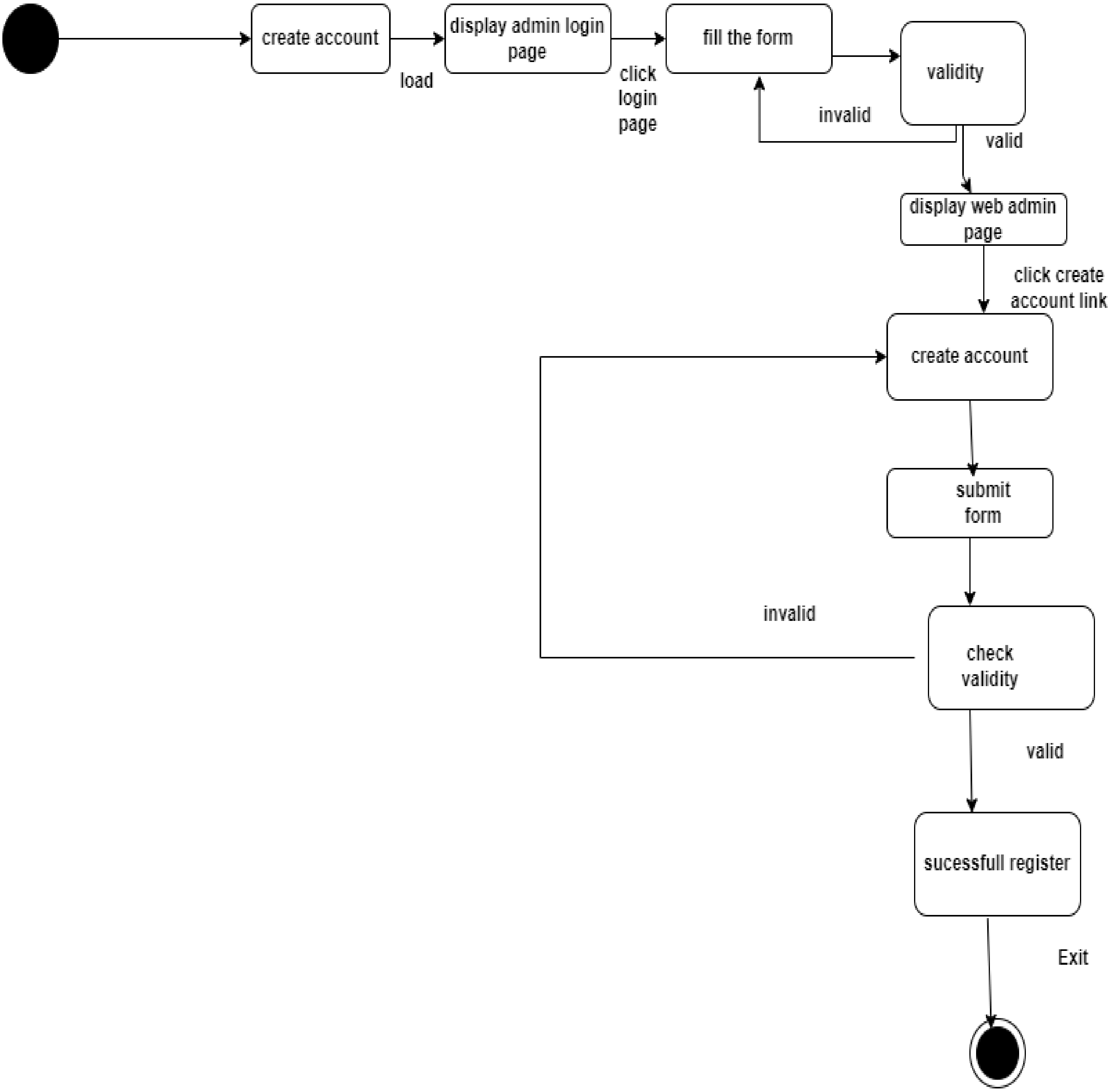


Figure2. 11 State machine diagram for create account

CHAPTER 3

3. System design

System design is the process of defining the components, modules, interfaces and data for a system to satisfy specified requirements.

System design is the transformation of analysis model in to a design model and the detail investigation of system elements from logical view.

3.1 design class Diagram

Design class diagrams are the main building blocks of object oriented modelling.it is used for general conceptual modelling of the structure of application, and for detailed modelling, translating the models in to programming code.

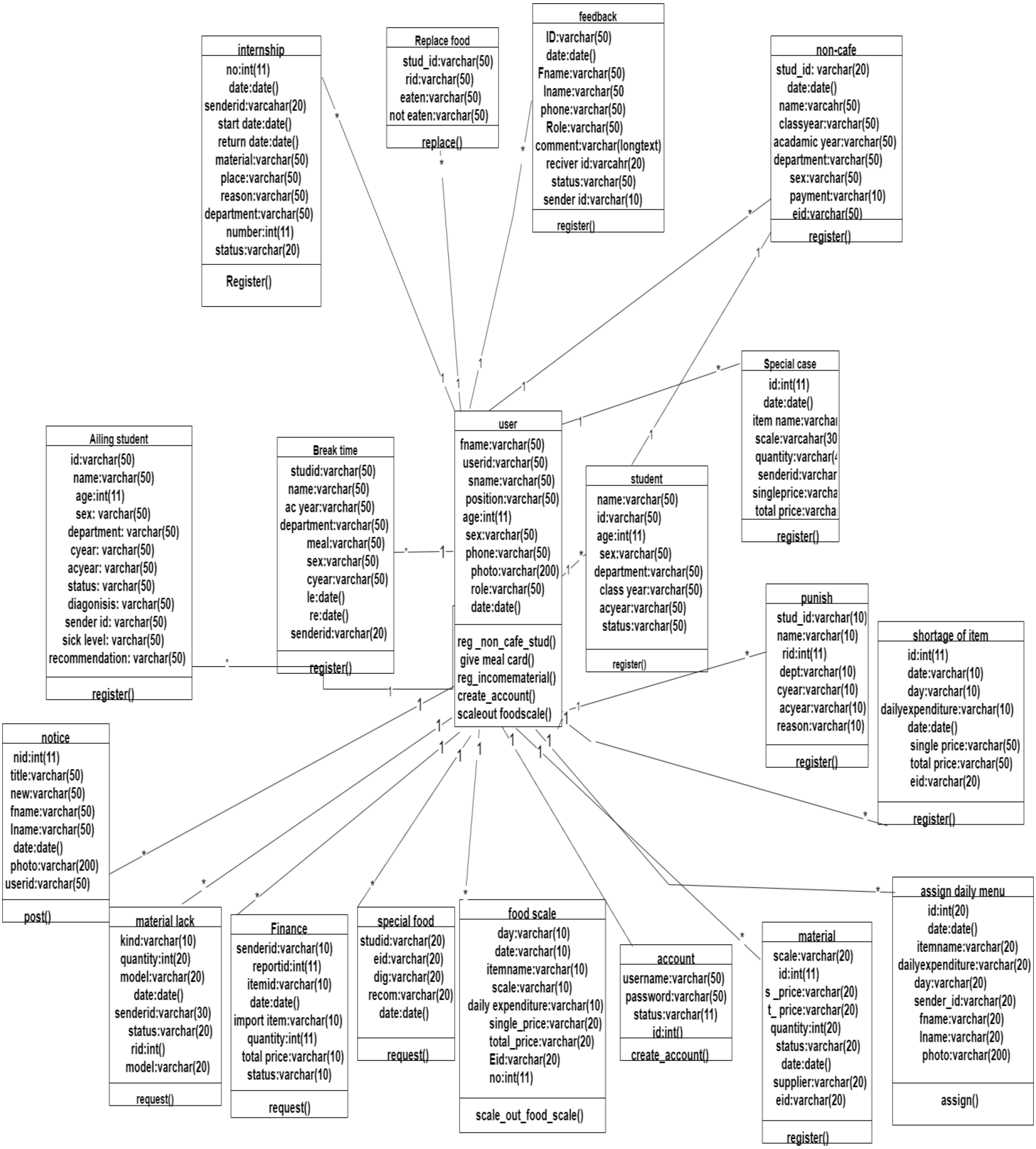
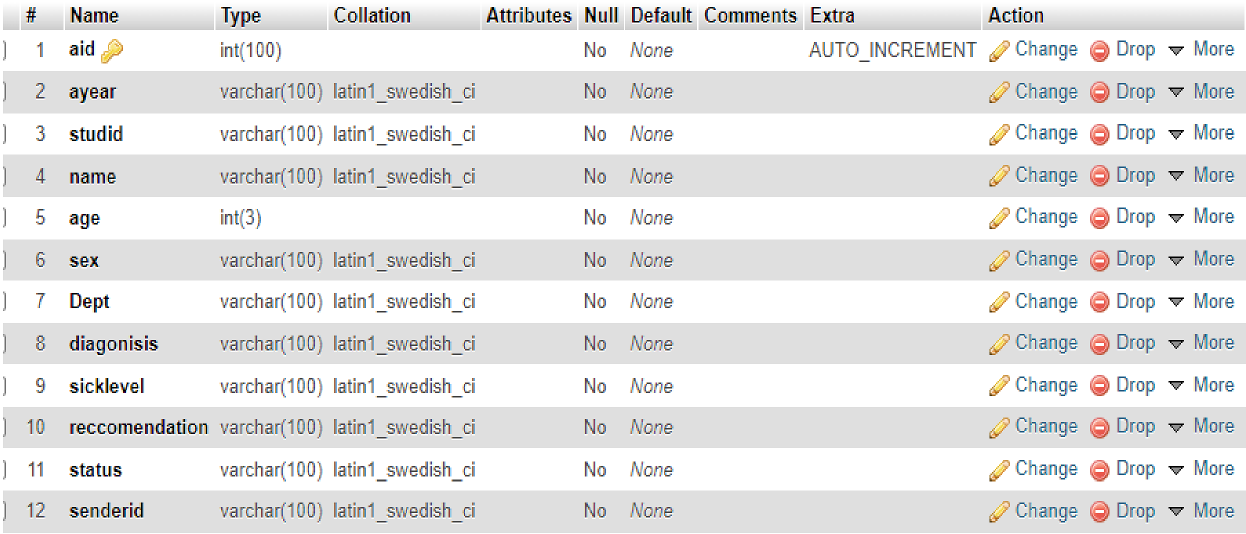


Figure3. 1 Design class diagram

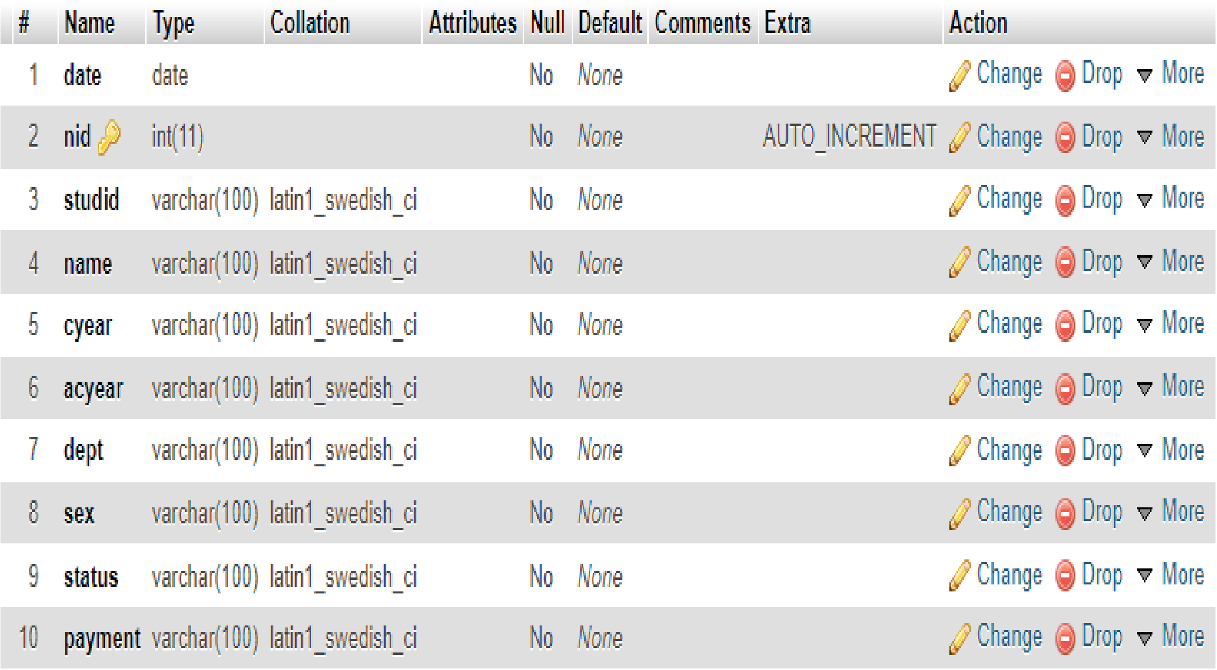
3.2 physical data model

Physical data model is a database specific model that represents relational data objects for example (tables, columns, primary and foreign keys and their relationships. A physical data model can be used to generate data definition language.



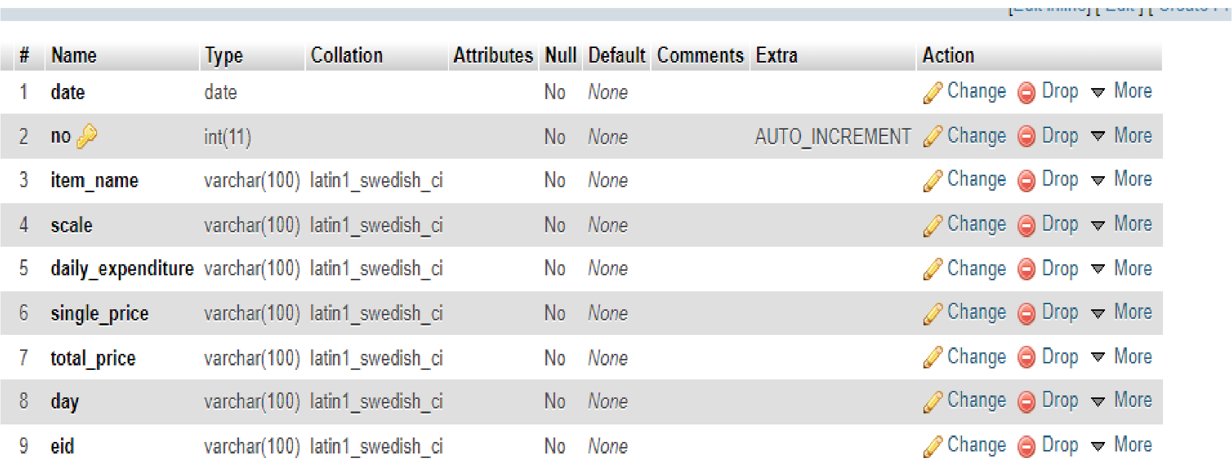
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	aid	int(100)			No	None		AUTO_INCREMENT	Change Drop More
2	ayear	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	studid	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
5	age	int(3)			No	None			Change Drop More
6	sex	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
7	Dept	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
8	diagonisis	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
9	sicklevel	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
10	reccomendation	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
11	status	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
12	senderid	varchar(100)	latin1_swedish_ci		No	None			Change Drop More

Figure3. 2 physical data model for ailing students



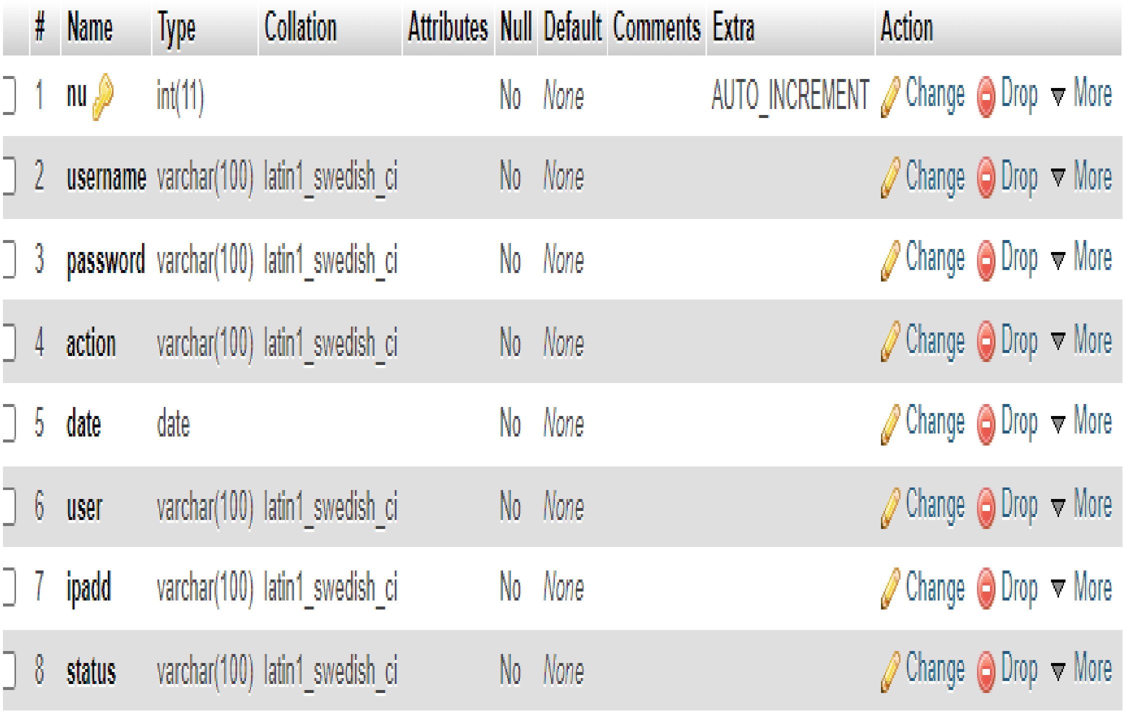
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	date	date			No	None			Change Drop More
2	nid	int(11)			No	None		AUTO_INCREMENT	Change Drop More
3	studid	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
5	cyear	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
6	acyear	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
7	dept	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
8	sex	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
9	status	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
10	payment	varchar(100)	latin1_swedish_ci		No	None			Change Drop More

Figure3. 3 physical data model for non-café students



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	date	date			No	None			Change Drop More
2	no	int(11)			No	None		AUTO_INCREMENT	Change Drop More
3	item_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	scale	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
5	daily_expenditure	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
6	single_price	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
7	total_price	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
8	day	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
9	eid	varchar(100)	latin1_swedish_ci		No	None			Change Drop More

Figure3. 4 physical data model for daily expenditure

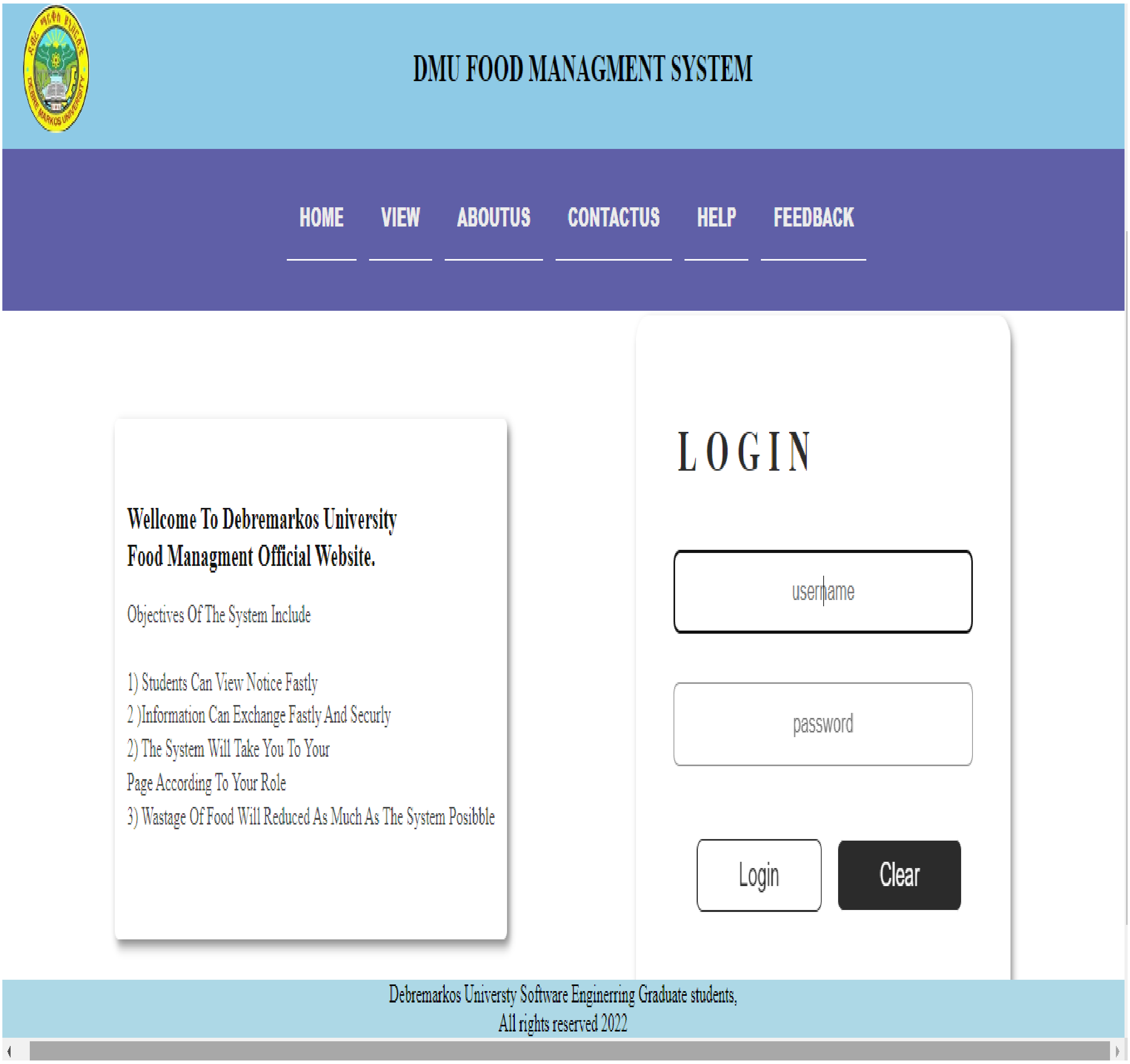


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	nu	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	password	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	action	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
5	date	date			No	None			Change Drop More
6	user	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
7	ipadd	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
8	status	varchar(100)	latin1_swedish_ci		No	None			Change Drop More

Figure3. 5 physical data model for log file

3.3 user interface design

User interface design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand and use to facilitate those actions. user interface brings concepts together from interaction design, visual design, and information architecture.



DMU FOOD MANAGMENT SYSTEM

Wellcome To Debremarkos University Food Managment Official Website.

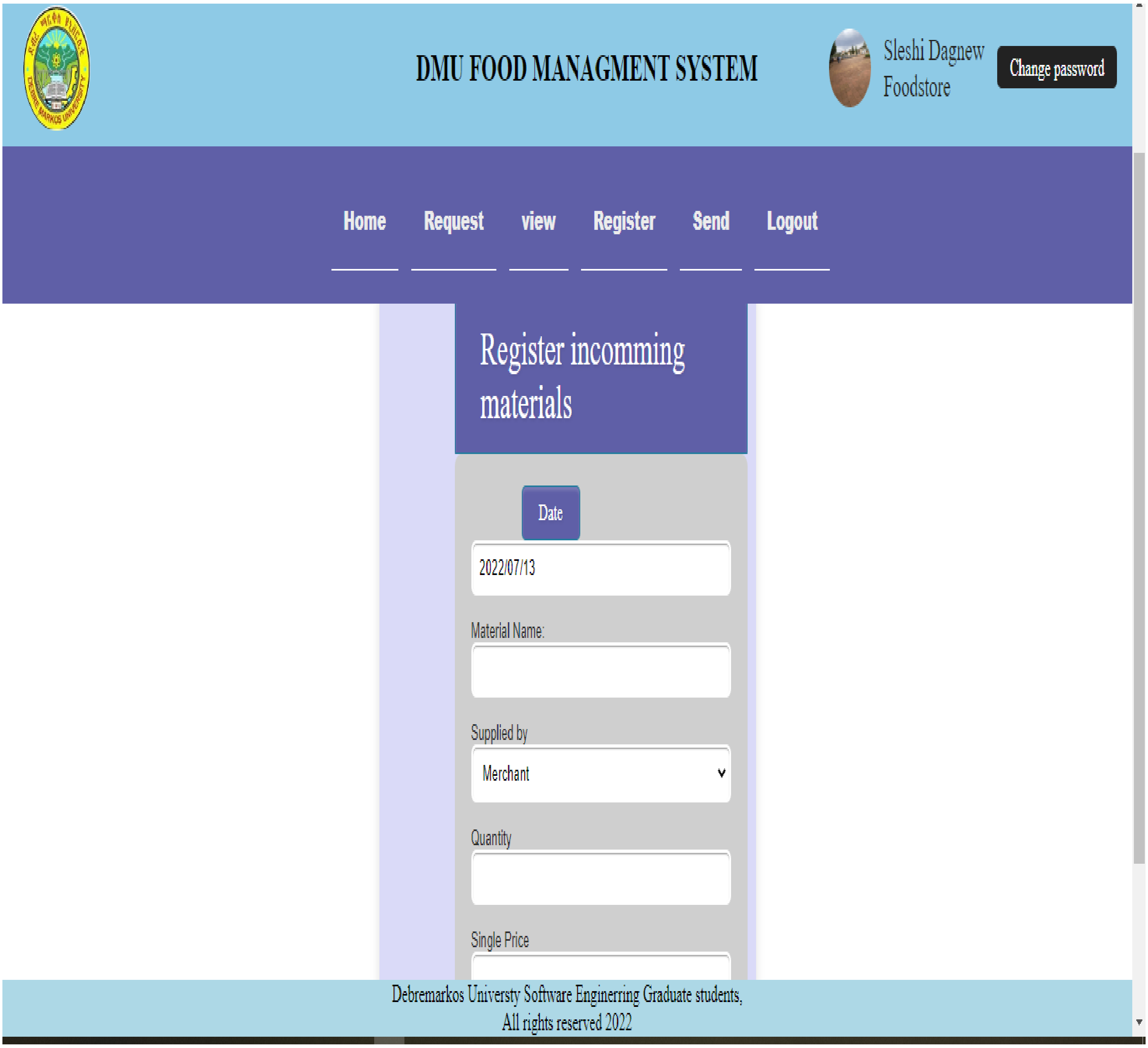
Objectives Of The System Include

1) Students Can View Notice Fastly
2)Information Can Exchange Fastly And Securly
2) The System Will Take You To Your Page According To Your Role
3) Wastage Of Food Will Reduced As Much As The System Posibble

LOGIN

Debremarkos Universty Software Enginerring Graduate students,
All rights reserved 2022

Figure3. 6 main home page interface



Register incomming materials

Date

2022/07/13

Material Name:

Supplied by

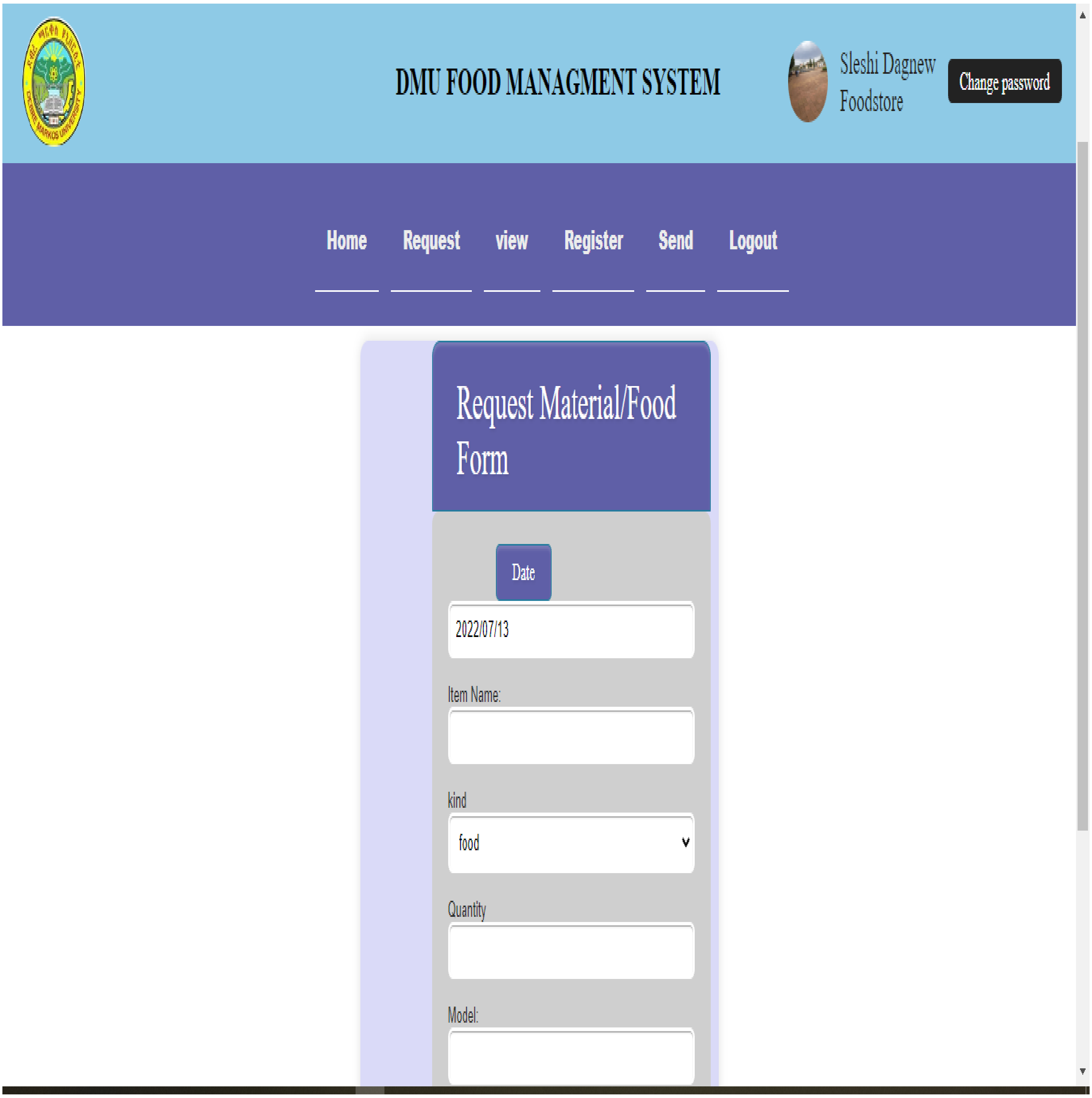
Merchant

Quantity

Single Price

Debremarkos Universty Software Enginerring Graduate students,
All rights reserved 2022

Figure3.7register incoming material



2022/07/13

Item Name:

kind

food

Quantity

Model:

Figure3. 8 request material user interface

3.4 deployment diagram

deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as software and hardware execution environments and the middle ware connecting them.

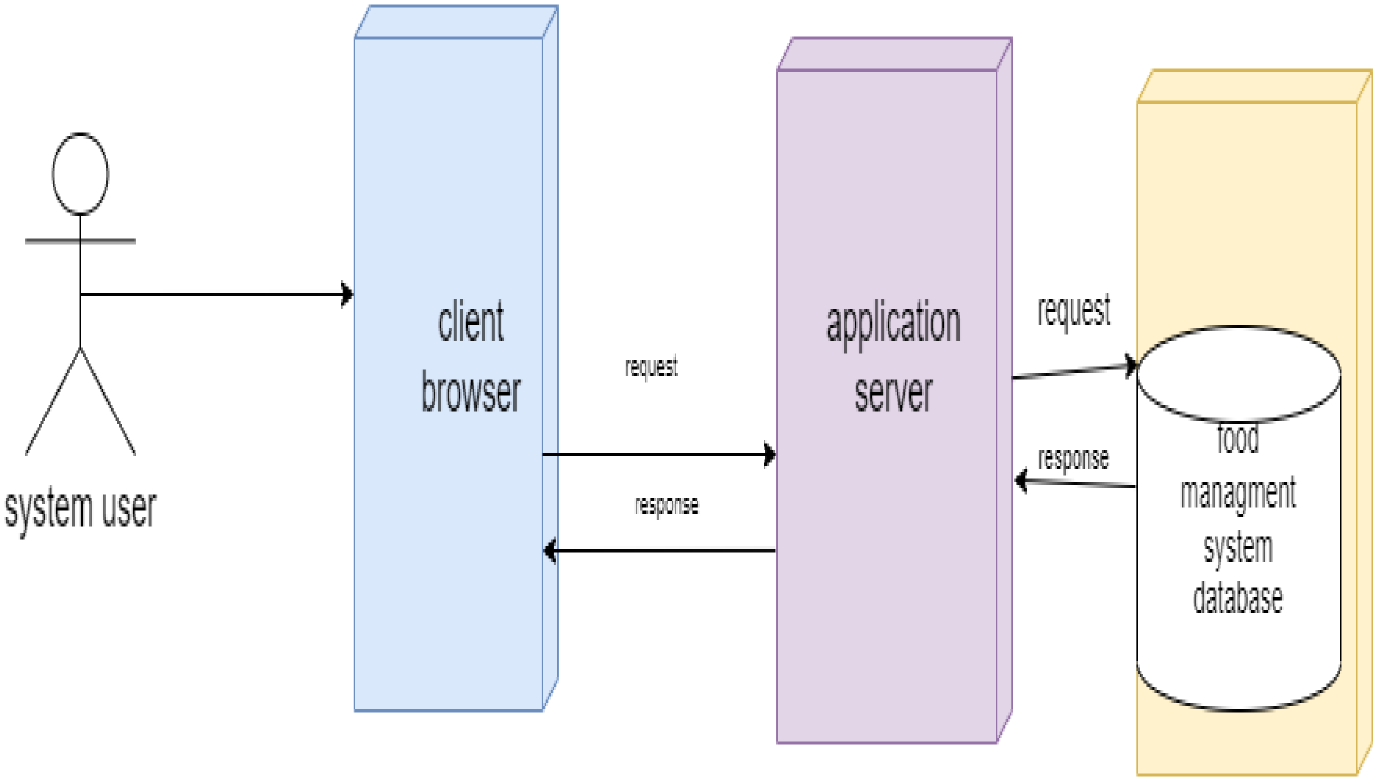


Figure3. 9 Deployment diagram

3.5 component diagram

Component diagram describes the organization and wiring of the physical components in the system. Component diagrams often drawn to help model implementation details and double checks that every aspect of systems required functions is covered by planned development.

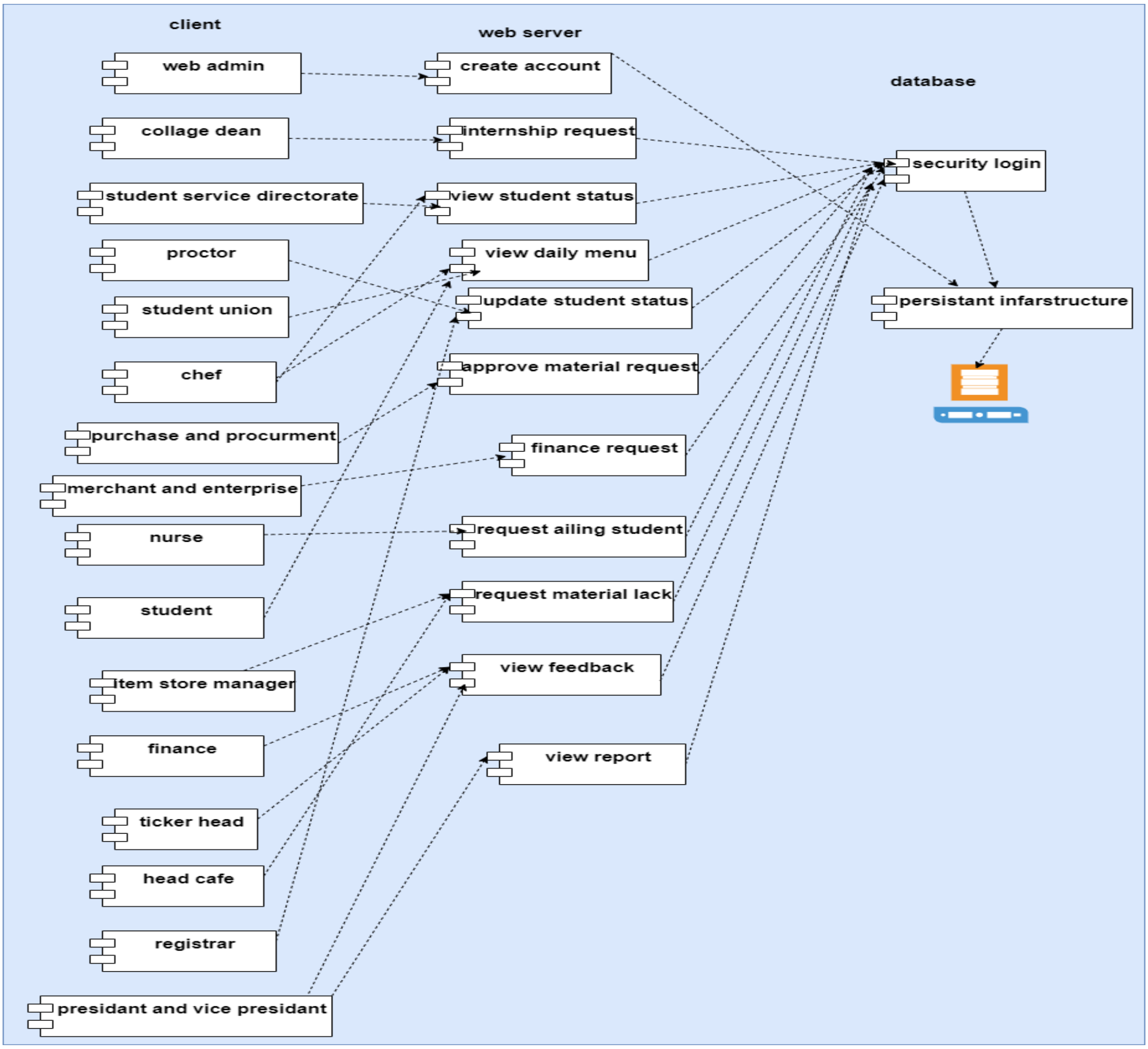


Figure3. 10 Component diagram

3.6 communication/collaboration diagram

A collaboration diagram also known as communication diagram, is an illustration of the intersections among software objects in the unified modelling language. These diagrams can be used to portray the dynamic behaviour of a particular use case and define the role of each object.

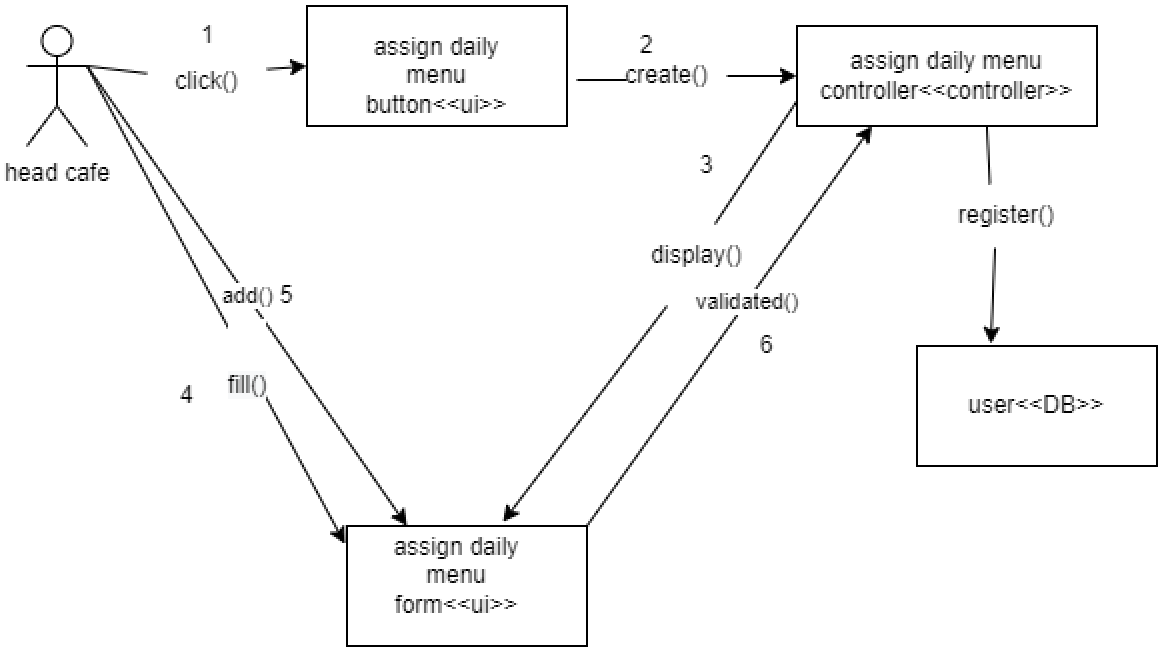

head cafe
1
click()
assign daily menu button<<ui>>
2
create()
assign daily menu controller<<controller>>
3
display()
validated()
6
register()
user<<DB>>
add() 5
4
fill()
assign daily menu form<<ui>>

Figure3. 11 collaboration diagram for head café assigning daily menu

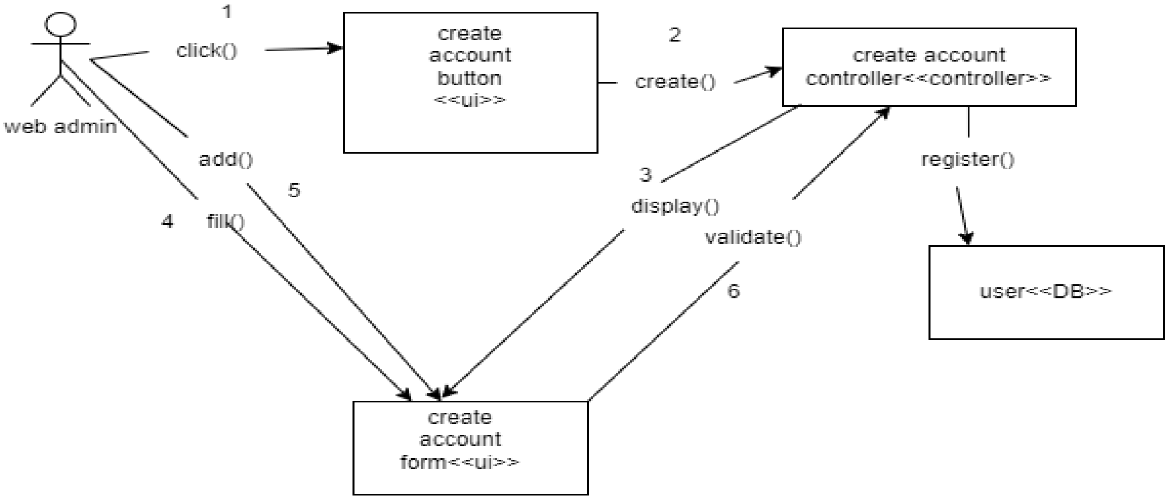

web admin
1
click()
create account button <<ui>>
2
create()
create account controller<<controller>>
3
display()
validate()
6
register()
user<<DB>>
add()
5
4
fill()
create account form<<ui>>

Figure3. 12 Collaboration diagram for web admin creating account

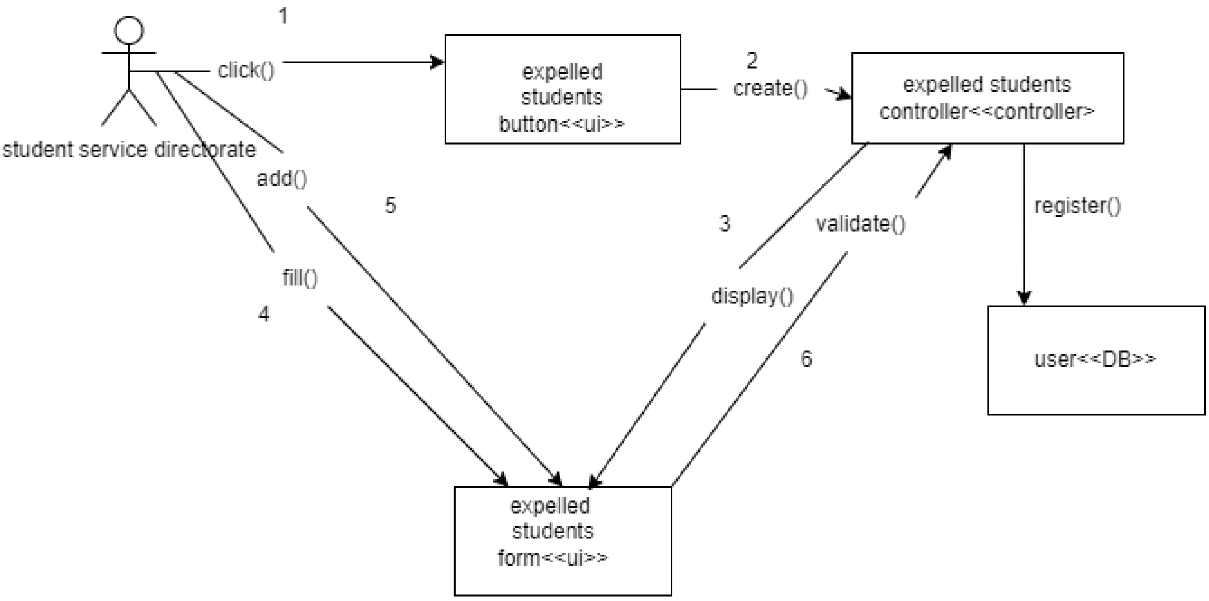

1
click()
expelled students button<<ui>>
2
create()
expelled students controller<<controller>
student service directorate
add()
5
3
validate()
register()
fill()
4
display()
6
user<<DB>>
expelled students form<<ui>>

Figure3. 13 Collaboration diagram for student service directorate

CHAPTER 4

4 Implementation

4.1 Overview of the programming language

For implementing this project, we use the following programming language: -

- PHP **(**We use PHP language for the system development, and our system (software) will be compatible on all hardware platforms such as windows &Linux)
- Scripting language: -used to develop different validations.
- HTML**:** -to display the web page.
- CSS**:** - for the formatting of the web site.

4.2 Algorithms Used

Pseudo code is compact and informal high-level description of a computer programming algorithm that uses the structural conventions of a programming language but is intended for human reading rather than machine reading. The purpose of using pseudo code is that it is easier for humans to understand than conventional programming language code, and that it is a compact and environment-independent description of the key principles of an algorithm.

```
Load item store page
 Open item registration form
Fill
The Login registration form
Click the register button
If (Form is filled)
        If (valid)
                Generate SQL select queries
                Connect to database
                Pass queries to database
                If (any query fails)
                        Display error message
                Else
                        Read session
                        If session exists on database, item is added to food store,
                        Else
                                If they're correct
                                Create session ID
                                Store session ID on database
                                Display the page
                        End if
                End if
        Else
          Display error message
          Ask the user to refill the form
    End if
End if
```

4.3 sample code for register incoming food item

```
<?php

include("../connection.php");

session_start ();

if(empty($_SESSION['user'])) {

   header ('location: ../index.php');

  }

  include("header.php");

  include("navbar.php");

?>

<?php

$queryincoming=mysqli_query ($conn,"select *from incomingfood ");

if($queryincoming) {

        $row=mysqli_fetch_array($queryincoming);
        $quantityincomming=$row['quantity'];
}
$queryshortage=mysqli_query ($conn,"select *from shortageofitem ");
if($queryshortage) {
        $row=mysqli_fetch_array($queryshortage);
        $quantityshortage=$row['dailyexpenditure'];
}
$querylack=mysqli_query ($conn,"select *from materiallack");
if($querylack) {
        $row=mysqli_fetch_array($querylack);
        $namelack=$row['itemname'];
}
?>
<?php
        $dat=date("Y/m/d");
        $tim=date("H:h:i:sa");
        ?>
<?php
        $date=date("Y/m/d");
        $time=date("H:h:i:sa");
        ?>

   <?php
```

```
    if(isset($_SESSION['user']['username'])) {

        $username=$_SESSION['user']['username'];

        $query1=mysqli_query ($conn,"select * from user where user_id='$username'");

        while($query2=mysqli_fetch_array($query1)) {

            $uid=$query2['user_id'];

                                    $fname=$query2['fname'];

                                    $sname=$query2['sname'];

                                    $photo=$query2['photo'];


        }

    }

    ?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="../css/forms.css">

    <title>Document</title>

</head>

<body>

<div class="form-style-10">
```

```
<h2>Registrer Incommming Food</h2>

<form action="" method="POST">

      <div class="inner-wrap">

      <label><input type="button"value="Date"onclick="dat();"/> <input type="text"name="date"
value="<?php echo $dat; ?>" readonly="true"/></label>



      <label>Item Name: <input type="text" name="iname" required="" pattern="^[a-zA-Z()
]+"class="demoInputBox"></label>

      <label>Scale <select name="scale" >

                                                      <option value="kilogram">kilogram</option>

                                                      <option value="liter">liter</option>

                                                      <option value="packet">packet</option>

                                                      <option value="number">number</option>

                                                      <option value="tasa">tasa</option>



                                              </select></label>

      <label>Quantity <input type="text" name="quantity" required="" pattern="^[0-9]
+"class="demoInputBox"></label>

       <label>Supplied by <select name="supply" >

                                                      <option value="merchant">Merchant</option>

                                                      <option value="enterprise">Enterprise</option>

                                                      <option value="university">University</option>



                                              </select></label>
      <label>Single Price <input type="text" name="sprice" pattern="^[a-zA-Z0-9./
]+"class="demoInputBox"></label>

   </div>

   <div class="button-section">

    <input type="submit" name="submit" value="Register" class="btnRegister" >

<input type="reset" name="reset" value="Reset" class="btnRegister">



   </div>

</form>

<?php

         $quan=0;

         if(isset($_POST['submit']))



if(isset($_POST['submit']))

{



         $date=$_POST['date'];

         $inamee=$_POST['iname'];

         $scale=$_POST['scale'];

         $quan=$_POST['quantity'];

         $sprice=$_POST['sprice'];

         $sup=$_POST['suplay'];

         $tprice=$sprice*$quan;
         $sql0=mysqli_query ($conn,"select * FROM incomingfood WHERE itemname='$inamee'");

         $num=mysqli_num_rows($sql0);

         $rowcheck=mysqli_fetch_assoc($sql0);

         if($num=='0')

         {

         $sql="INSERT INTO incomingfood (date, itemname,scale,quantity,sprice,tprice,eid,supplayer,status)
VALUES('$date','$inamee','$scale','$quan','$sprice','$tprice','$uid','$sup','notpay')";

         $query=mysqli_query($conn,$sql);


         $qa="INSERTINTOuserview(date,time,userid,fname,lname,photo,activity)
VALUES('$date','$uid','$fname','$sname','$photo','register new income $quan $scale $inamee')";

         $query1=mysqli_query($conn,$qa);

         echo '<script type="text/javascript">

         alert (" food item is added to store")</script>';

         }

         elseif($quantityincomming>$quantityshortage || $queryincomming===$name lack) {

             $sql=mysqli_query ($conn,"delete from shortageofitem where itemname='$inamee'") or
die(mysqli_error());

             $sql=mysqli_query ($conn,"delete from materiallack where itemname='$namelack'");



             $quantity1=$rowcheck['quantity'];

             $sum=$quantity1+$quan;
```

```
                $id2=$rowcheck['id'];

                $price=$rowcheck['tprice'];

                $totalprice=$price+$tprice;

                $sql2="UPDATE incomingfood set quantity='$sum' , tprice='$totalprice' WHERE id=$id2";

                $query9=mysqli_query ($conn, $sql2);


echo '<script type="text/javascript">

        alert ("Food item is already available so it will be added")</script>';

}}

        ?>


</body>

</html>
```

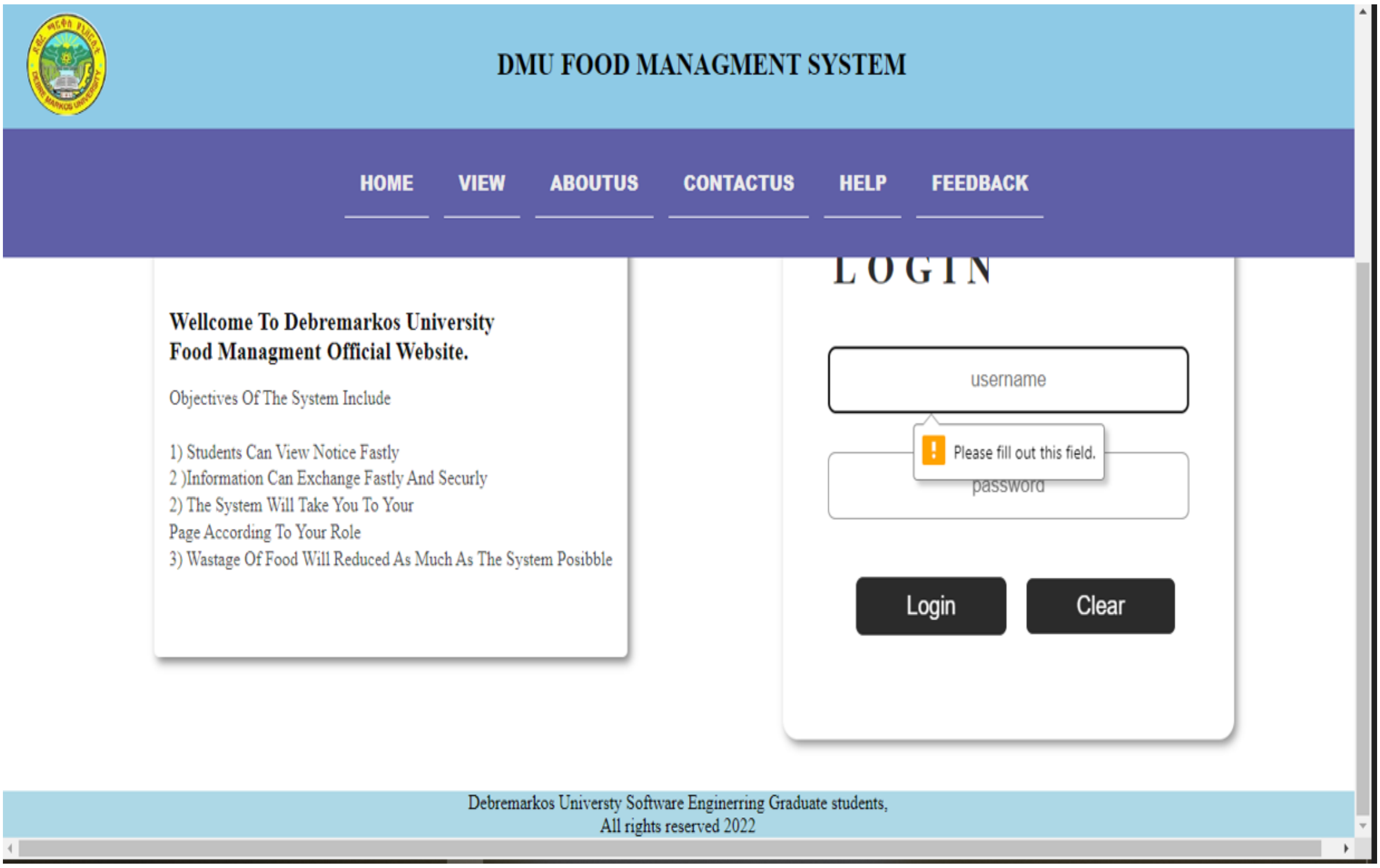
CHAPTER 5

5.1 Testing

Unit testing: -Every module of the System is separately tested. I.e., the team tests every module by applying some selection mechanism. Through this mechanism every module gets tested. If an error occurs correction will be taken without affecting another module.

Integrated testing: - In this testing part, all the modules combined together and tested for fitness with each other and with the systems functionality. If error occurs in combining them, the module with problem will be identified and re combined.

System testing: - In this testing, the team performs over all functional testing by checking whether it meets the required target.



DMU FOOD MANAGMENT SYSTEM

Wellcome To Debremarkos University Food Managment Official Website.

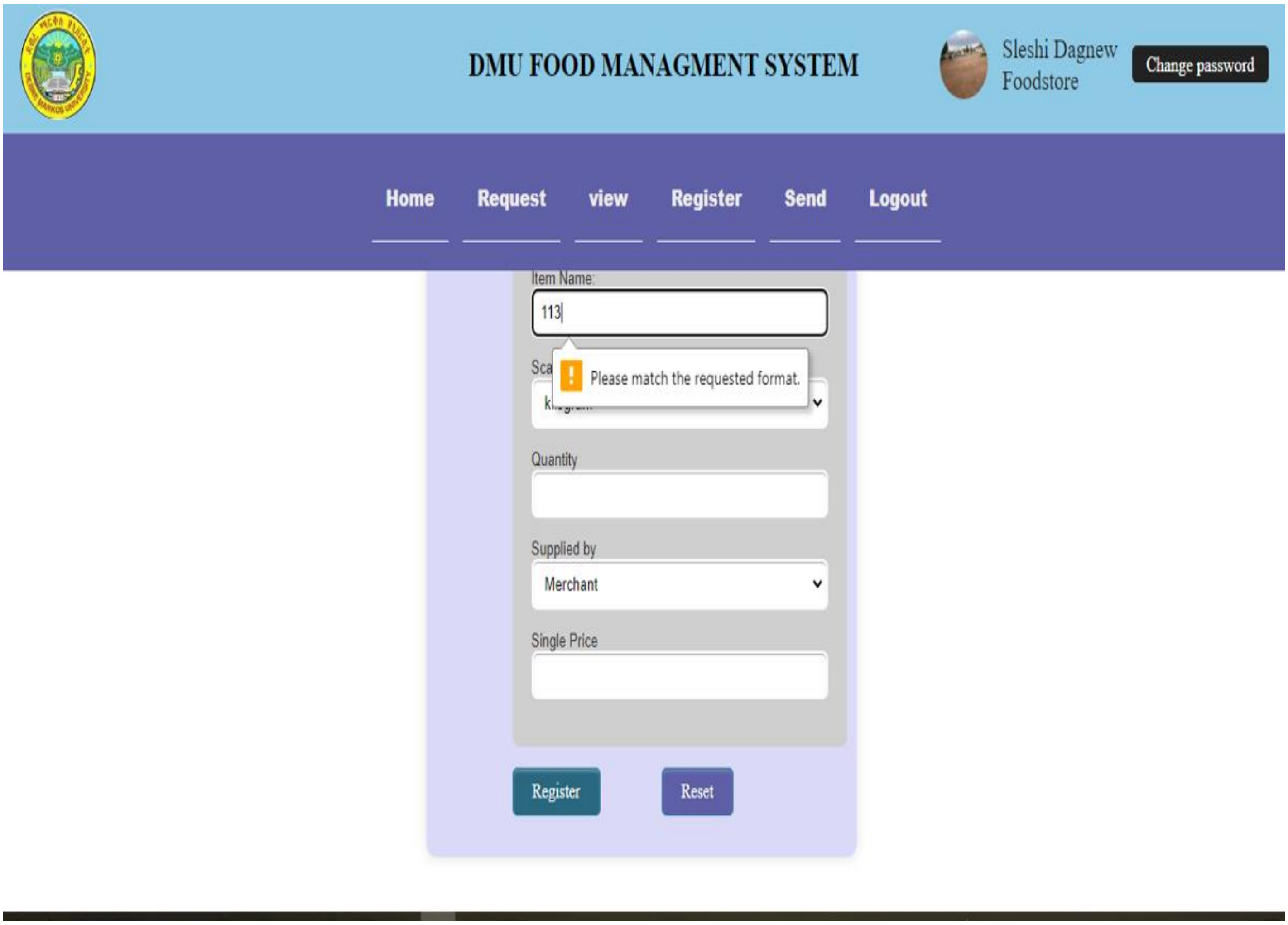
Objectives Of The System Include

1) Students Can View Notice Fastly
2)Information Can Exchange Fastly And Securly
2) The System Will Take You To Your Page According To Your Role
3) Wastage Of Food Will Reduced As Much As The System Posibble

Please fill out this field.

Debremarkos Universty Software Enginerring Graduate students,
All rights reserved 2022

Figure 4. 1 sample test for login



Item Name: 113

Sca... ki...

Please match the requested format.

Quantity

Supplied by: Merchant

Single Price

Figure 4. 2 sample test for registering incoming food

CHAPTER 6

6. Conclusion and Recommendations

6.1. Conclusion

The project titled as “Student’s Food management system” is a web-based application. This system provides online student’s information exchange among stakeholders of the food service, facilitate work process, reduce work load for food service workers like head café, item store manager, student directorate and also for other participants. Since the system is developed by user friendly language there is no fear on failures of the system because it can be easily maintainable.

6.2Recommendation and Future Enhancement

6.2.1. Recommendation

We recommend Debre Markos University to use the developed system in order to facilitate the work process and to reduce wastage of food items since the system can calculate everything regarding to food and the system protect the university from being fraud by eligible stack holders.

6.2.2. Future Enhancement

For future enhancement it is better to include the following activities

- ✓ The system will be developing that support mobile phone to do the tasks that can do without presenting at work place and notification will address with mobile
- ✓ The system will be announced system alert for web admin whenever the change arise that can harm the system and can interrupt this problem.

References

[1] [online].avilable:, “feasibility study/ivestopedia.com,” Accessed date march 16 2022.

[2] [online], “Avilable:/system requirement specification/prefoce.com,” march 23 2022.

[3] [online].Avilable, “functional and non-functional/altesoft.com,” march 12 2022.

[4] [online].Avilable, “https://www.lucidchart.com/pages/uml-sequence-diagram#:~:text=A%20sequence%20diagram%20is%20a,to%20document%20an%20existing%20process.,” [acessed date march 18 2022.

[5] [online].Avillable, “https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/#:~:text=An%20activity%20diagram%20is%20a,the%20activity%20is%20being%20executed.,” [Accessed date march 20 2022].

Appendences

The case tools and software's we used in the project

PHP-is a server-side scripting language used to design the backend.

UML- is a uniform modelling language which we used

HTML-is hyper text markup language which is used to

CSS: - Cascading Style Sheet used to style the interface of web.

Class diagram: Class diagrams show the classes of a system and their interrelationship's. Class diagrams are often mistakenly referred to as object models.

Use case diagram: In UML, the diagram showing the external actors, the system boundary, the use cases as ellipses, and arrows connecting actors to ellipses or ellipses to ellipses. Primarily useful as a context diagram and table of contents.

Use case: A use case expresses a contract between the stakeholders of a system about its behaviour. It describes the system's behaviour and interactions under various conditions as it responds to a request on behalf of the stakeholders, the primary actor, showing how the primary Actor's goal gets delivered or fails. The use case collects together the scenarios related to the primary actor's goal.