



WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION SYSTEMS

CAFETERIA MANAGEMENT SYSTEM USING

FACE RECOGNITION FOR WOLKITE UNIVERSITY

COMPILED: INDUSTRIAL PROJECT

BY

NAME

ID NO

DEJENE TOLERACIR/136/11

DENDEA TAREKEGNCIR/167/11

DHERESA AMANTECIR/139/11

PROJECT ADVISOR: MR. BADIMAW TEREFE

JUNE 14, 2022

WOLKITE UNIVERSITY
COLLEGE OF COMPUTING AND INFORMATICS
DEPARTMENT OF INFORMATION SYSTEMS
CAFETERIA MANAGEMENT SYSTEM USING
FACE RECOGNITION FOR WOLKITE UNIVERSITY
SUBMITTED TO DEPARTMENT OF INFORMATION SYSTEMS
IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE IN INFORMATION SYSTEMS

BY

<u>NAME</u>	<u>ID NO</u>
DEJENE TOLERA	CIR/136/11
DENDEA TAREKEGN	CIR/167/11
DHERESA AMANTE	CIR/139/11

PROJECT ADVISOR: MR. BADIMAW TEREFE

Wolkite University, Wolkite, Ethiopia

June 14, 2022

DECLARATION

This is to declare that this project work which is done under the supervision of Mr. Badimaw Terefe and having the title Cafeteria Management System Using Face Recognition for Wolkite University is the sole contribution of: Dejene Tolera, Dendea Tarekegn, and Dheresa Amante.

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 would be responsible and liable for any consequence if a violation of this declaration is proven.

Date: _____

Group Members:

Full Name

Signature

1. _____

2. _____

3. _____

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to several individuals for supporting us throughout our Industrial Project. First of all, we would like to thank God for helping us in many ways. Next, we would like to express our gratitude to our Advisor **Mr. Badimaw Terefe** for his enthusiasm, patience, insightful comments, helpful information, and practical advice that helped us tremendously at all times throughout our project. We also express our gratitude to Cafeteria Manager **Mr. Getachew Beraga** who support us in different ways by sharing his precious idea and information about the cafeteria management system and its' problems and for our University for accepting us into a program. Lastly, all those who have contributed to our Industrial Project have a special place in us.

TABLES OF CONTENTS

Table of Content	Pages
DECLARATION.....	i
APPROVAL FORM	ii
ACKNOWLEDGEMENT	iii
TABLES OF CONTENTS.....	iv
LIST OF TABLE.....	x
LIST OF FIGURES.....	xi
LIST OF ABBREVIATION	xii
EXECUTIVE SUMMARY	xiii
CHAPTER ONE.....	1
1. INTRODUCTION.....	1
1.1 Background of the Organization.....	2
1.2 Statement of the problems	3
1.3 Objective of the project.....	3
1.3.1 General Objective	3
1.3.2 Specific objectives	4
1.4 Feasibility Analysis.....	4
1.4.1 Operational feasibility.....	4
1.4.2 Technical feasibility	5
1.4.3 Economic feasibility	5
1.4.4 Political feasibility	7
1.4.5 Schedule feasibility	7
1.5 Scope and limitation of the project.....	7
1.5.1 Scope of the project.....	7
1.5.2 Limitations of the project.....	7

1.6 Significance and beneficiaries of the system.....	8
1.6.1 Significance of the project	8
1.6.2 Beneficiaries of the Project	8
1.7 Methodology of the project	9
1.7.1 Data gathering methodology	9
1.7.2 System Analysis and Design methodology.....	9
1.7.3 The system development model.....	10
1.7.4 System testing methodology	11
1.7.5 Development tool and technology	12
1.8. Document Organization.....	14
CHAPTER TWO.....	15
2. DESCRIPTION OF THE EXISTING SYSTEM	15
2.1. Introduction of Existing System	15
2.1.1 Manual book-based cafeteria management system.....	15
2.1.2 Digital ID-based cafeteria management system.....	15
2.2. Users of Existing Cafeteria Management System	16
2.3. Major Functions of the Existing System	17
2.3.1 Major function of the Manual book based system.....	17
2.3.2 Major function of the Digital ID-based system	18
2.4. Forms and Other Documents of the Existing Systems	19
2.5. Drawbacks of the Existing System	20
2.5.1. The drawback of the manual book-based system	21
2.5.2 The drawback of the digital ID-based system.....	21
2.6. Business Rules of the Existing System.....	22
CHAPTER THREE.....	23
3. PROPOSED SYSTEM.....	23
3.1. Functional Requirements	23

3.1.1. System Administrator Module	23
3.1.2. Registrar Officer Module	24
3.1.3. Student Dean Module.....	25
3.1.4. Cafeteria Manager Module	26
3.1.5. Student Controller Module.....	26
3.1.6. Student Union Module	27
3.1.7. Face recognition process	27
3.2. Non-functional Requirements.....	30
3.2.1. User Interface and Human Factors.....	30
3.2.2. Hardware Consideration	31
3.2.3. Security Issues.....	31
3.2.4. Performance Consideration.....	32
3.2.5. Error Handling and Validation.....	32
3.2.6. Quality Issues	32
3.2.7. Backup and Recovery	33
3.2.8. Physical Environment	33
3.2.9. Resource Issues	33
CHAPTER FOUR	34
4. SYSTEM ANALYSIS	34
4.1. System Model	34
4.1.1. Use Case Model	34
4.2. Object Model	50
4.2.1. Class Diagram	50
4.2.2. Data Dictionary	51
4.3. Dynamic Model	53
4.3.1. Sequence Diagram	54

4.3.2. Activity Diagram	60
4.3.3. State Chart Diagram.....	66
CHAPTER FIVE.....	69
5. SYSTEM DESIGN	69
5.1. Design Goals.....	69
5.1.1. User Interface and Human Factors.....	69
5.1.2. Hardware Consideration	69
5.1.3. Security Issues.....	69
5.1.4. Performance Consideration.....	70
5.1.5. Error Handling and Validation.....	70
5.1.6. Quality Issues	70
5.1.7. Backup and Recovery	70
5.1.8. Physical Environment	70
5.1.9. Resource Issues	70
5.2. Current System Process	71
5.3. Proposed System Architecture.....	72
5.3.1. Subsystem Decomposition and Description	73
5.3.2. Hardware/Software Mapping.....	76
5.3.3. Detailed Class Diagram	77
5.3.4. Persistent Data Management.....	78
5.3.5. Access Control and Security	79
5.4. Packages.....	81
5.5. Algorithm Design	82
5.6. User Interface Design	84
CHAPTER SIX	89
6. IMPLEMENTATION AND TESTING	89

6.1. Implementation of the Database	89
6.2. Implementation of the Class Diagram	89
6.3. Configuration of the application server	89
6.4. Configuration of Application Security	89
6.5. Implementation of User Interface	90
6.6. Testing the System.....	90
6.6.1. Test Case	91
6.6.2. Testing tools and Environments.....	91
6.6.3. Unit testing	92
6.6.4. System testing	92
6.6.5. Integration testing	92
6.7. Model Development	93
6.7.1. Convolutional Layer	93
6.7.2. Max Pooling Layer.....	93
6.7.3. Flatten layer.....	94
6.7.4. Dense Layer (Fully connected layer)	94
6.7.5. Train the network compilation processes.....	95
6.7. Activation function	96
6.8. Loss function.....	96
6.9. Optimization Algorithm.....	96
6.10. Learning Rate.....	97
6.11. Summary () function	97
6.12. Epoch	98
6.13. Discussion of the Result	98
CHAPTER SEVEN	99
7. CONCLUSION AND RECOMMENDATION.....	99
7.1. CONCLUSION.....	99

7.2 RECOMMENDATION.....	100
8. REFERENCES	101
9. APPENDICES.....	102

LIST OF TABLE

Table 1. 1 Materials cost for the existing system.....	6
Table 1. 2 Materials cost for the proposed system.....	6
Table 2. 1 User of existing system.....	17
Table 3. 1 Hardware consideration	31
Table 4. 1 Use case Identification.....	34
Table 4. 2 Use Case Description of Login.....	39
Table 4. 3 Use Case Description of Add User	40
Table 4. 4 Use Case Description of Register Student.....	41
Table 4. 5 Use Case Description of Taking Image	43
Table 4. 6 Use Case Description of View Report.....	45
Table 4. 7 Use Case Description of View Feedback	46
Table 4. 8 Use Case Description of Log out.....	47
Table 4. 9 Data Dictionary for User.....	51
Table 4. 10 Data Dictionary for Student.....	52
Table 4. 11 Data Dictionary for Student Status.....	52
Table 4. 12 Data Dictionary for Feedback.....	53
Table 5. 1 Access control and security	80

LIST OF FIGURES

Figure 2. 1 User structure of digital ID-based system	16
Figure 2. 2 Manual Book form	19
Figure 2. 3 Digital ID form	20
Figure 3. 1 Convolutional Neural Network (CNN) Architecture	29
Figure 4. 1 Use case diagram	37
Figure 4. 2 Class diagram	49
Figure 4. 3 Sequence diagram for Login	53
Figure 4. 4 Sequence diagram for Add User.....	54
Figure 4. 5 Sequence diagram for Register Student	55
Figure 4. 6 Sequence diagram for taking Student's Image.....	56
Figure 4. 7 Sequence diagram for Initiate System	57
Figure 4. 8 Sequence diagram for View Report	58
Figure 4. 9 Sequence diagram for Logout	58
Figure 4. 10 Activity diagram for Login.....	59
Figure 4. 11 Activity diagram for Add User.....	60
Figure 4. 12 Activity diagram for Register Student.....	61
Figure 4. 13 Activity diagram for Taking Student's Image.....	62
Figure 4. 14 Activity diagram for Initiate System	63
Figure 4. 15 Activity diagram for Face Recognition	64
Figure 4. 16 State chart diagram for Add User.....	65
Figure 4. 17 State chart diagram for Register Student.....	66
Figure 4. 18 State chart diagram for Face Recognition	67
Figure 5. 1 Current system process.....	70
Figure 5. 2 Proposed system Process	71
Figure 5. 3 General proposed architecture	72
Figure 5. 4 Subsystem decomposing	74
Figure 5. 5 Hardware/Software diagram.....	75
Figure 5. 6 Detail class diagram	76
Figure 5. 7 Persistent data management	78
Figure 5. 8 Packages diagram	80
Figure 5. 9 Login page	83
Figure 5. 10 Admin home page.....	84
Figure 5. 11 Add user page	85
Figure 5. 12 Initiate system page	86
Figure 5. 13 Manage feedback page	87

LIST OF ABBREVIATION

No.	Abbreviation	Description
1	Admin	System Administrator
2	BR	Business Rule
3	CSS	Cascading Style Sheet
4	CNN	Convolutional Neural Network
5	DTL	Django Template Language
6	E.C	Ethiopian Calendar
7	ETB	Ethiopian Birr
8	HTML	Hypertext Markup Language
9	ID	Identification Number
10	MS	Microsoft
11	MTCNN	Multi-Task Convolutional Neural Network
12	MVT	Model View Template
13	OOSAD	Object-Oriented System Analysis and Design
14	PBKDF2	Password-Based Key Derivation Function 2
15	RAM	Random Access Memory
16	SIMS	Student Information Management System
17	SNNPR	South Nation Nationality of People's Region
18	UCID	Use Case Identifier
19	UML	Unified Modeling Language
20	WKU	Wolkite University

EXECUTIVE SUMMARY

Cafeteria Management System using Face Recognition is a Machine Learning system application and it provides to handles the activity of giving food service to Students using face recognition by generating a dataset of Student's face images, training it, detecting and recognizing the Students' faces to identify and control the Student to eat the food at the Cafeteria entry.

The current mechanism for handling the activity of the Cafeteria Management System of the Wolkite University uses manual books for fresh students and uses digital IIDs for senior students at cafeteria entry is prevent a student who lost or forget his/her ID, the student can get food service by another ID and can hold double ID, and the cost to prepare the student's ID.

The main objective of this project is to develop Cafeteria Management System using Face Recognition for Wolkite University. Different Methodology is used to solve the problem throughout the development of the project. Interview, direct observation for requirement gathering, Object-Oriented system analysis, and design and iterative model for development approach.

Cafeteria Management System using Face Recognition is to solve problems in the existing system with the use of the basic idea of image processing. The system first stores the face in the dataset, and then the faces are detected in the images. The detected faces are compared with the faces stored in the dataset during face recognition. If the system recognizes faces, the student status at each period (breakfast, lunch, and dinner) is marked immediately by the recognized face.

CHAPTER ONE

1. INTRODUCTION

Cafeteria Management System is a system used to manage the activities that take place at the cafeteria food service for students at the university. The current mechanism for handling the activity of the Cafeteria Management System of the Wolkite University is a semi-automated system that operates manually for fresh students using the manual book and an automated manner for senior students using digital ID technology. The manual method use pen for marking a book and is difficult to manage students, and the automated method uses a digital card for giving service. In both ways, the current system faces major problems. The main problem is that for a student to get access to the food service, he/she must have his manual book (in the manual system) or his/her digital ID at hand. But these objects are losable in their nature. As a result, a student who lost his/her object cannot get the service at all.

In order to solve the aforementioned problem, we developed a new system called ‘Cafeteria Management System using Face Recognition’ which uses the most efficient and latest Artificial Intelligence scheme, face recognition, to handle the cafeteria management system by recognizing a face of a student together with a web feature. This allows a student to get food services only using his/her physiological features (his face in this case) which do not have the possibility of getting lost.

The system developed would work as follows; first, students would be registered into the system by a registrar officer. Students then would be categorized as café and non-café by the student dean. The main focus of the system is only on café students. Second, the system admin prepares an image dataset for every student registered by the registrar officer and categorized as café by the student dean. Third, a model would be built and it would be trained using the image data prepared previously. The final and major activity of our system would be detecting the faces of students and deciding whether to allow or prohibit a student from entering the cafeteria.

1.1 Background of the Organization

Wolkite University is one of the third-generation universities in Ethiopia. It was established in 2004 E.C and is found in the southern nation nationality and Peoples (SNNPR) state Gurage zone in Wolkite specifically in the Gubreye sub-city. As a higher educational institution, besides teaching-learning, participating in research, projects, and community services on the major identified prioritized problem areas are the concern of the university. The University gives educational services for regular students as well as extension programs for students.

Wolkite University has the **vision** of being one of the Applied Science universities in Ethiopia by 2022 E.C. Also, the organization has the following **missions**: -To produce graduates who are knowledgeable, attitudinally mature, and practically innovative; to supply relevant demanded technology and knowledge that address national and community level development problems; to help make the operation of the government and non-government origination efficient effective and competitive; to provide training and consulting services to the community and the government. The **core** values of WKU are Excellency, innovativeness, inclusiveness, the truth of integrity, being a learning organization, accountability, and academic freedom.

The cafeteria is one of the university's organizational units and it started in 2004E.C to provide food service for Students. Therefore, it follows the above-mentioned vision, mission, and core values.

In 2004 E.C the Cafeteria System started working by having 4500 students and 200 employees at the first time. Starting from that time, it was facing some challenges. For better understanding, it is better to see those challenges in two time periods.

The first one was the time 2004-2007 E.C. During this time period, the cafeteria system was functioning manually without having any computerized system. The challenges associated with this situation were not enough manpower, difficult working structure because of the unfinished building and not enough infrastructures.

The second one was the time 2008 E.C. until now. During this time period, the cafeteria makes use of two approaches; a manual approach to manage or control the fresh students and a

computerized approach or a digital ID-based system to manage the senior and above students. Today around 12000 students in the university get food service from the cafeteria and the cafeteria serves them using either of the two approaches based on their batches.

1.2 Statement of the problems

Currently, the services that are given at Wolkite University are handled by a semi-automated system that operates manually and automated methods to control and identify the students who getting food service from the cafeteria.

The manual method use pen for marking a book and is difficult to manage students, and the automated method uses a digital card for giving service. In both ways, the current system faces major problems. The main problem is that in order for a student to get access to food service, he/she must have his manual book (in the manual system) or his/her digital ID at hand. But these objects are losable in their nature. As a result, a student who lost his/her object cannot get the service at all. And also, a student may hold the double-manual book and digital ID and can get access to food service with another student's manual book and digital ID. Lastly, a cost to prepare the student's digital ID and the manual book because from time to time the number of students is increasing as the University is growing fast.

To overcome the above problems in the existing cafeteria management system we are going to develop a Cafeteria Management System using Face Recognition which uses the basic idea of image processing. The system first stores the face in the dataset, and then the faces are detected in the images. The detected faces are compared with the faces stored in the dataset during face recognition. If the system recognizes faces, the student status at each period (breakfast, lunch, and dinner) is marked immediately by the recognized face. Finally, the system can generate reports based on the user's needs from the student status.

1.3 Objective of the project

1.3.1 General Objective

The general objective of this project is to develop Cafeteria Management System using Face Recognition for Wolkite University.

1.3.2 Specific objectives

We can achieve the General objective stated above by fulfilling the following specific objectives.

- ✓ To identify problems and analyze the existing manual and automated system.
- ✓ To design and develop the system based on the requirements identified.
- ✓ To develop a prototype for the proposed system.
- ✓ To test the new system.
- ✓ To store the student's face in the dataset.
- ✓ To detect the face segment from a video frame.
- ✓ To extract the useful features from the face detected.
- ✓ To classify the features to recognize the face detected.
- ✓ To record the student status of the identified student.
- ✓ To produce reports from the student status database.
- ✓ To deploy the new system.

1.4 Feasibility Analysis

A feasibility study is a test of a system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources. It is essential to evaluate the cost and benefits of the new system [1].

1.4.1 Operational feasibility

It is a measure of how well a proposed system solves the problems. The proposed system solves the problem for the organization. Therefore, the Admin, Cafeteria Manager, Registrar Officer, Student Dean, Student Controller, and Student Union can get efficient service from the system.

- ✓ When the users access the system they would not face a problem, we mean it would have accuracy and simplicity during users perform their operations.
- ✓ The system to be developed would provide accurate, active, and secured service.
- ✓ More efficient and user-friendly.
- ✓ The new system would increase and improve the activity of the cafeteria management system by providing a short response time, and increasing the efficiency of work in the cafeteria management system since the system was operationally feasible.

1.4.2 Technical feasibility

The proposed system can be easily maintained and repaired without requiring high Experts or Technical support because the system would be installed in adaptable technologies and the employees of the organization have some knowledge about technology by providing training and help on how to use the system, and can use the system easily.

The required technologies such as Python, PostgreSQL, Django Framework, Bootstraps, JavaScript, HTML, and CSS integrated with Django, MS Word 2013 which we would use to work for this project are available, familiar, free, widely used, maintainable, repair and they are compatible with the hardware that we would use. Also, human resources like requirement elicitor, analyzer, designer, programmer, and tester for this project are available. We expect that the system can be operated simply and all users can access it easily by gaining simple training. Thus, the proposed system was technically feasible.

1.4.3 Economic feasibility

The existing system manages the fresh student using the manual book, the senior and above students using a digital ID. Since this project to develop the system, by now the reduction of cost for the manual book, and digital ID, and decreasing the workload of the users which in turn is beneficial to the organization because from time to time the number of students is increasing as the University is growing fast.

Generally, the system that we would develop, would bring several benefits which can be further divided into tangible and intangible benefits.

I) Tangible benefits: - reduces resources used and time spent during manual one.

- ✓ The workload is decreased.
- ✓ Decreasing the Hardware cost like - manual books, and digital ID.
- ✓ Miscellaneous Cost.
- ✓ Decreasing a report generating using manual, and no redundancy.

Table 1. 1 Materials cost for the existing system

Item	Quantity per year	Unit price	Total price
Manual book	3500	65 ETB	$3500 * 65 = 227,500$ ETB
Digital ID	3000	160 ETB	$3000 * 160 = 480,000$ ETB
Total cost			707,500 ETB

Table 1. 2 Materials cost for the proposed system

Item	Quantity per five years at least	Unit price	Total price
Camera	6	100,000 ETB	$6 * 100,000 = 600,000$ ETB
Total cost			600,000 ETB

Cost-Benefit Analysis

The existing system cost per year = 707,500 ETB and also per five year $707,500 * 5 = 3,537,500$ ETB. The proposed system cost per five years at least = 600,000 ETB, Moreover, the proposed system would remove almost all usage of manual book and digital ID in the existing cafeteria management system. Cost Reduction per five years = $3,537,500$ ETB - $600,000$ ETB = $2,937,500$ ETB.

The above cost prediction of the existing system is only for five years but since our university would continue working for n years the cost would also be multiplied by n years. So we can say that the existing system is economically infeasible, but the cost prediction of the proposed system for five years was economically feasible.

II) Intangible Benefits: -those benefits that are believed to be difficult to quantify from our system easily are:

- ✓ Improvement of management planning, and control.
- ✓ Increasing processing efficiency, faster decision making.
- ✓ Error reduction.
- ✓ Increased speed of activity and flexibility of the system.
- ✓ Increase accuracy and authentication.

1.4.4 Political feasibility

We developed a new system was not contradict or conflict with the politics of the current system. So we developed a new system by comparing the current situation of Cafeteria Management System rules and regulations. So no situation exists that in our system conflicts with the current system.

1.4.5 Schedule feasibility

The schedule for this project would be feasible due to the 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. This project is scheduled feasibly because we hope that it would be completed within the time scheduled.

1.5 Scope and limitation of the project

1.5.1 Scope of the project

This project would address the following subsystems:-

- ✓ Our system's focus is only to detect students at cafeteria entrances and be able to allow or prohibit them from entering the cafeteria.
- ✓ It would only be about café students.
- ✓ User management
- ✓ Student management
- ✓ Face recognition and detection
- ✓ Report management
- ✓ Feedback management

1.5.2 Limitations of the project

We don't include:

- ✓ It could not accurately recognize the twine's face.
- ✓ It could not accurately recognize the student, who has the eyeglass and cap or hat.

1.6 Significance and beneficiaries of the system.

1.6.1 Significance of the project

This project is finished and properly used it gives services and great benefits for Wolkite University to control the student when giving access food service at Cafeteria.

Generally, the significance of the new system includes:

- ✓ Used to record data on the database
- ✓ Used to save time.
- ✓ Ensure data accuracies.
- ✓ Increase efficiency.
- ✓ Used to handle the error.
- ✓ Used to manage student's information easily
- ✓ Used to make the recording and retrieving very fast.
- ✓ Used to reduce organization expense.

1.6.2 Beneficiaries of the Project

The major beneficiary of the system includes: -

For the Team Members

- ✓ The project has been initiating our team to get knowledge of how to develop the required system.
- ✓ The team members are getting a lot of experience in solving the problem.
- ✓ It helps to create team spirit among the development team.
- ✓ Personal satisfaction for developing the system.
- ✓ Enables us to get enough marks.

For the Wolkite University

- ✓ The proposed system can reduce university expenses.
- ✓ The proposed system contributes to the productivity of the organization by making the cafeteria management system was easier and more secure.
- ✓ To ensure the university goal.

For the Student Controller

- ✓ The proposed system helps the Student Controller in making the controlling and analyzing process easier and more comfortable.
- ✓ Save time for work to be done.

For the Student

The proposed system helps the students in getting access to food service easier way when students lost or forgot their manual book and digital ID.

1.7 Methodology of the project

1.7.1 Data gathering methodology

Data collection methods or fact-gathering techniques which are used while gathering the information about Cafeteria Management System are: -

Interviewing:

- ✓ The project has an interview for interviewing the cafeteria manager, thicker, student, student dean, system admin, registrar officer, and student union for recording the existing working procedure of the organization.
- ✓ We have used the interview to gather detailed information about the organization manager through face-to-face contact and we have asked all things about how the existing system works.

Direct observation:

- ✓ By observing the existing system, we get some information on their activities and services. We observe the existing systems and how they perform their work manually and automated methods with the students.

1.7.2 System Analysis and Design methodology

Our project was developed by using the object-oriented system analysis and design (**OOSAD**) methodology. Since this system development methodology has so many important features. Some of them are: -

- ✓ **System stability:** - Object-oriented techniques tend to produce systems that are resilient to change, where changes can be made without major disruption, with minimal time and effort, and with little fear of disturbing something else in the system.
- ✓ **Maintainability:** - Object-oriented methods produce systems that can be maintained and enhanced more readily.
- ✓ **Reusable components:**-Object-oriented features such as inheritance and polymorphism lead to smoother, more efficient reuse in both code reuse and design reuse.
- ✓ **Reality-based systems:** - The techniques used with object-oriented methods give a far more accurate picture of the users' business operation and its information needs, leading to a final system that is closer to what the customer wants.
- ✓ **Data accessibility:** - Object-oriented methods tend to lead to better design of databases, this data is more accessible and usable.

1.7.3 The system development model

We used the Iterative system development model to develop Cafeteria Management System using Face Recognition for Wolkite University. This iterative process starts with the requirement stage, then design followed by implementation. Iterative development of each phase is undertaken in distinct increments based on requirements, user feedback, and other analysis tools.

- ✓ Iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.
- ✓ The life cycle model does not attempt to start with a full specification of requirements.
- ✓ This process is then repeated, producing a new version of the software at the end of each iteration of the model.
- ✓ The progress is easily measurable.
- ✓ More flexible – less costly to change scope and requirements.
- ✓ Easier to test and debug during a smaller iteration.
- ✓ Easier to manage risk because risky pieces are identified and handled during its iteration.

1.7.4 System testing methodology

Our group mainly focuses on the key issues that occur in the testing process and is to be taken into consideration by the project manager, a team of developers, and testers. Testing methodologies are the methods or approaches to testing that includes: - unit testing, integration testing, acceptance, and system testing.

Our goal is to ensure the high control of students using face recognition at Cafeteria entry. This system enables ensures the Wolkite University's goal to be one of the Applied Science Universities.

Unit Testing:

In this level of the testing process, we would test the different sub-procedures or functions.

- ✓ Check whether the return type of the functions is correct.
- ✓ Check how the sub-procedures or functions are called correctly.
- ✓ Check if the correct output is produced for different inputs.
- ✓ Check the efficiency of the code concerning the memory and CPU time.

Integration Testing:

In this level of testing, we would examine how the different procedures work together to achieve the goal of the subsystem. This is tested by our team or developer.

System Testing:

In this level of the testing process, we would examine how nicely the subsystems of the whole management system work together to achieve the desired goal. This is tested by the developer and user.

Acceptance Testing:

In this level testing process, the system would be tested by the user who checks how we are and what the system looks like by their testing mechanism. This is the final testing in the system development .so, this test system is on the user side.

1.7.5 Development tool and technology

1.7.5.1. Front-End Technologies

The part of a website that the user interacts with directly is termed the front end. It is also referred to as the ‘client side’ of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and a navigation menu. HTML, CSS, JavaScript, and Bootstraps are the languages used for Front End development.

❖ HTML 5

- ✓ Include new attributes, elements, and abilities.
- ✓ Makes creating accessible sites easier.
- ✓ All browsers support HTML.
- ✓ Integrated with Django Template Language.

❖ CSS

- ✓ It allows adapting the presentation to different types of devices, such as large screens, small screens, or printers.
- ✓ It allows for position and re-positions of the components of a web page.
- ✓ It is browser compatible.

❖ JavaScript

- ✓ To produce the same result on all modern browsers.
- ✓ User interface interactivity: JavaScript is used to fill web page data dynamically.
- ✓ Rapid development: JavaScript is easy and flexible for developers.

❖ Bootstraps

- ✓ To develop a responsive website.
- ✓ Makes creating accessible sites easier.
- ✓ Increases the semantic value of the web page.

1.7.5.2. Back-end technologies

❖ Python Programming Language

It is an easy-to-learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming.

It is often used as a support language for software developers, for building control and management, testing, and in many other ways. Python's feature highlights include:

- ✓ **Easy-to-learn:** Python has relatively few keywords, a simple structure, and defined syntax.
- ✓ **Easy-to-read:** Python code is much more defined and visible to the eyes.
- ✓ **Easy-to-maintain:** Python's success is that its source code is fairly easy-to-maintain.
- ✓ **A broad standard library:** One of Python's greatest strengths is the bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

❖ **PostgreSQL**

It is a database used to store data and information. So, it enables data to be stored and accessed across multiple storage engines.

❖ **Django Framework**

Django is a web application framework written in the Python programming language. It is based on MVT (Model View Template) design pattern. The template is an HTML file mixed with Django Template Language (DTL). Django is very demanding due to its rapid development feature. It takes less time to build an application after collecting client requirements.

❖ **Packages/Libraries**

- ✓ Face recognition: - used for face recognition process.
- ✓ OpenCV: - used for computer vision process.
- ✓ Keras: - used for training the dataset.

1.7.5.3. Documentation and Tools

❖ **Documentation Tool**

- ✓ Microsoft word 2013: - for documentation purposes.
- ✓ MS PowerPoint 2013: for presentation purposes.

❖ **Modeling Tool**

- ✓ Edraw Max 7.9: - for designing UML diagrams.
- ✓ Wondershare EdrawMax 10.1.4:- for designing sequence diagrams.

❖ **Editor Tool**

- ✓ Visual Studio Code: - Editor for implementation.
- ✓ Google colab (colaboratory):- Editor for training the model with a dataset.

❖ **Hardware Tool**

- ✓ Computer (desktop or laptop): - documentation and implementation.
- ✓ Flash disk: - For backup and storage.

1.7.5.4. Deployment Environment

- ✓ Personal computer and Desktop: - to write, deploy and store the collected information.
- ✓ University's main Server: - for deployment purposes.

1.8. Document Organization

Chapter One: - This chapter contains a range of parts that are concerned with the proposed project such as the introduction of the proposed system, background of the organization, statement of the problem, the objective of the project, the feasibility study, scope and limitation, significant, beneficiary, methodology, budget, and schedule.

Chapter Two: - This chapter is mainly concerned with studying the existing system by elaborating its existing system, business rules, main functions, forms and other documents, and bottlenecks.

Chapter Three: - This chapter deals with the proposed system by deeply explaining its functional and non-functional requirements.

Chapter Four: - This chapter deals with the system model, object model, and dynamic model of the proposed system.

Chapter Five: - This chapter deals with the design goals of the proposed system, the proposed system's architecture (including subsystem decomposition, hardware, and software mapping), algorithm design, and user interface of the proposed system.

Chapter Six: - This chapter deals with the implementation of the database, class diagram, application server, application security, user interface, system testing, and model development

Chapter Seven: - This chapter deals with the conclusion and recommendation of the system.

CHAPTER TWO

2. DESCRIPTION OF THE EXISTING SYSTEM

2.1. Introduction of Existing System

Cafeteria Management System is a system used to manage the activities that take place at the Cafeteria Foodservice at University. The current mechanism for handling the activity of the Cafeteria of Wolkite University is using digital ID to control the senior students and is using the manual book to control fresh students. Both modes of controlling and managing students have their drawbacks.

Thus, Cafeteria Management System that operates manually using manual books is prone to human mistakes moreover it is time-consuming, resource-wasting. In addition to these, it prevents a student, who forgot and lost his/ her manual book from getting food service from the cafeteria. The second mode, the digital ID-based system is an automated system but it didn't control and manage the students securely and it prevents a student, who forgot and lost his/her digital ID. So he/she didn't get food service from the cafeteria. In the following categories, we enumerated both systems in detail.

2.1.1 Manual book-based cafeteria management system

A manual book-based cafeteria management system is controlling the students when they are getting food service from the cafeteria manually. In a manual book-based cafeteria management system, a manual book is generated by the university registrar and gets students' information from the SIMS (Student Information Management System).

In a manual book-based cafeteria management system, the thicker thick or mark the student's manual books at each period time of food service that given by cafeteria at cafeteria entry.

2.1.2 Digital ID-based cafeteria management system

A digital ID-based cafeteria management system is an automated system to control and manage the students who get food service from the cafeteria. In a digital ID-based cafeteria management system, a digital ID is generated by the university registrar and gets students' information from the SIMS (Student Information Management System).

In a digital ID-based cafeteria management system, the thicker makes the system read the student's digital ID barcode by barcode reader at each period time of food service given by the cafeteria at the cafeteria entry.

2.2. Users of Existing Cafeteria Management System

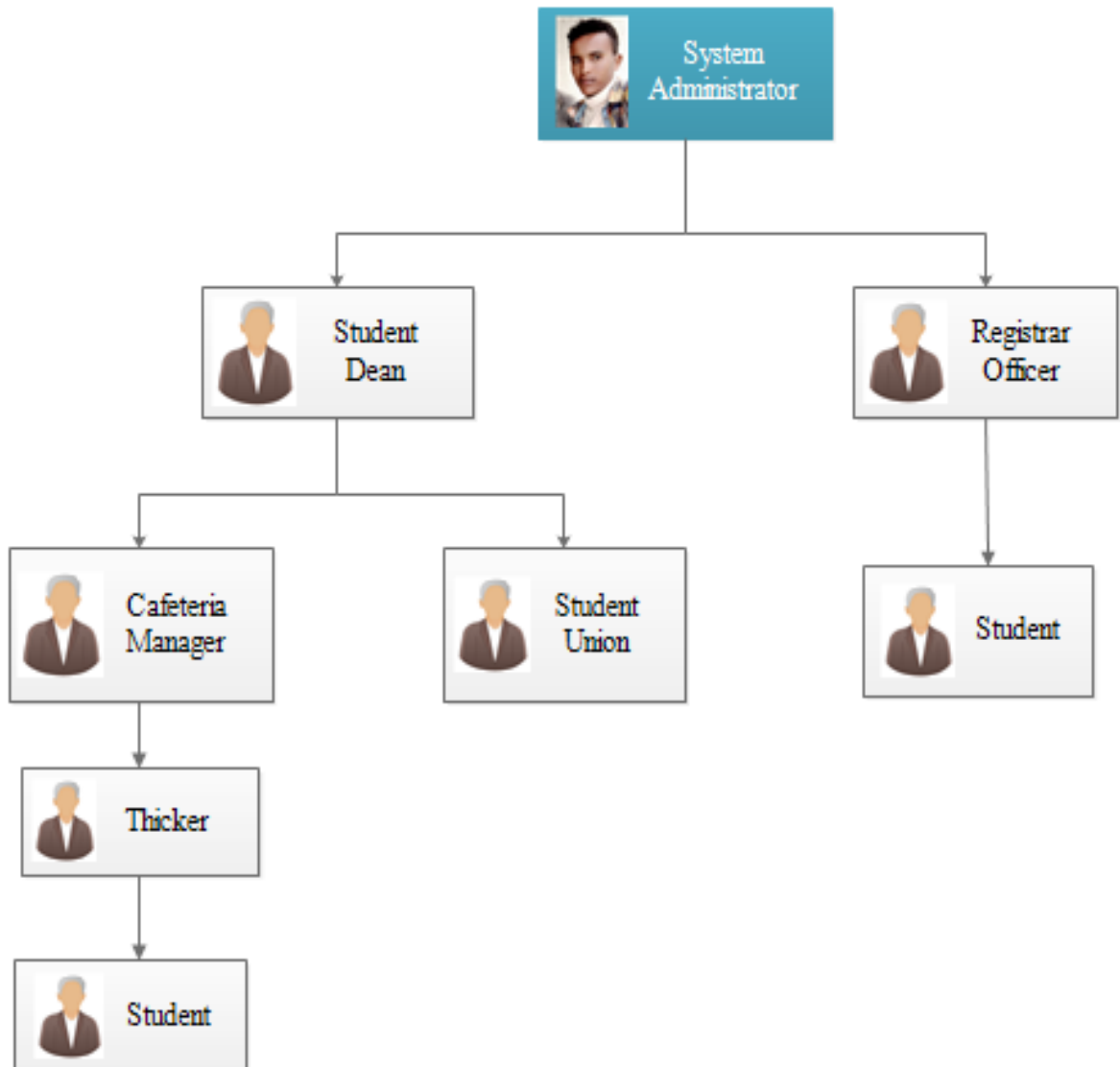


Figure 2. 1 User structure of digital ID-based system

The main users and their responsibility of the existing system: -

Table 2. 1 User of the existing system

Users	Responsibility
System Administrator	✓ A person who adds the user of the system.
Cafeteria Manager	✓ A person who manages the food, material. ✓ A person who views the report
Thicker	✓ A person who controls the students at the cafeteria entry. ✓ A person who thick the student's digital ID.
Student Dean	✓ A person who views the report
Registrar Officer	✓ A person who manages the students.
Student Union	✓ A student who approves the food and material assigned for the student.
Student	✓ A student who holds and ready the ID for thick or mark.

2.3. Major Functions of the Existing System

2.3.1 Major function of the Manual book based system

1. Generate the manual book

The manual book is prepared or generated from SIMS and printed by the registrar officer then would give to the student for getting food service from the cafeteria.

2. Marking the manual book

The Thicker is marking the student's manual book at each period time of the food service and controlling the students at the cafeteria entry.

3. View the manual book

The Thicker views the student's manual book to easily control the students at cafeteria entry.

2.3.2 Major function of the Digital ID-based system

1. User management

The System Administrator adds, updates, views, deletes, activates and deactivates the user of the system.

2. Student management

The Registrar Officer registers, updates, views, deletes, and manages the student's information.

3. Generate the digital ID

The digital ID is prepared or generated from SIMS and printed by the registrar officer then would give to the student for getting food service from the cafeteria.

4. Reading the digital ID

The Thicker makes the barcode reader of the system read the student's digital ID and control the student who gets food service from the cafeteria at cafeteria entry.

5. Report management

The Cafeteria Manager generates the report about the material and food then Student Dean views the report generated by the Cafeteria Manager.

6. View student information

The Thicker views the student's information when they are making read their digital ID and controls the student who gets food service from the cafeteria at cafeteria entry.

2.4. Forms and Other Documents of the Existing Systems

Manual books of the fresh student.

The image shows an open manual book with two pages. The left page is titled 'ወር ታህሳስ' (Month of the Year) and the right page is titled 'ወር ጥር' (Month of the Year). Both pages contain a table with columns for Day, B, L, and D, and rows numbered 1 to 15. The tables are filled with blue and red checkmarks and lines, indicating attendance records. There is also handwritten text in Amharic and English on the right page, including 'የተማሪዎች ምክርቤት ሰነድ' (Student Leader Certificate) and 'Student Leader'.

Figure 2. 2 Manual Book form

The digital ID of the senior student.

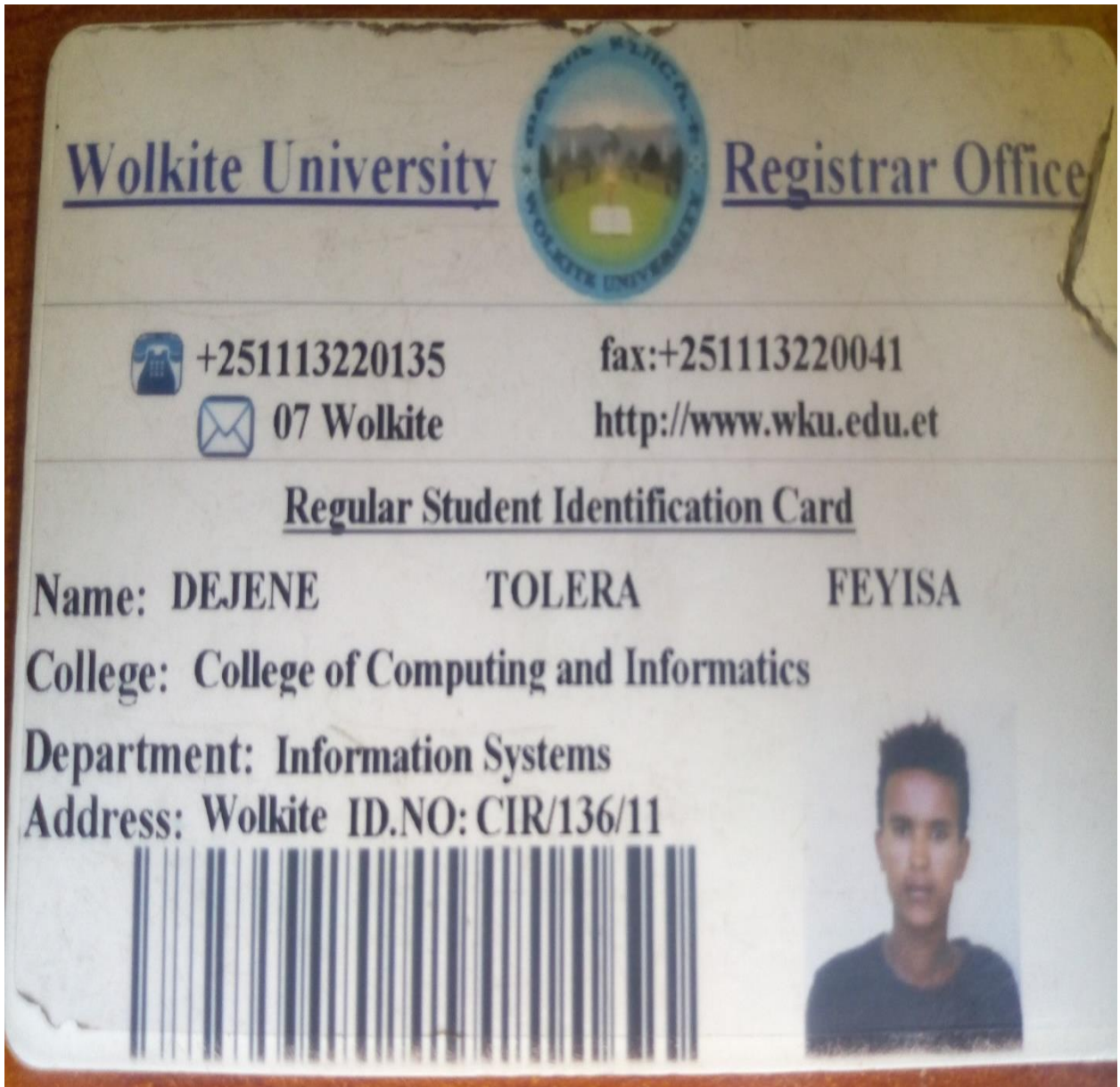


Figure 2. 3 Digital ID form

2.5. Drawbacks of the Existing System

We categorized the drawbacks of the existing cafeteria management system as the bottlenecks of the manual and automated system.

2.5.1. The drawback of the manual book-based system

The drawback of the manual book based cafeteria management system:

- ✓ Time consume and tedious.
- ✓ There is a chance a student gets food service from the cafeteria with another student's manual book.
- ✓ If the student lost or forgot his or her manual book, he/she couldn't get food service from the cafeteria.
- ✓ If the student lost his/her manual book, he/ she must pay the amount of birr to get the new manual book.
- ✓ A student may hold a double manual book.
- ✓ The cost used to prepare the manual book is increasing as the number of students is increasing.
- ✓ Difficult to report the number of students who are getting or not getting food service from the cafeteria.
- ✓ Generally, difficult to manage the student by the manual book.

2.5.2 The drawback of the digital ID-based system

The drawback of the digital ID-based cafeteria management system:

- ✓ The student gets food service from the cafeteria by another student's digital ID because the system checks only the student's digital ID is authorized or not.
- ✓ If the student lost or forgot his/her digital ID, he or she couldn't get food service from the cafeteria.
- ✓ If the student lost or forgot his/her digital ID, he or she must pay the amount of birr to get the new digital ID.
- ✓ A student holds a double digital ID.
- ✓ The cost used to prepare the digital ID because from time to time the number of students is increasing as the University is growing fast.

2.6. Business Rules of the Existing System

The following are some business rules for the cafeteria management system at Wolkite University.

BR1:- The Students must be registered for getting food service from the cafeteria by the registrar officer and generate the identification number they would give for the student.

BR2:- The Student must have an identification card to get food service from the cafeteria.

BR3:- The food service should be given weekly at the regular time only (for Breakfast from 1:00 to 2:30, for Lunch from 5:30 to 7:30, for Dinner from 11:30 to 1:00).

BR4:- The Student Dean should make eligible the student to get food service and ineligible a student to prevent from getting food service.

BR5:- The Student must be made ready his/her digital ID for the barcode reader and the student must be given his or her manual book to the thicker or marker.

BR6:- The Thicker should control the students when they make read their digital ID and when they make mark their manual book.

BR7:- The System Administrator, Cafeteria Manager, Thicker, Registrar Officer, and Student Dean can only view and do his or her role that is given to him or her.

BR8:- The Student cannot wear the political and religion t-shirt or any cloth at the cafeteria.

CHAPTER THREE

3. PROPOSED SYSTEM

Cafeteria management system using face recognition is a machine learning-based system that handles and controls the students using the web application and deep learning algorithms/models. Face recognition is a broad problem of identifying or verifying people in photographs and videos.

To develop our proposed face recognition system, we would use a deep learning approach which is currently a state-of-the-art technology. Some of the main components of the proposed system are recognizing the Student's face and the permitting of getting food service for the recognized face after the students are registered by the university registrar officer then Admin takes the student's face images and training the model by taken images.

Then, the Student Controller initiates the system to recognize and identify the students at cafeteria entry. And also, the Admin manages a user of the system and manages feedback, Student Dean and Cafeteria Manager view the report generated by the system.

3.1. Functional Requirements

The functional requirements are features that are needed to satisfy the users, the proposed system would be designed according to the roles of the users. For a better understanding of our system, we divided the functionalities of our system by modules. The proposed system is expected to provide the following services.

3.1.1. System Administrator Module

- ✓ **Add user:** - The system should allow the Admin to add a new user's (Cafeteria Manager, Student Dean, Student Controller, Registrar Officer, Student Union, and Admin) information.
- ✓ **View user:** - The system should allow the Admin to view the user's (Cafeteria Manager, Student Dean, Student Controller, Registrar Officer, Student Union, and Admin) information.

- ✓ **Update user:** - The system should allow the Admin to update the user's (Cafeteria Manager, Student Dean, Student Controller, Registrar Officer, Student Union, and Admin) information when needed.
- ✓ **Delete user:-**The system should allow the Admin to delete the user's (Cafeteria Manager, Student Dean, Student Controller, Registrar Officer, Student Union, and Admin) information when they leave the university and stop their job at the university.
- ✓ **Activate user:** - The system should allow the Admin to activate the user's (Cafeteria Manager, Student Dean, Student Controller, Registrar Officer, Student Union, and Admin) information when the user comes back from leaves by different case.
- ✓ **Deactivate user:** - The system should allow the Admin to deactivate the user's (Cafeteria Manager, Student Dean, Student Controller, Registrar Officer, Student Union, and Admin) information when user is forbidden to access system due to a discipline case.
- ✓ **Manage user role:** - The system should allow the Admin to manage, assign, add, update, and delete the role of the user for the use of the system.
- ✓ **Take student's image:** - The system should allow the Admin to capture or take the student's images then generate the dataset of the student's image.
- ✓ **Train the model:** - The system should allow the Admin to train the model by the generated dataset of students' images.
- ✓ **Reply feedback:** - The system should allow the Admin to reply to the Student Dean and Registrar Officer's feedback.
- ✓ **View feedback:** - The system should allow the Admin to view the feedback sent by the Student Dean and Registrar Officer.
- ✓ **Update profile:** - The system should allow the Admin to update his or her profile.
- ✓ **Change password:** - The system should allow the Admin to change his or her password.

3.1.2. Registrar Officer Module

- ✓ **Register Student:** - The system should allow the Registrar Officer to register new students' information

- ✓ **View Student:** - The system should allow the Registrar Officer to view the student's information when needed.
- ✓ **Update Student:** - The system should allow the Registrar Officer to update the student's information when needed.
- ✓ **Delete Student:** - The system should allow the Registrar Officer to delete the student's information when leaving the university.
- ✓ **Send feedback:** - The system should allow the Registrar Officer to send feedback to the System Administrator.
- ✓ **View feedback:** - The system should allow the Registrar Officer to view the feedback that contains sent to Admin and replied feedback by Admin.
- ✓ **Update profile:** - The system should allow the Registrar Officer to update his or her profile.
- ✓ **Change password:** - The system should allow the Registrar Officer to change his or her password.

3.1.3. Student Dean Module

- ✓ **Make café student:** - The system should allow the Student Dean to make a cafe student get the food service.
- ✓ **Make non-café student:** - The system should allow the Student Dean to make a non-cafe student prevent a student from the food service.
- ✓ **View report:** - The system should allow the Student Dean to view the period (breakfast, lunch, and dinner) time and one-day report from the student status that is marked or not through face recognition.
- ✓ **Send feedback:** - The system should allow the Student Dean to send feedback to the System Administrator.
- ✓ **Reply feedback:** - The system should allow the Student Dean to reply to the Cafeteria Manager and Student Union feedback.
- ✓ **View feedback:** - The system should allow the Student Dean to view the feedback sent by Cafeteria Manager and Student Union and replied by System Administrator.
- ✓ **Update profile:** - The system should allow the Student Dean to update his or her profile.

- ✓ **Change password:** - The system should allow the Student Dean to change his or her password.

3.1.4. Cafeteria Manager Module

- ✓ **View report:** - The system should allow the Cafeteria Manager to view the period (breakfast, lunch, and dinner) time and one-day report from the student status that is marked or not through face recognition.
- ✓ **View student status:** - The system should allow the Cafeteria Manager to view the student status that is marked or not through face recognition.
- ✓ **Send feedback:** The system should allow the Cafeteria Manager to send feedback to the Student Dean.
- ✓ **Reply feedback:** The system should allow the Cafeteria Manager to reply to Student Controller and Student Union feedback.
- ✓ **View feedback:** - The system should allow the Cafeteria Manager to view the feedback sent by Student Controller and Student Union and replied to by Student Dean.
- ✓ **Update profile:** - The system should allow the Cafeteria Manager to update his or her profile.
- ✓ **Change password:** - The system should allow the Cafeteria Manager to change his or her password.

3.1.5. Student Controller Module

- ✓ **Initiate system:** - The system should allow the Student Controller to initiate the system for student's face detection and recognition at each period (breakfast, lunch, and dinner) time.
- ✓ **Cease system:** - The system should allow the Student Controller to cease the system after the time of each period (breakfast, lunch, and dinner) is end.
- ✓ **View student status:** - The system should allow the Student Controller to view the student status that is marked or not through face recognition.
- ✓ **Mark student status for twine:** - The system should allow the Student Controller to mark or fill form of the student status for twine students because are not accurately recognized through face recognition process.

- ✓ **Send feedback:** The system should allow the Student Controller to send feedback to the Cafeteria Manager.
- ✓ **View feedback:** - The system should allow the Student Controller to view the feedback that contain sent to Cafeteria Manager and replied to by Cafeteria Manager.
- ✓ **Update profile:** - The system should allow the Student Controller to update his or her profile.
- ✓ **Change password:** - The system should allow the Student Controller to change his or her password.

3.1.6. Student Union Module

- ✓ **Send feedback:** The system should allow the Student Controller to send feedback to the Cafeteria Manager and Student Dean.
- ✓ **View feedback:** - The system should allow the Student Controller to view the feedback that contain sent to Cafeteria Manager and Student Dean, and replied by them.
- ✓ **Update profile:** - The system should allow the Student Controller to update his or her profile.
- ✓ **Change password:** - The system should allow the Student Controller to change his or her password.

3.1.7. Face recognition process

A. Pre-Processing of Face Images/ Dataset Creation

Pre-Processing of Face Images is the process of categorizing and normalizing the dataset (student's face images) to train a machine learning model. This step is to pre-process the extracted faces to make the training and classifying phase of the model easier and improve the probability to recognize a person correctly. The proposed model for facial recognition requires the same size for the entire training set and classification purpose.

Dataset of students would be created before the recognition process. The dataset is created by taking the student's face/photo by the Admin. The dataset would be created only to train this system. Whenever we register students' data and images in our system to create a dataset, deep learning applies to each face to compute 128-d facial features and store them in a student face

data file to recall that face in the recognition process. This process is applied to each image taken during registration.

This phase of the proposed system would perform the preprocessing of face images or the creating of the dataset to make the dataset fit with the model and improve the accuracy of the face recognition. We would develop this part of the proposed system mainly employing different kinds of tools provided by the Keras deep learning framework, and other deep learning (machine learning) libraries.

B. Train the model

After a dataset of students' images was generated then easy to train the model using the dataset of images. For training a model we initially split the model into 3 three sections which are 'Training data', 'Validation data', and 'Testing data'. Train the classifier using 'training data set', tune the parameters using 'validation set' and then test the performance of our model on unseen 'test data set'. During training the model the training and validation set is available. The test data set must not be used during training the model. The test set would only be available during testing the model.

Training set: The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning is to fit the parameters of the classifier.

Test set: A set of unseen data used only to assess the performance of a fully-specified classifier. After setting up the dataset would train and evaluate our model in the Google CoLab.

C. Face detection and extraction

Face detection is the non-trivial first step in face recognition. It is the process of detecting all human faces in an image or video and extracting (cropping) these faces. To carry out image classification, the Convolutional Neural Networks (CNN) algorithm can carry out face detection and feature extraction by using Multi-Task Convolutional Neural Networks (MTCNN) [2]. In the specific case of image recognition, the features are the groups of pixels, like edges and points, of an object that the network is analyzed for patterns [3].

D. Face recognition

Face Recognition is a technique that is used to recognize a student from his/her face that has been previously trained model from a dataset. The proposed system would use a Convolutional Neural Network (CNN) algorithm. CNN is a deep learning architecture that is mostly known as **ConvNets** and was developed in 1998 by Yann LeCun. A CNN consists of neurons or nodes that have learnable weights and biases. Every node performs a convolution operation (dot product with filters) on the received input, which may be followed with a nonlinearity. Three kinds of layers build up a CNN: convolution layers, pooling layers, and fully connected layers (same as regular neural networks). The first layer is **Convolutional Layer** consists of Rectified Linear Unit (ReLU) that outlasts to rectify the feature map. The second layer is **Pooling layer** is used to rectify these feature maps into the next feed. Pooling is generally a sampling algorithm that is down-sampled and it reduces the dimensions of the feature map. Later, the result generated consists of 2-D arrays consisting of single, long, continuous, and linear vectors flattened in the map. The last layer is called **Fully Connected Layer** which forms the flattened matrix or 2-D array fetched from the Pooling Layer as input and identifies the image by classifying it. These are stacked together to form a CNN or ConvNet. CNN's have applications in several areas like image recognition, image classification, video recognition, and natural language processing (NLP). The major advantage of CNNs is that the preprocessing required is much less as compared to other similar deep learning networks.

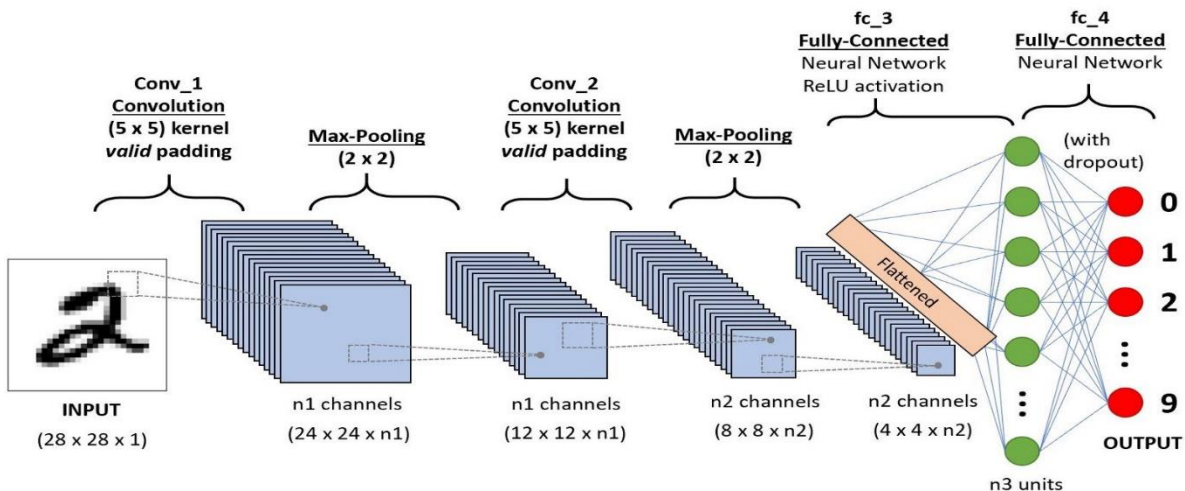


Figure 3. 1 Convolutional Neural Network (CNN) Architecture

We would build the new model using convolutional neural network to compete the face recognition process. After model is built, the model is trained and used for prediction and recognition with matching. If the student's current image is matched with the existing dataset, it would move to student status marking.

E. Student status marking

After the recognition process, the student recognized is searched in the database and mark the student status if it is not marked then allow to get the food service, otherwise prevent a student from getting the food service because that student needs to get for a second time (double) at each period (breakfast, lunch, and dinner) time.

F. Generate report

After student status marking is performed through face recognition, then the system generates a report from the student status based on the user's needs (for each period time like breakfast, lunch, and dinner and one-day report).

3.2. Non-functional Requirements

Non-functional requirements describe aspects of the system that are not directly related to the functional behavior of the system, only related to how the system would do the functional requirement. It include a broad variety of requirements that apply to many different aspects of the system, from usability to performance.

3.2.1. User Interface and Human Factors

The system we are going to develop follows a good interface principle and it would provide easy, attractive, simple, and interactive interfaces for the user of our system. The user of the system should have a computer to connect to the internet and access the system.

3.2.2. Hardware Consideration

The system would be developed by considering hardware requirements. The system would support computers, and digital cameras positioned at the cafeteria entry and secondary memory to store all the images and database to provide the service for the user

Table 3. 1 Hardware consideration

NO	Items	Specification
1	Computer Brand	HP and Dell
2	Processor	Core(i5) with 2.5 GHz
3	RAM	8 GB of RAM
4	Camera	8 Megapixels
5	System Type	64-bit Operating system
6	Deployment server	Wolkite University's server and Hard disk of the Computer

3.2.3. Security Issues

The golden rule of Web application security is to never trust data from untrusted sources. Protecting user data is an essential part of any system, the proposed system would allow only authorized users to enter using their username and password that would be encrypted using the Password-Based Key Derivation Function 2 (PBKDF2) encryption algorithm.

Session and cookie would be used in our proposed system. Users would access only their privileged concerning areas of the system and data modification should be done only by privileged users.

By default, Django uses the pbkdf2 and sha256 methods to encrypt the user's password. Once get the password stored in the database table, you need to compare it with others if brute force cracking. It is recommended that you use hash table comparison.

3.2.4. Performance Consideration

The proposed system would have easy and efficient code manipulation and a clear database.

Response Time: Upon the request for a user query the system under normal conditions would display results as quickly as possible.

Processing Time: Since the system would be developed with efficient programming language and database upon request for users' activity. The system under normal conditions would process the request as quickly as possible.

Concurrent - Processing: since the proposed system is web-based it concurrently deals with the requests rather than waiting for a previous one to be completed, the system would support multiple users at a time.

3.2.5. Error Handling and Validation

Our system would check user inputs to the system to handle errors. Django provides full support for cookie- and session-based messaging, for both anonymous and authenticated users. Every message is tagged with a specific level that determines its priority (e.g., info, warning, or error). It handles and shows errors in a user-friendly manner, without exaggerating the user. For server-side error, it is only visible on the server log for the developer.

3.2.6. Quality Issues

Our system would be reliable by handling exceptions when the user uses the system and performs its function consistently. The users would be supported by a feedback mechanism in which they can give comments on the system for quality assessment. The functions of the system are appropriate, implemented correctly, and handle data securely. It would be user-friendly, so the usability of the system would be clear and understandable by users and do not require much effort.

3.2.7. Backup and Recovery

In addition to authorizing access to users, the system could have problems in such areas as data crime, disaster, and human errors. For this purpose, incremental backup copies (copies of only the change to files) and full backup copies (copies of all files) would be made and stored in a secured location.

3.2.8. Physical Environment

The system would be deployed on Wolkite University's main server and authorized users can access the system using a web browser through their computer.

3.2.9. Resource Issues

Since the proposed system is a machine learning system, it needs high computing power to behave efficiently and in a good performance. The performance of the proposed system is dependent on the computing power such as RAM, Processor, and the quality of the Camera. But, we cannot afford the quality of the Cameras currently due to budget.

CHAPTER FOUR

4. SYSTEM ANALYSIS

System analysis is the process of collecting and interacting facts, identifying the problems, and decomposition a system into its components. The analysis model contains the functional model (system model), object model, and dynamic model.

4.1. System Model

The system model is the process of developing abstract models of a system, with each model presenting a different view and perspective of the system [1].

4.1.1. Use Case Model

Table 4. 1 Use case Identification

Actor	Use Case Under Identified Actor	Use Case ID
System Administrator	Login	01
	Manage user ✓ Add user ✓ View user ✓ Update user ✓ Delete user ✓ Activate user ✓ Deactivate user	02
	Manage user role	03
	Take Student's Image	04
	Train Model	05

	Manage Feedback ✓ View Feedback ✓ Reply Feedback	06
	Update Profile	07
	Change Password	08
	Logout	09
Student Dean	Login	01
	Manage Cafe Student ✓ Make cafe Student ✓ Make non-Cafe Student	10
	View Report	11
	Manage Feedback ✓ Send Feedback ✓ View Feedback ✓ Reply Feedback	06
	Update Profile	07
	Change Password	08
	Logout	09
Registrar Officer	Login	01
	Manage Student	12

	<ul style="list-style-type: none"> ✓ Register Student ✓ View Student ✓ Update Student ✓ Delete Student 	
	Manage Feedback <ul style="list-style-type: none"> ✓ View Feedback ✓ Send Feedback 	06
	Update Profile	07
	Change Password	08
	Logout	09
Cafeteria Manager	Login	01
	View Student Status	13
	View Report	11
	Manage Feedback <ul style="list-style-type: none"> ✓ View Feedback ✓ Send Feedback ✓ Reply Feedback 	06
	Update Profile	07
	Change Password	08
	Logout	09

Student Controller	Login	01
	Manage System <ul style="list-style-type: none"> ✓ Initiate System ✓ Cease System 	14
	View Student Status	13
	Manage Feedback <ul style="list-style-type: none"> ✓ Send Feedback ✓ View Feedback 	06
	Mark student status for twine	15
	Update Profile	07
	Change Password	08
	Logout	09
Student Union	Login	01
	Manage Feedback <ul style="list-style-type: none"> ✓ Send Feedback ✓ View Feedback 	06
	Update Profile	07
	Change Password	08
	Logout	09

4.1.1.1. Use Case Diagram

We have used use case diagrams to capture the functional requirements of the system and it is the functionality of the system or the service provided by the system.

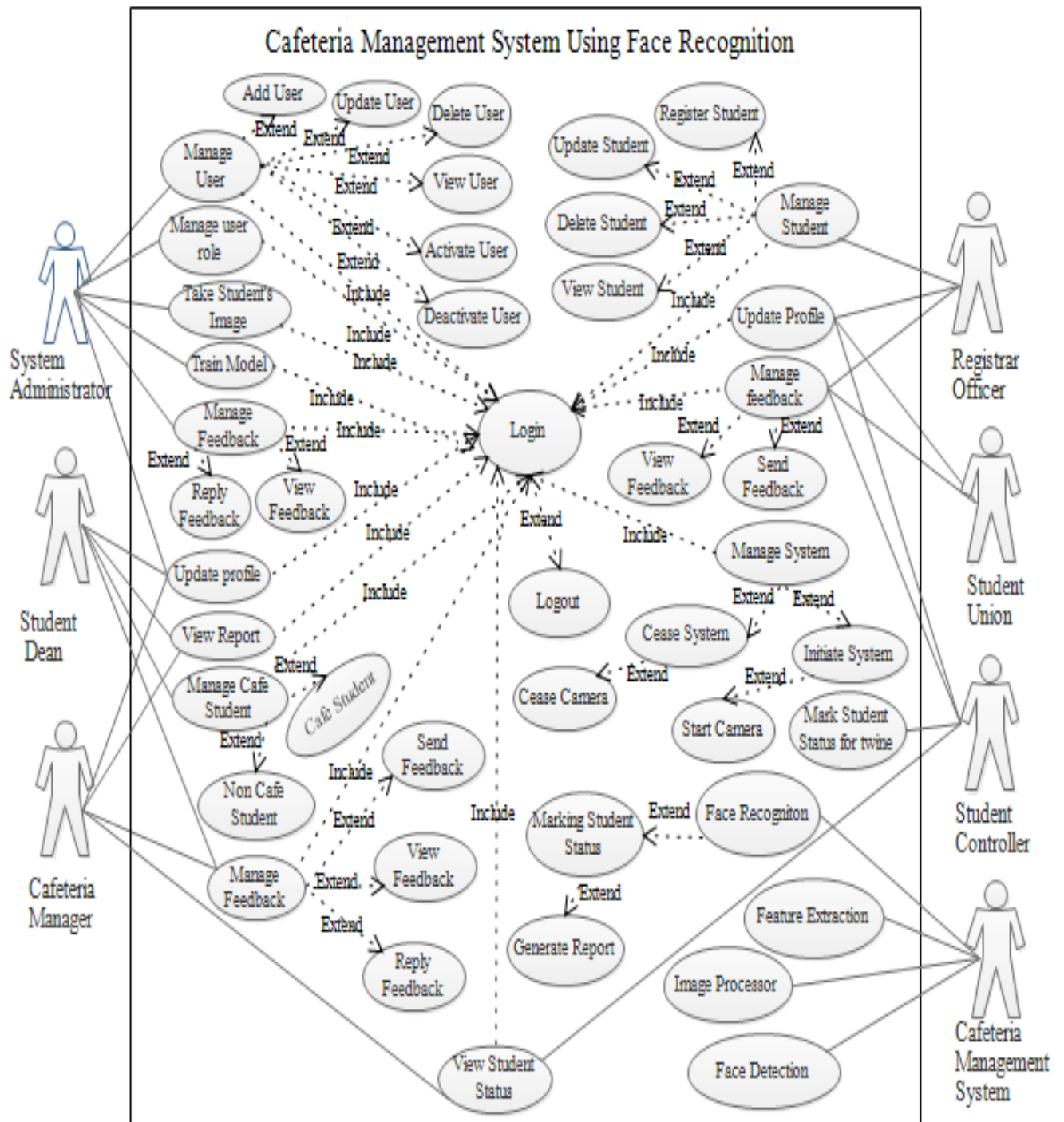


Figure 4. 1 Use case diagram

4.1.1.2. Use Case Description

The use case description is used to detail the description of the use case and how the use case works to perform user and system functionality. The following tables show Use case descriptions based on the system business or context of our system.

Table 4. 2 Use Case Description of Login

Use Case Name	Login	
Use case Identifier	UCID-01	
Actors	System Administrator, Cafeteria Manager, Student Dean, Student Controller, Student Union, and Registrar Officer.	
Description	Allows user's login into the system and access their privilege in the system.	
Precondition	The users must have a username and password (user account).	
Basic Course Of Action	Actor Action	Cafeteria Management System Responses
	<p>Step1: Open the system web page.</p> <p>Step3: Click the Login on the home page.</p> <p>Step5: Enter username and password then click the login button.</p> <p>Step8: End use case.</p>	<p>Step2: Display the home page.</p> <p>Step4: Display the login interface for use.</p> <p>Step6: Validate username and password.</p> <p>Step7: Displays a page with the privileges of a user.</p>

Alternative Course of Action	<p>Step6.1: If the input data (username and password) is invalid.</p> <ul style="list-style-type: none"> I. The system displays an error message II. The User returned to Step5.
Post Condition	The user logged into the system (system display the user's privilege).

Table 4. 3 Use Case Description of Add User

Use Case Name	Add User	
Use case Identifier	UCID-02	
Actor	System Administrator	
Description	Allow System Administrator to add a new user.	
Precondition	The System Administrator must log in to the system.	
Basic Course Of Action	Actor Action	Cafeteria Management System Responses
	<p>Step1: Click on manage user.</p> <p>Step3: Click on add user.</p> <p>Step5: Admin fills out the form and clicks on add user button.</p> <p>Step9: End use case.</p>	<p>Step2: The system displays the management option.</p> <p>Step4: The system displays the user add form.</p> <p>Step6: The System checks input data.</p> <p>Step7: The System adds user data to the database.</p>

		Step8: The System displays a successful message.
Alternative Course of Action	Step6.1: If there is invalid data filled by System Administrator <ul style="list-style-type: none"> I. The system displays a message that contains “you entered wrong data.” II. The System Administrator returned to Step5. Step7.1: If the user already exists. <ul style="list-style-type: none"> I. The system displays a message that contains “the user is already added.” II. The System Administrator returned to Step5. 	
Post Condition	The user was successfully added to the database.	

Table 4. 4 Use Case Description of Register Student

Use Case Name	Register Student	
Use case Identifier	UCID-12	
Actor	Registrar Officer.	
Description	Allow Registrar Officer to register students.	
Precondition	The Registrar Officer must log in to the system.	
Basic Course Of Action	Actor Action	Cafeteria Management System Responses

	<p>Step1: Click on manage student.</p> <p>Step3: Click on register student.</p> <p>Step5: Registrar Officer fills out the form and clicks on the Register button.</p> <p>Step9: End use case.</p>	<p>Step2: The system displays the management option.</p> <p>Step4: The system displays the registration form.</p> <p>Step6: The System checks input data.</p> <p>Step7: The System adds Student data to the database.</p> <p>Step8: The System displays a successful message.</p>
Alternative Course of Action	<p>Step6.1: If there is invalid data filled by a Registrar Officer</p> <ul style="list-style-type: none"> I. The system displays a message that contains “you entered wrong data.” II. The registrar returned to Step5. <p>Step6.1: If the student already exists.</p> <ul style="list-style-type: none"> I. The system displays a message that contains “the student is already registered.” II. The registrar returned to Step5. 	
Post Condition	The student was successfully registered to the database.	

Table 4. 5 Use Case Description of Taking Image

Use Case Name	Take Student Image	
Use case Identifier	UCID-04	
Actor	System Administrator	
Description	Allow System Administrator to take student’s image.	
Precondition	The System Administrator must log in to the system.	
Basic Course Of Action	Actor Action	Cafeteria Management System Responses
	<p>Step1: Click on Take Image</p> <p>Step3: Admin fills label and clicks on the Capture Image button.</p> <p>Step8: End use case.</p>	<p>Step2: The system displays the page for taking the student’s image.</p> <p>Step4: The System checks input data.</p> <p>Step5: The System performs a face detection algorithm and preprocesses the captured image.</p> <p>Step6: The System store the captured image in the dataset.</p> <p>Step7: The System displays a successful message.</p>
Alternative Course of Action	<p>Step4.1: If Admin enters an incorrect label or incorrect image type.</p> <p>I. The system displays a message that contains “you entered incorrect label or incorrect image type.”</p> <p>II. The Admin returned to Step3.</p>	
Post Condition	The student’s image was successfully captured.	

Table 4.6 Use Case Description of Initiate System

Use Case Name	Initiate System	
Use case Identifier	UCID-14	
Actor	Student Controller	
Description	Allow Student Controller to initiate a system for the recognition process.	
Precondition	The Student Controller must log in to the system.	
Basic Course Of Action	Actor Action	Cafeteria Management System Responses
	<p>Step1: Click on the manage system.</p> <p>Step3: Click on the initiate system link.</p> <p>Step5: Student Controller fills out the form and clicks the Initiate System button.</p> <p>Step9: End use case.</p>	<p>Step2: The system displays the needed option.</p> <p>Step4: The system displays the initiate system page.</p> <p>Step6: The System checks input data.</p> <p>Step7: The System is open to the camera automatically.</p> <p>Step8: The System displays a successful message.</p>
Alternative Course of Action	<p>Step6.1: If the Student Controller enters incorrect data.</p> <p>I. The system displays a message that contains “you entered incorrect data”.</p>	

	II. The Student Controller returned to Step5 .
Post Condition	The system was initiated successfully.

Table 4. 6 Use Case Description of View Report.

Use Case Name	View Report	
Use case Identifier	UCID-11	
Actor	Cafeteria Manager and Student Dean	
Description	Allow the Cafeteria Manager and Student Dean to view the report.	
Precondition	The Cafeteria Manager and Student Dean are must log in to the system.	
Basic Course Of Action	Actor Action	Cafeteria Management System Responses
	<p>Step1: Click on the View report link.</p> <p>Step3: Click on the report option (period report, and one day report) link.</p> <p>Step5: Select and click on the view report button.</p> <p>Step7: End use case.</p>	<p>Step2: The system displays the report option.</p> <p>Step4: The system displays a selection menu for clicked report option (criteria).</p> <p>Step6: The system displays a report preview.</p>

Alternative Course of Action	There is not any alternative course of action under this use case.
Post Condition	View organized reports for the needed purpose.

Table 4. 7 Use Case Description of View Feedback

Use Case Name	View Feedback	
Use case Identifier	UCID-06	
Actor	System Administrator, Cafeteria Manager, Student Dean, Student Controller, Student Union, and Registrar Officer.	
Description	Allow users to view the replied feedback.	
Precondition	The users must log in to the system.	
Basic Course Of Action	Actor Action	Cafeteria Management System
	<p>Step1: Click on manage Feedback.</p> <p>Step3: Click on the view feedback link.</p> <p>Step5: End use case.</p>	<p>Step2: The system displays the management option.</p> <p>Step4: The system displays the replied feedback.</p>

Alternative Course of Action	There is not any alternative course of action under this use case.
Post Condition	The feedback was successfully viewed.

Table 4. 8 Use Case Description of Log out.

Use Case Name	Logout	
Use case Identifier	UCID-09	
Actors	System Administrator, Cafeteria Manager, Student Dean, Student Controller, Student Union, and Registrar Officer.	
Description	Allow users to log out from the system.	
Precondition	The user must log in to the system.	
Basic Course Of Action	Actor Action	Cafeteria Management System Responses
	<p>Step1: User click on the profile menu.</p> <p>Step3: User clicks on the logout link.</p> <p>Step5: End use case.</p>	<p>Step2: The system displays the profile option.</p> <p>Step4: The system exits the user from the system.</p>

Alternative Course of Action	There is not any alternative course of action under this use case.
Post Condition	The user was logout or exited from the system.

4.1.1.3. Use case Scenario

Use case scenario describes the real-world examples of how the user of the system interacts with the system. It describes the steps and action events that take place to interact with the system [1].

Scenario name: Login

Participating actor: Mr. User

To log in to the system, first, the user opens the web page. The system displays the home page. Then User clicks the login link on the home page. The system displays the login page for the User then User enters the username and password then clicks the login button. The system validates username and password. If the username and password are invalid, the system displays the message (password and username are incorrect). If the input is valid, the system displays a page with the privileges of a user then use case end.

Scenario name: Register Student

Participating actor: Mr. Dejene

To register a student, first, Dejene opens the web page. The system displays the home page. Then Dejene logs into the system. Then the system displays manage student and click manage student. The system displays the management option. Then Dejene clicks the register student link. The system displays the Student Registration page. Then Dejene fills the data required by fields then clicks the Register button. The System checks input Data. Then if there is invalid data filled by a Dejene the system displays error messages or if the student already exists. The

system displays error messages or displays the student already exists message. If data filled by Dejene is valid and the System checks Database. Then if the student already exists, the system displays the student already exists message. Else the System adds Student data to the database. Finally, the student is registered and the system display student registered successfully message then use case end.

Scenario name: Take Student's Image

Participating actor: Mr. Dendea

To take the image, first, Dendea opens the web page. The system displays the home page. Then Dendea logs into the system. Then the system displays a take image page. Then Dendea clicks the take image link. The system displays the take image page. Then Dendea fills the data required by fields then clicks the capture image button. The System checks input Data. Then if there is invalid data filled by a Dendea or incorrect image type. The system displays error messages or incorrect image type messages. If data filled by Dendea is valid and the image with this data, The System performs a face detection algorithm and preprocesses the captured image, and store it in the dataset. Finally, the student's image is captured and the system displays the student's image captured successfully message then use case end.

Scenario name: Initiate System

Participating actor: Mr. Dheresa

To initiate the system, first, Dheresa opens the web page. The system displays the home page. Then Dheresa logs into the system Then the system displays a manage system and clicks manage system. The system displays the management option. Then Dheresa clicks the initiate system link. The system displays the initiate system page. Then Dheresa fills the data required by fields then clicks the initiate system button. The System checks input Data. Then if there is invalid data filled by a Dheresa, the system displays error messages. If data filled by Dheresa is valid, The System is open to the camera automatically. Finally, the system is initiated and the system display system initiated successfully message then use case end.

4.2. Object Model

An object model is a description of an object-oriented architecture, including the details of the object structure, and interfaces between objects [1]. It identifies the attributes and functions of each class. In this section, we discuss the object model class diagram and data dictionary [4].

4.2.1. Class Diagram

UML class diagram is the main building block of object-oriented modeling. The class diagram below shows the classes' attributes, operations, and association within our proposed system.

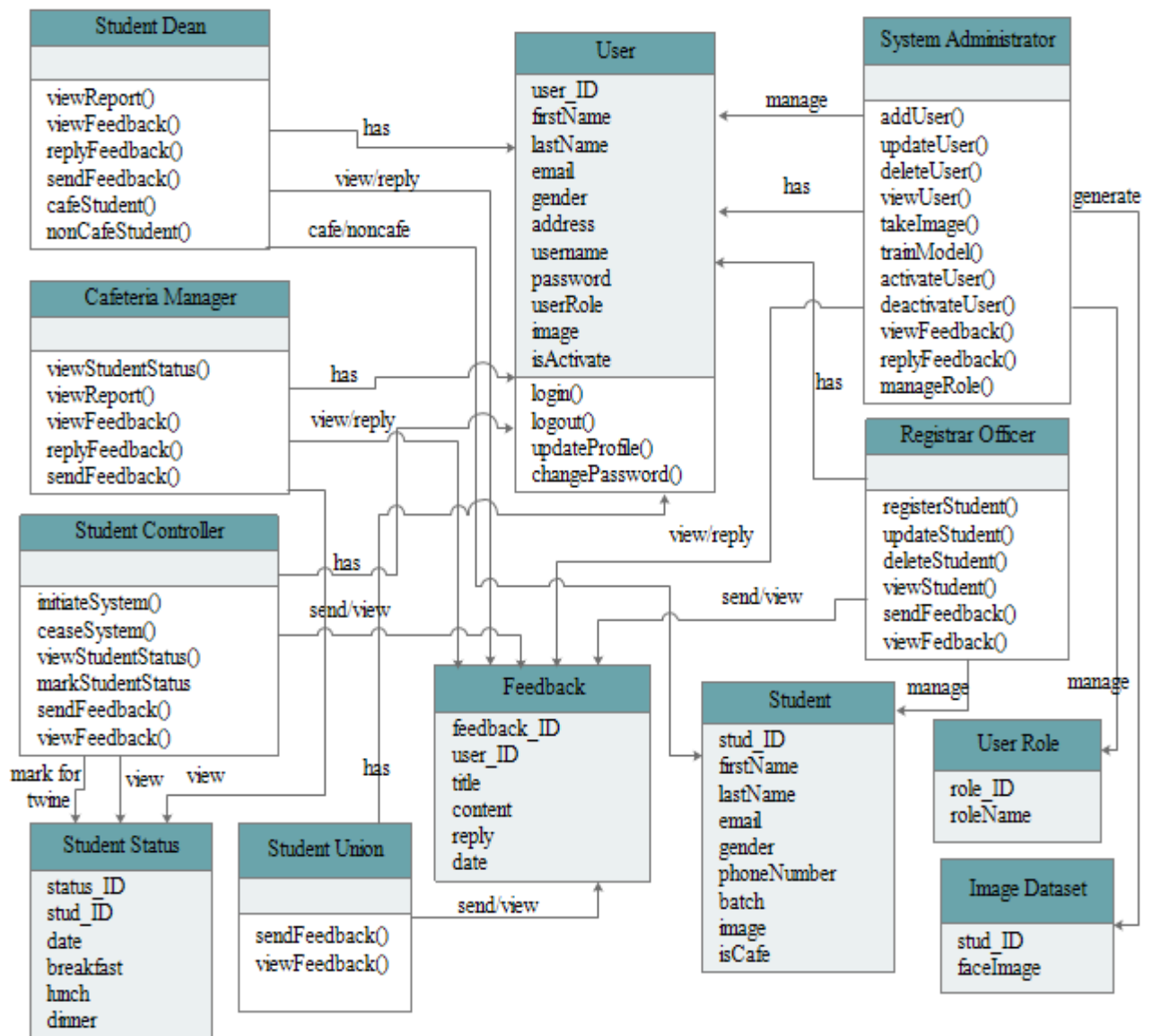


Figure 4. 2 Class diagram

4.2.2. Data Dictionary

The following table shows the attributes, data type, and key constraints of the classes in our proposed system.

Table 4. 9 Data Dictionary for User.

Attributes	Caption	Example	Data Type	Attribute Size	Constraints
user_ID	User ID	Admin/1	Varchar	20	Primary Key
firstName	First Name	Dejene	Varchar	30	Not Null
lastName	Last Name	Tolera	Varchar	30	Not Null
email	Email	dj47@gmail.com	Varchar	30	Not Null
gender	Gender	Male	Varchar	6	Not Null
address	Address	Wolkite	Varchar	30	Not Null
username	Username	dejene	Varchar	30	Not Null
password	Password	Dejene123	Varchar	30	Not Null
userRole	User Role	Admin	Varchar	30	Not Null
image	Image		Binary		Not Null
isActive	Is Active	True	Boolean	5	Not Null

Table 4. 10 Data Dictionary for Student.

Attributes	Caption	Example	Data Type	Attribute Size	Constraints
stud_ID	Stud ID	CIR/167/11	Varchar	20	Primary Key
firstName	First Name	Dendea	Varchar	30	Not Null
lastName	Last Name	Tarekegn	Varchar	30	Not Null
email	Email	dt47@gmail.com	Varchar	30	Not Null
gender	Gender	Male	Varchar	6	Not Null
phoneNumber	Phone Number	0947401089	Varchar	10	Not Null
batch	Batch	2011	Int	4	Not Null
image	Image		Binary		Not Null
isCafe	Is Active	True	Boolean	5	Not Null

Table 4. 11 Data Dictionary for Student Status.

Attributes	Caption	Example	Data Type	Attribute Size	Constraints
status_ID	Status ID	1	Int	6	Primary Key
stud_ID	Stud ID	CIR/136/11	Varchar	20	Foreign Key
date	Date	15/2/2022	Date time	10	Not Null

breakfast	Breakfast	True	Boolean	5	Null
lunch	Lunch	False	Boolean	5	Null
dinner	Dinner	True	Boolean	5	Null

Table 4. 12 Data Dictionary for Feedback.

Attributes	Caption	Example	Data Type	Attribute Size	Constraints
feedback_ID	Feedback ID	10	Int	6	Primary Key
user_ID	User ID	Manag/11	Varchar	20	Foreign Key
title	Title	Food problem	Varchar	30	Not Null
content	Content	There is no amount of food	Varchar	50	Not Null
reply	Reply	It must be solved at the moment	Varchar	50	Not Null
date	Date	15/2/2022	Date time	10	Not Null

4.3. Dynamic Model

The dynamic model represents the time-dependent aspects of a system. It is concerned with the temporal changes in the states of the objects in a system. In this section, you are expected to document the behavior of the object model, in terms of sequence, activity, and state chart diagrams [1].

4.3.1. Sequence Diagram

The sequence diagram is used to describe the dynamic behavior of several objects over time [1]. It provides a sequential map of messages passing between objects over time, how users interact with the system, and what happens internally to get the work done in the cafeteria management system using face recognition.

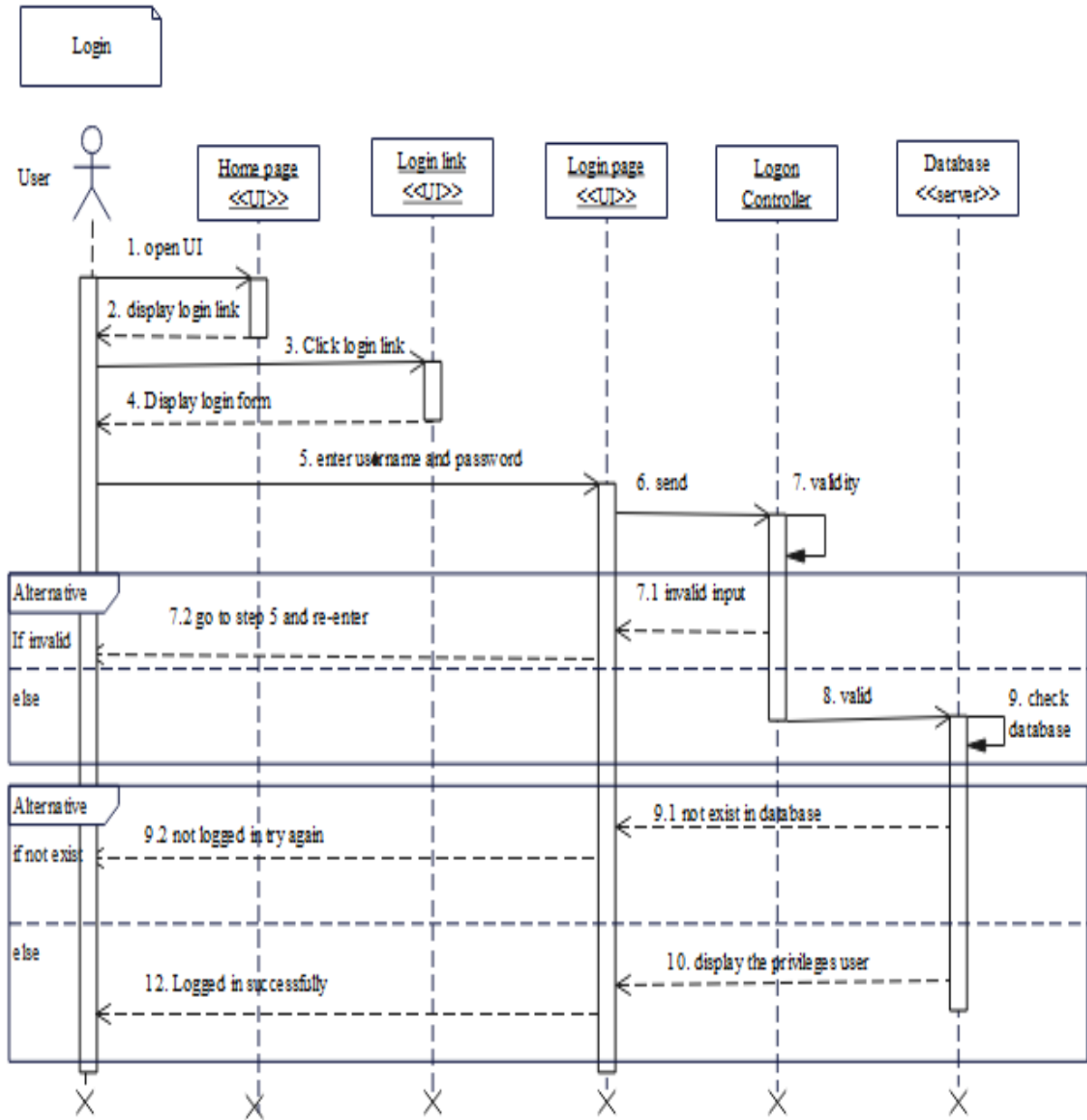


Figure 4. 3 Sequence diagram for Login

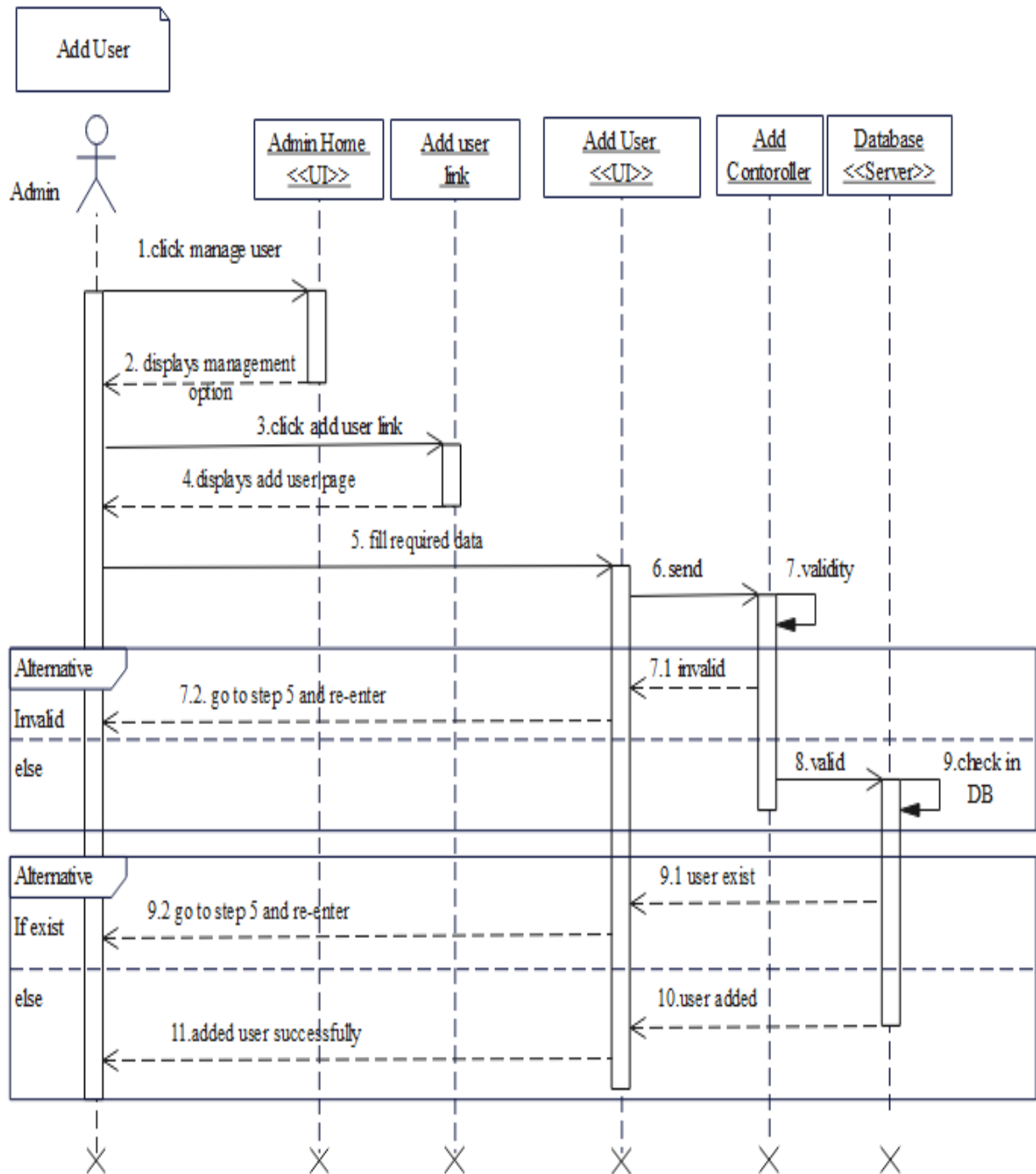


Figure 4. 4 Sequence diagram for Add User

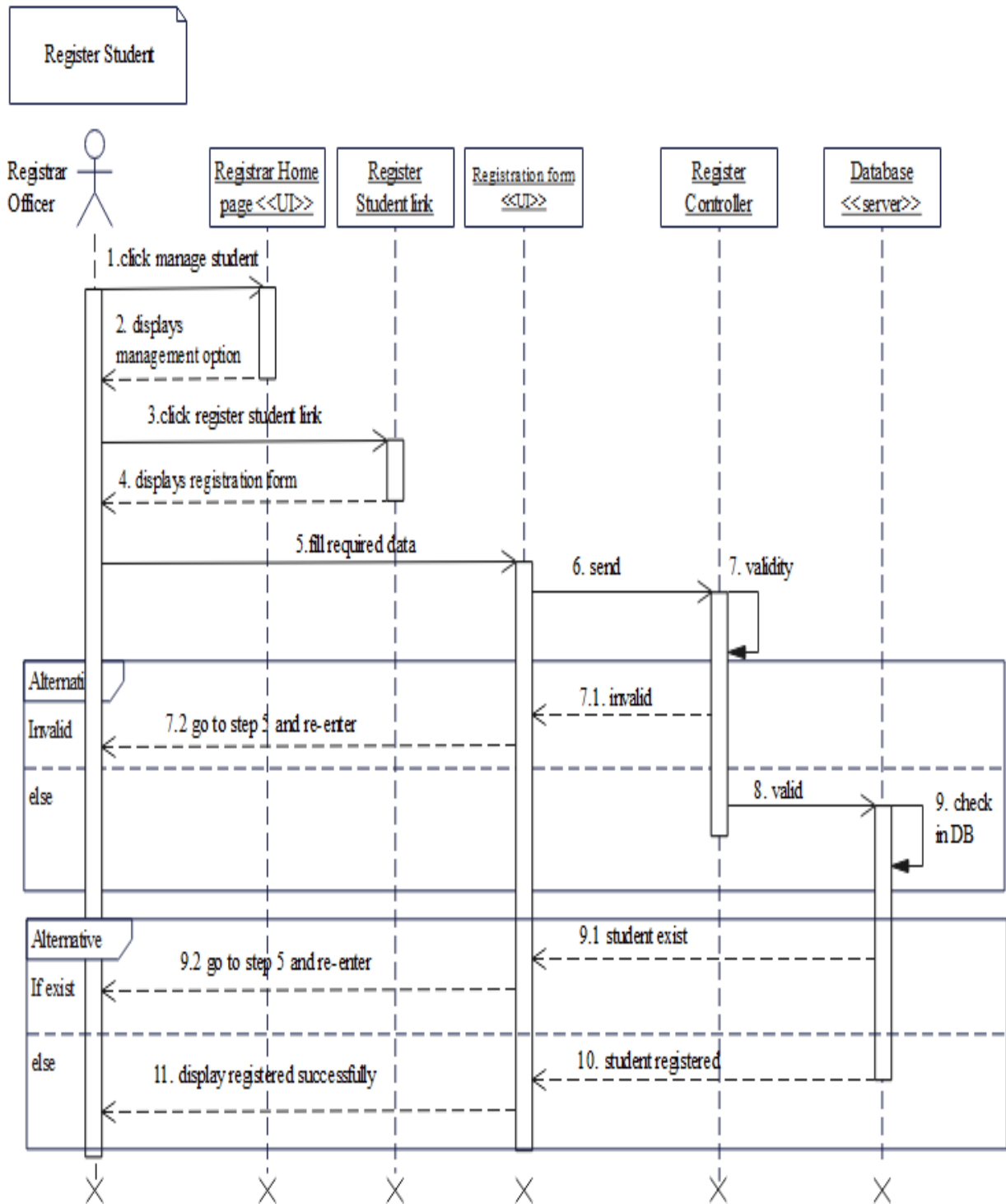


Figure 4. 5 Sequence diagram for Register Student

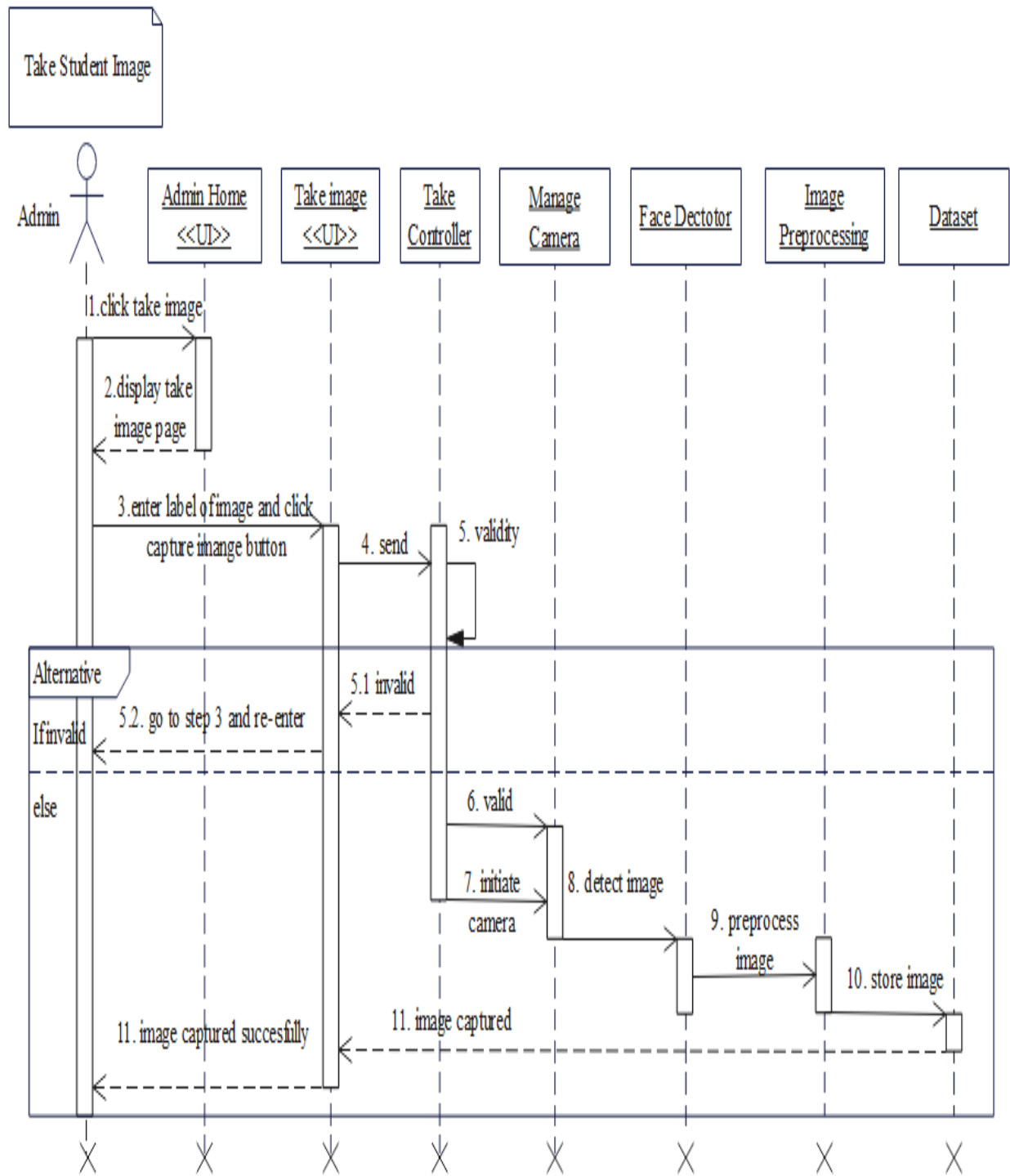


Figure 4. 6 Sequence diagram for taking Student’s Image

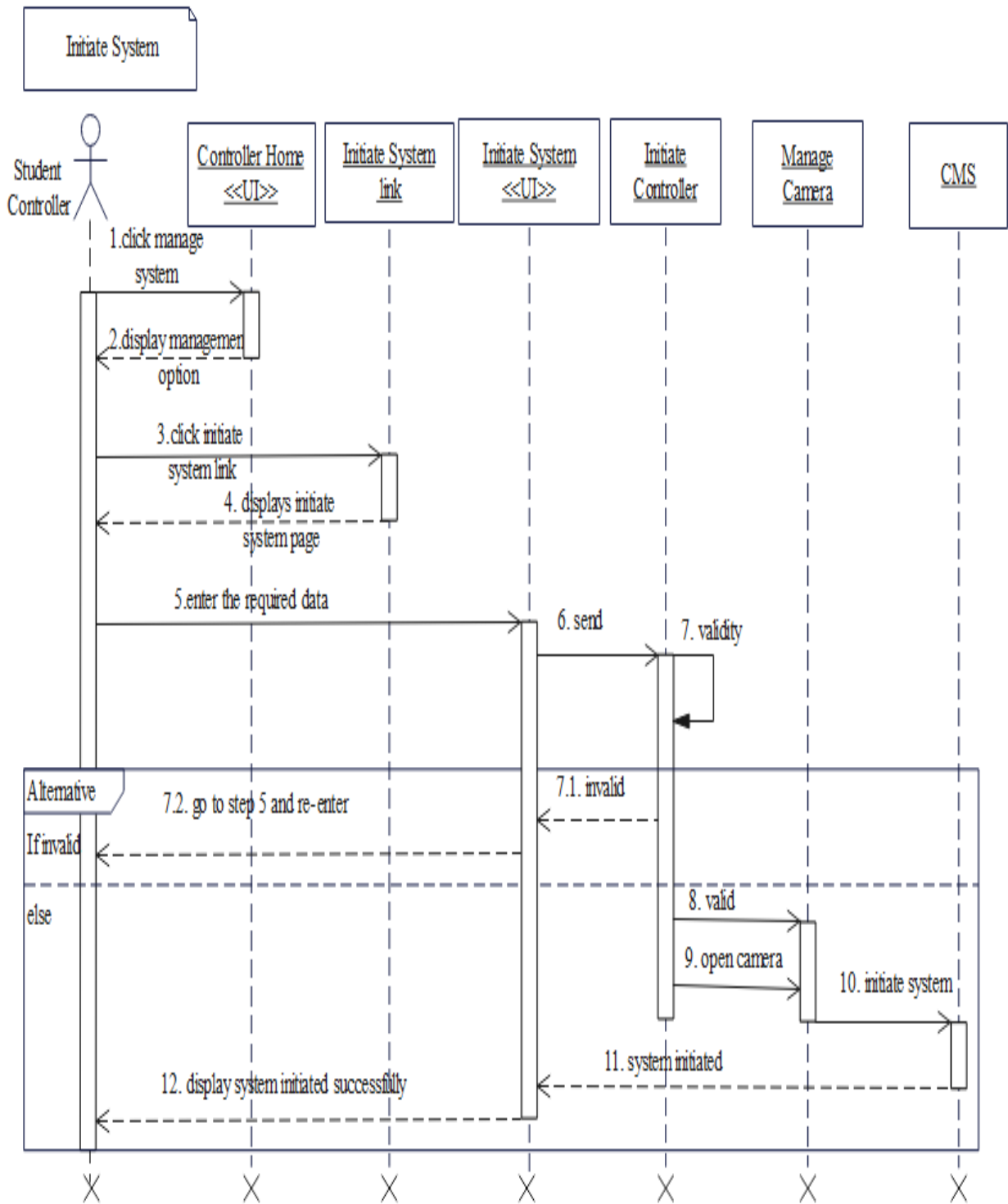


Figure 4. 7 Sequence diagram for Initiate System

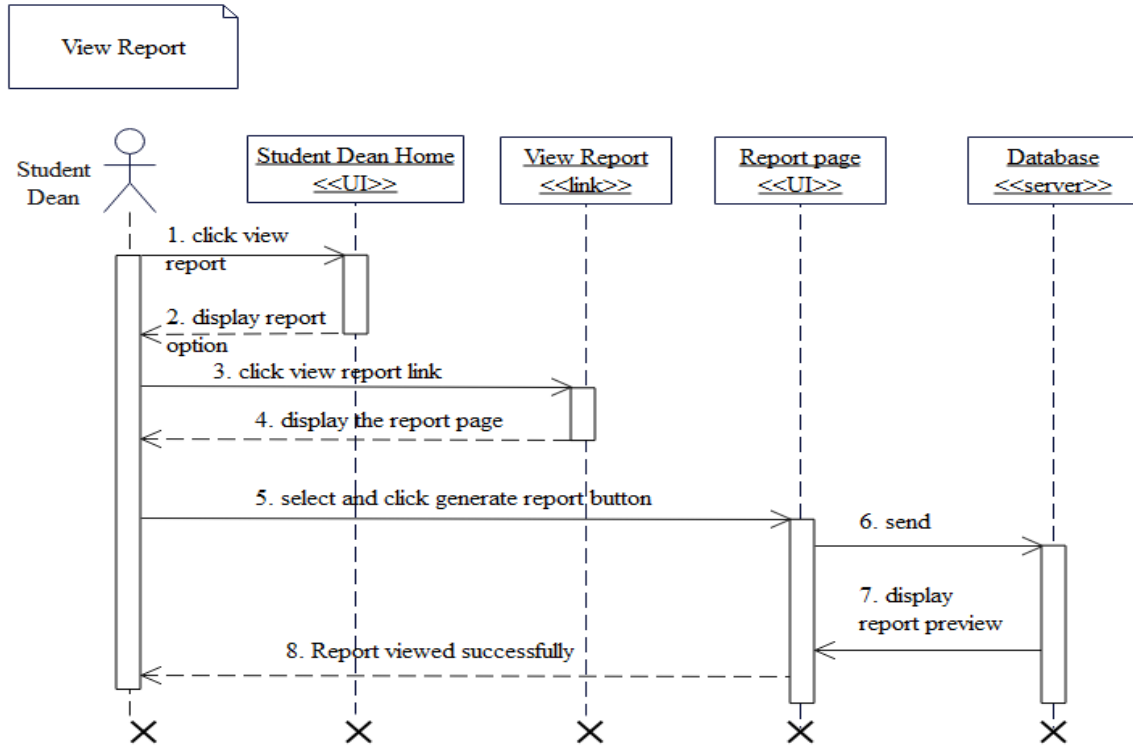


Figure 4. 8 Sequence diagram for View Report

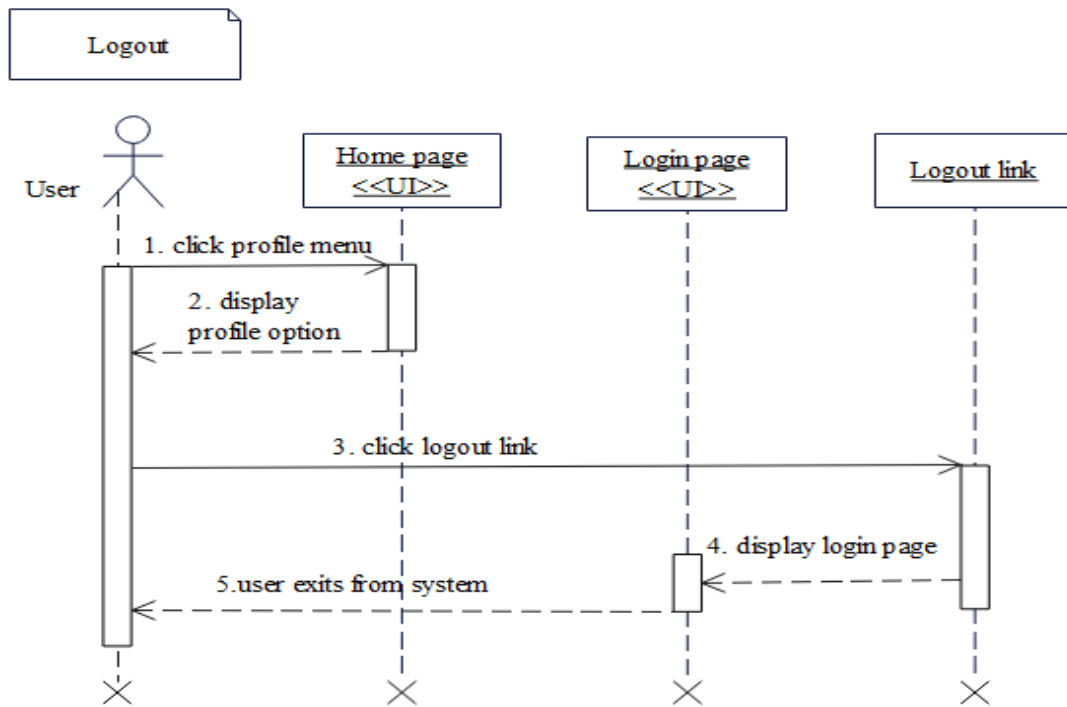


Figure 4. 9 Sequence diagram for Logout

4.3.2. Activity Diagram

An activity diagram is a flow chart to represent the flow from one activity to another activity.

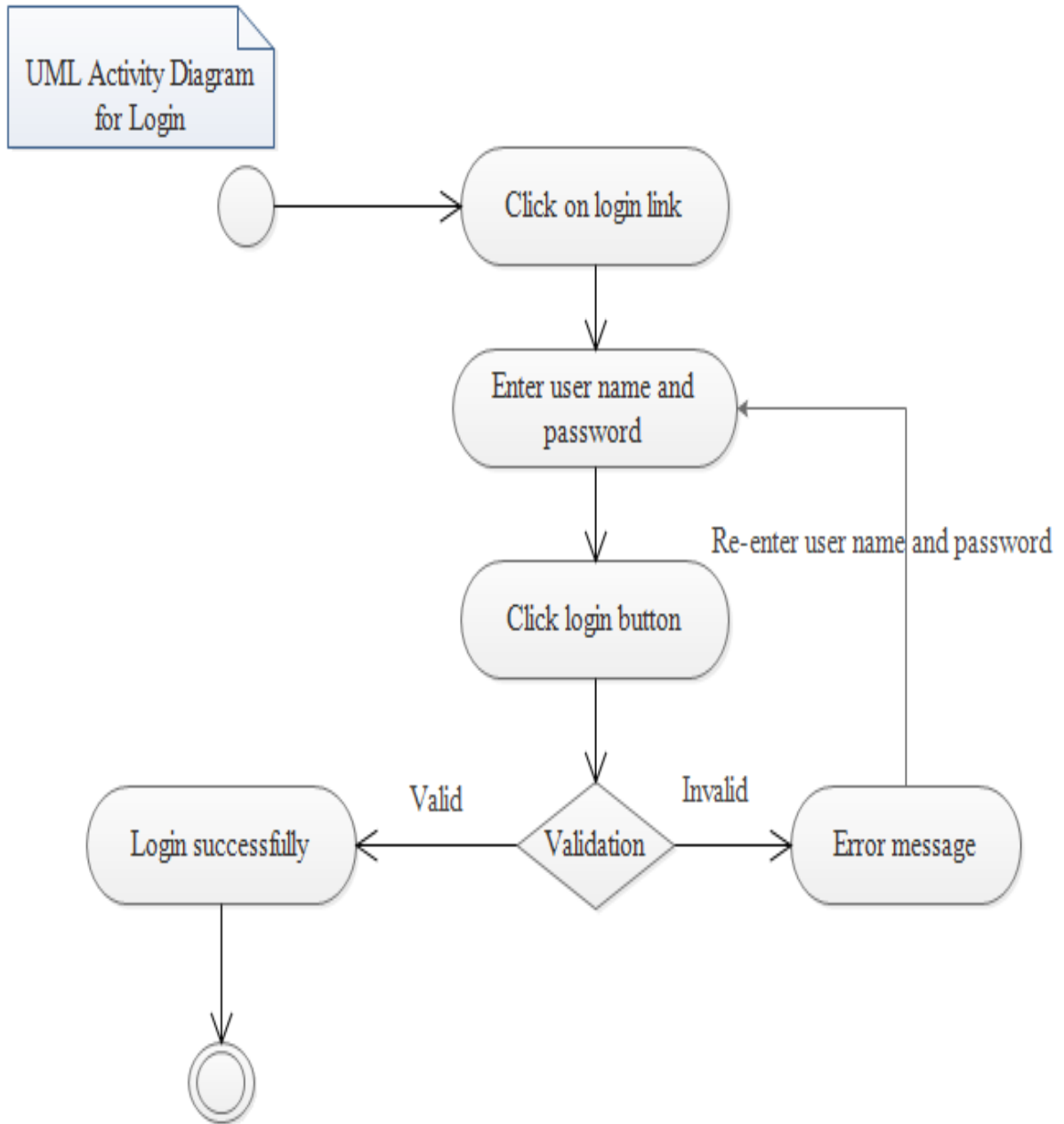


Figure 4. 10 Activity diagram for Login

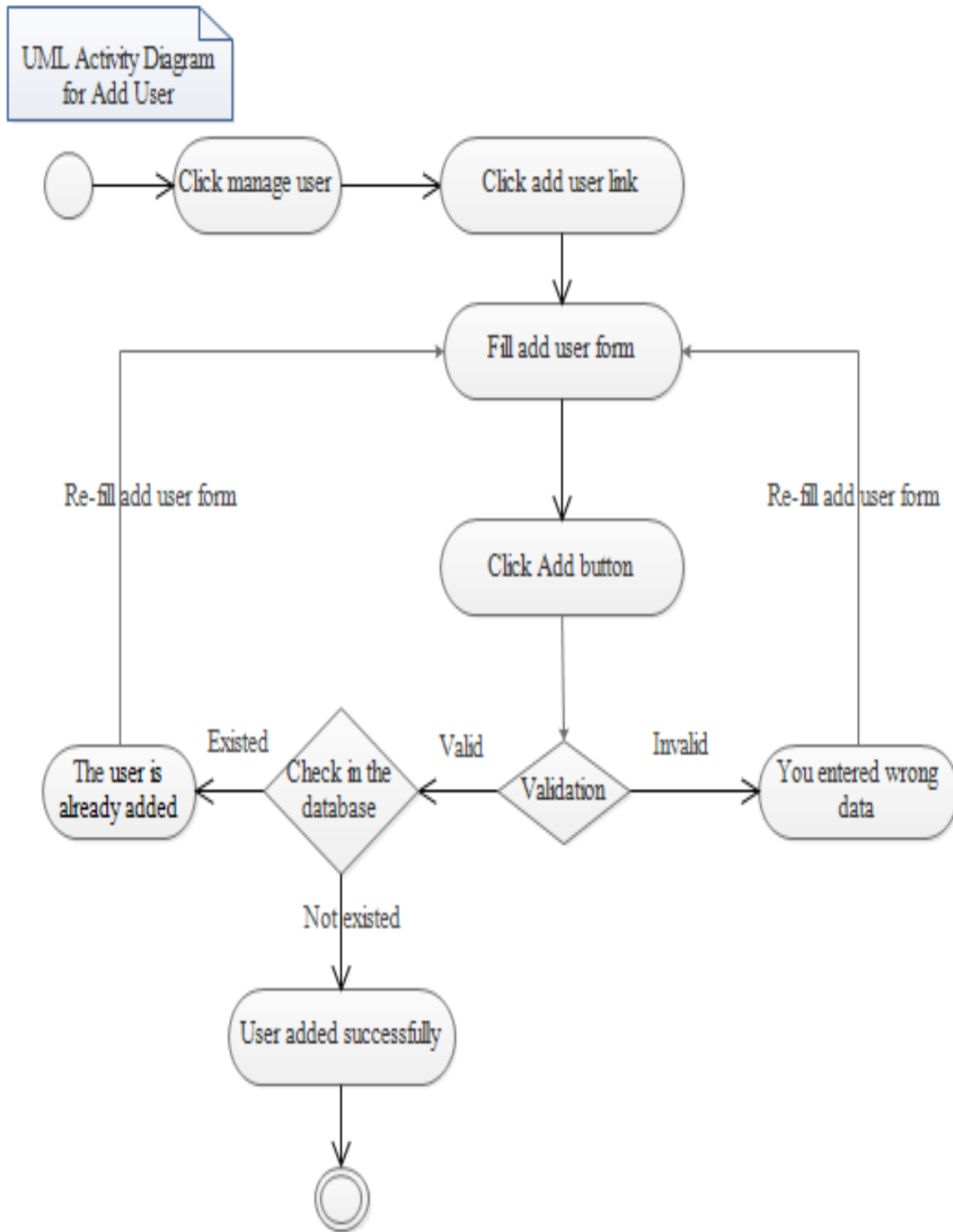


Figure 4. 11 Activity diagram for Add User

UML Activity Diagram for Student Registration

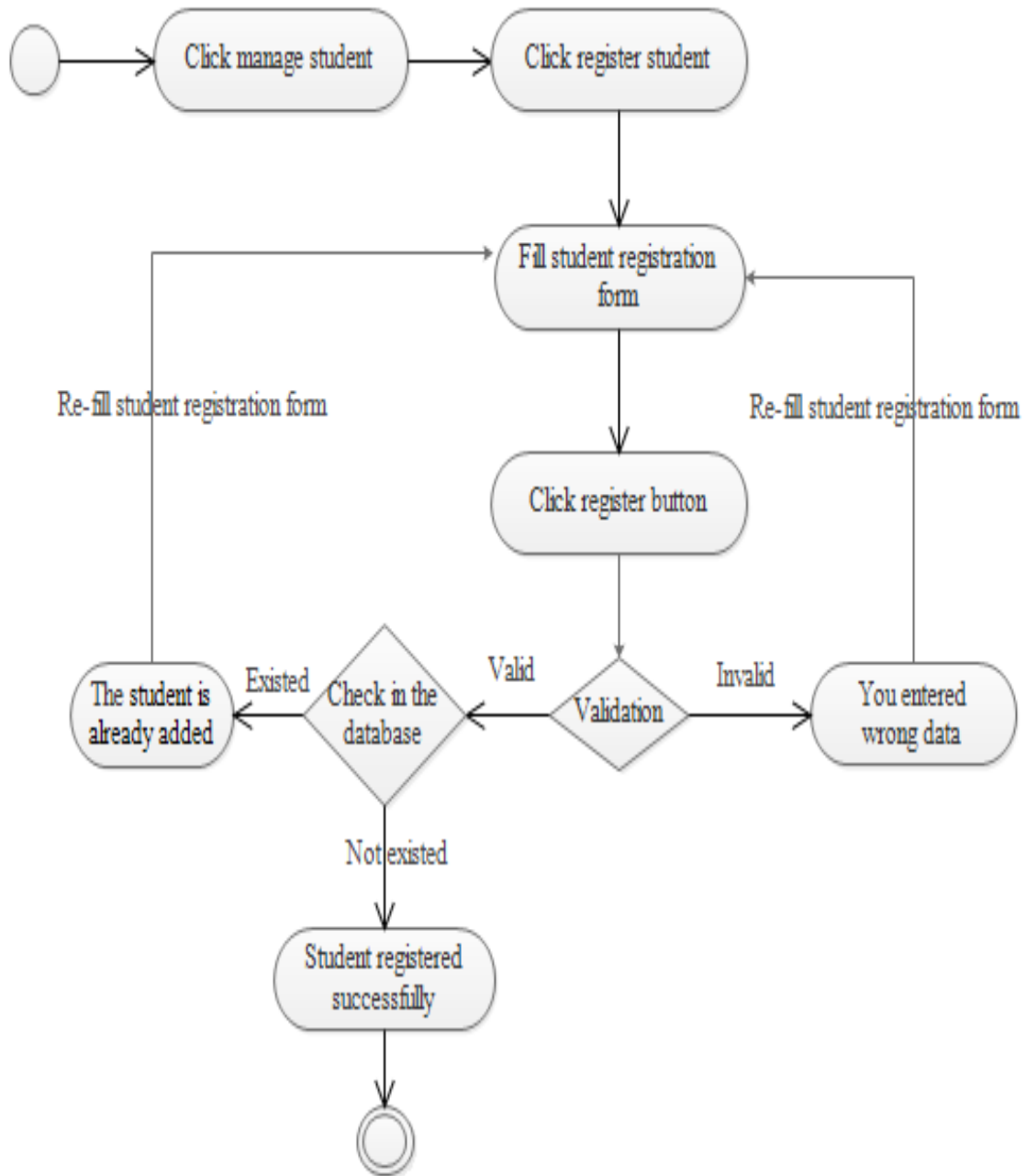


Figure 4. 12 Activity diagram for Register Student

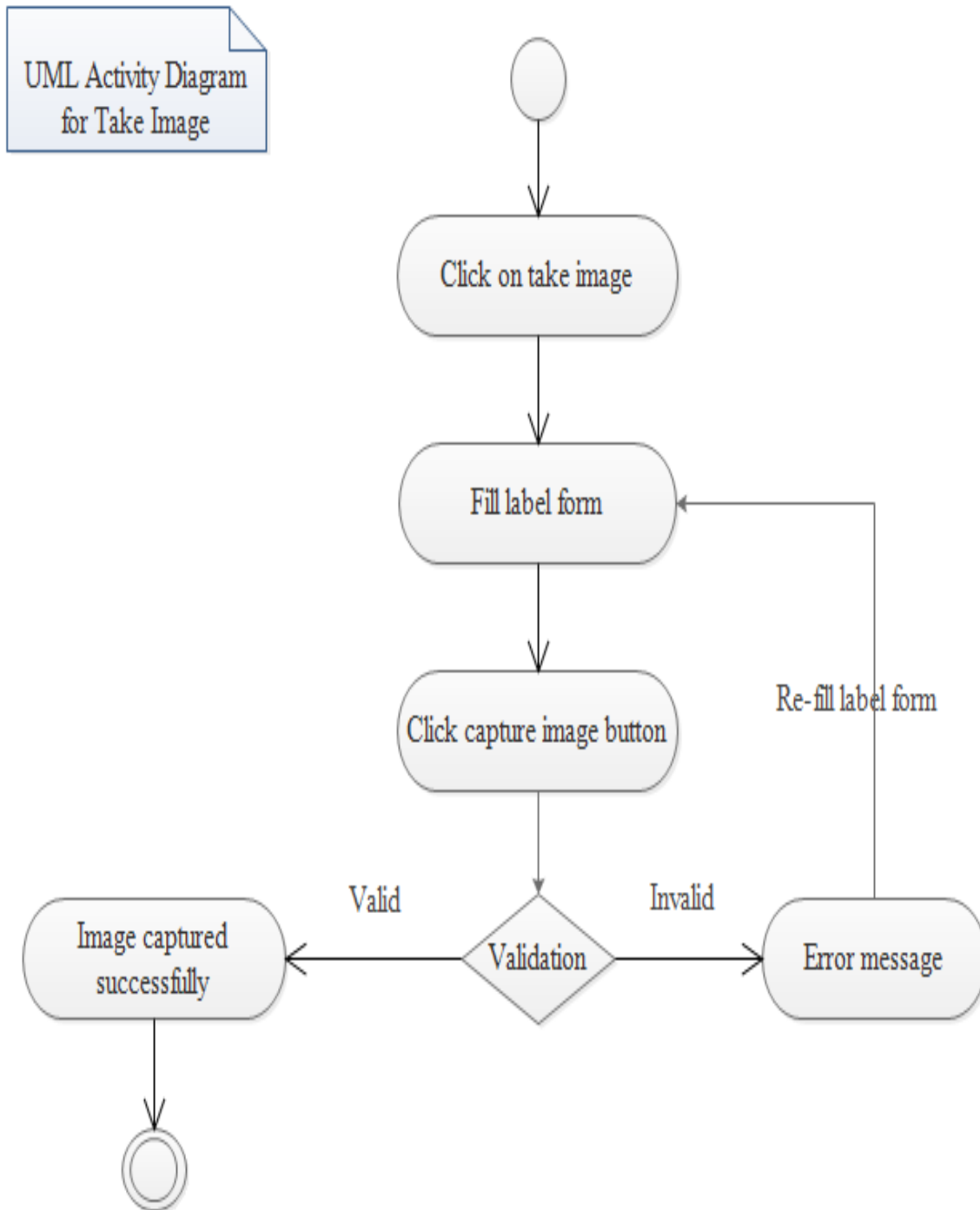


Figure 4. 13 Activity diagram for Taking Student's Image

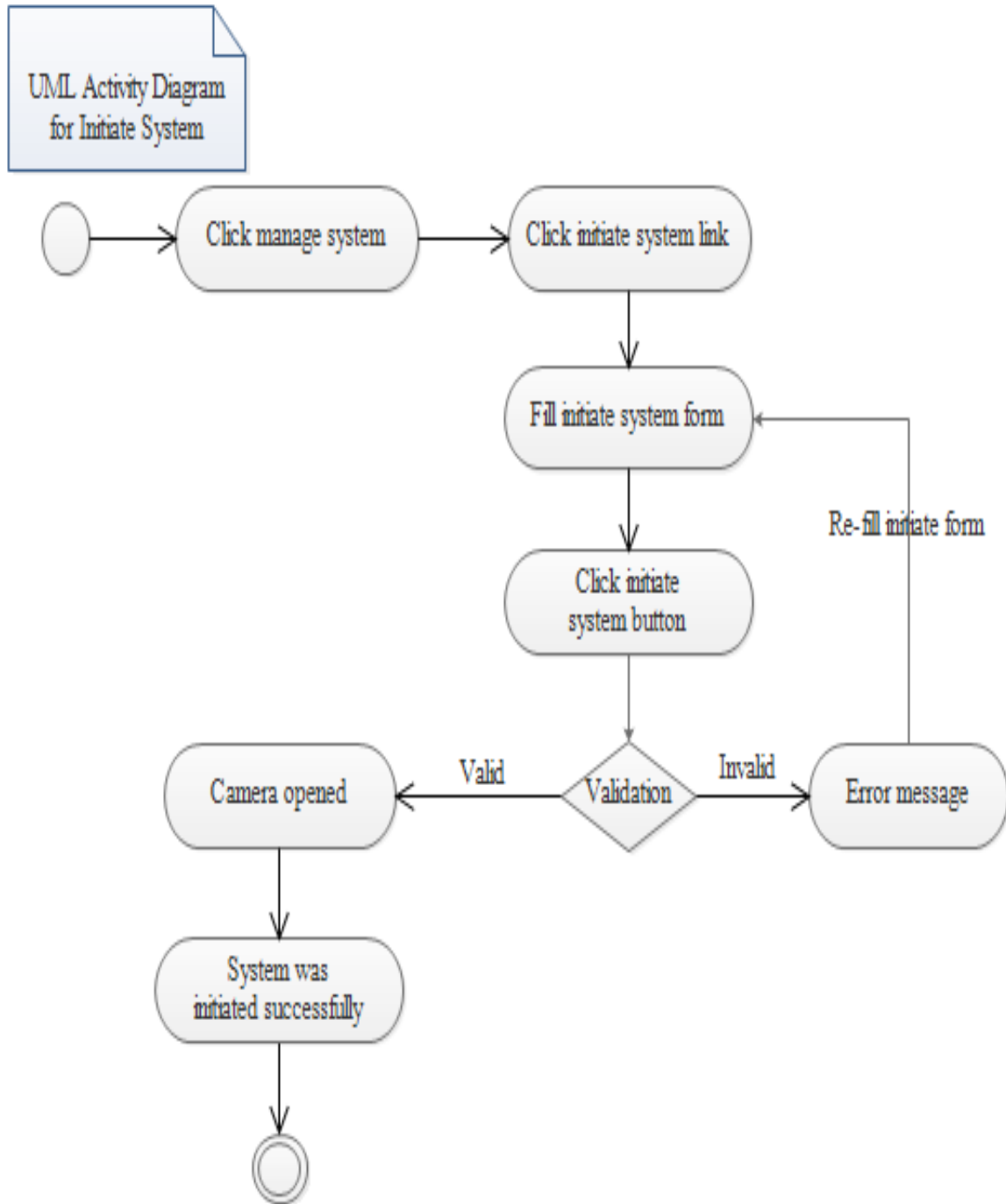


Figure 4. 14 Activity diagram for Initiate System

UML Activity Diagram
for Face Recognition

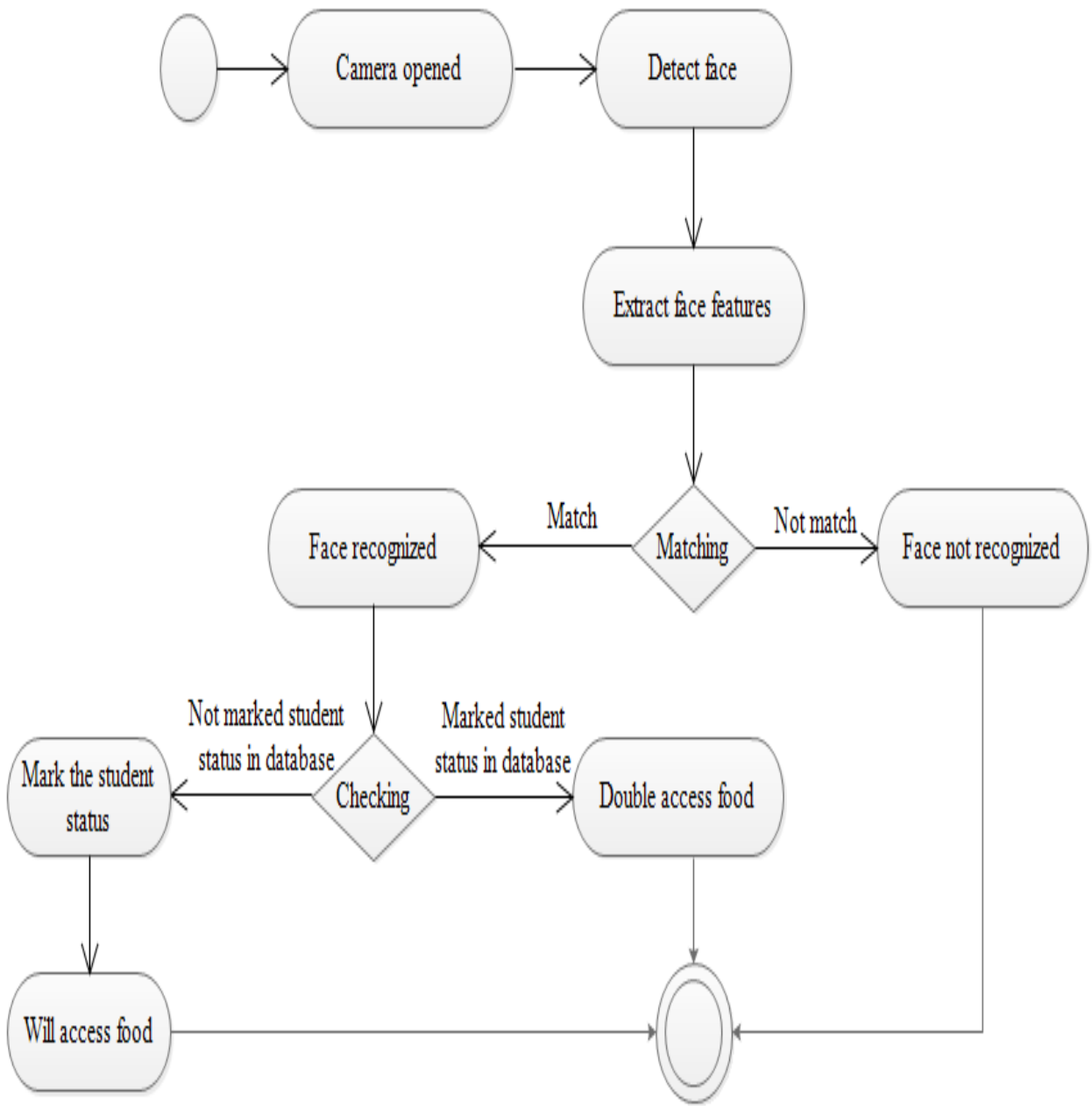


Figure 4. 15 Activity diagram for Face Recognition

4.3.3. State Chart Diagram

A state chart diagram is a UML diagram used to model the dynamic nature of a system. It is simply a presentation of a state machine that shows the flow of control from state to state [1].

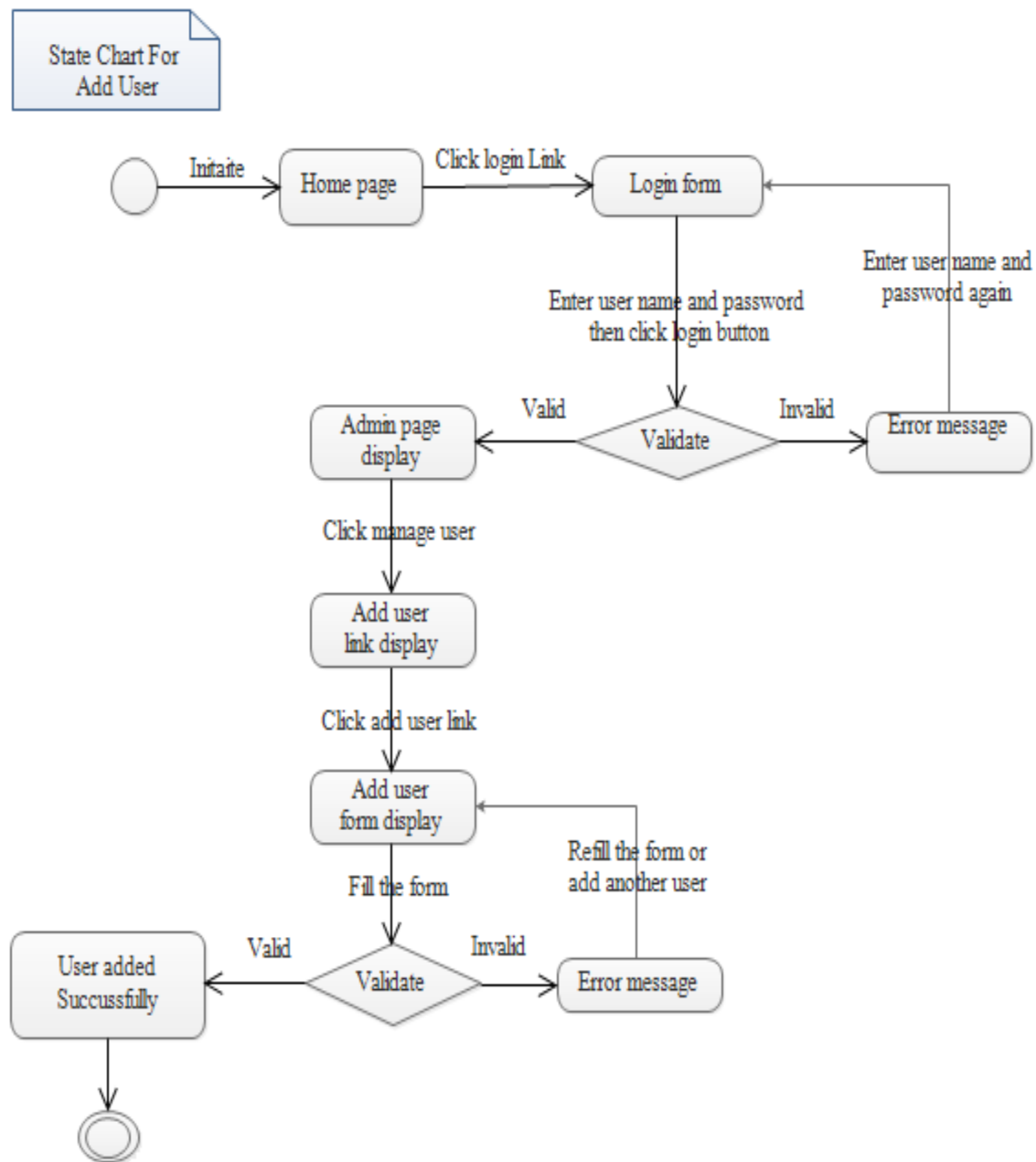


Figure 4. 16 State chart diagram for Add User

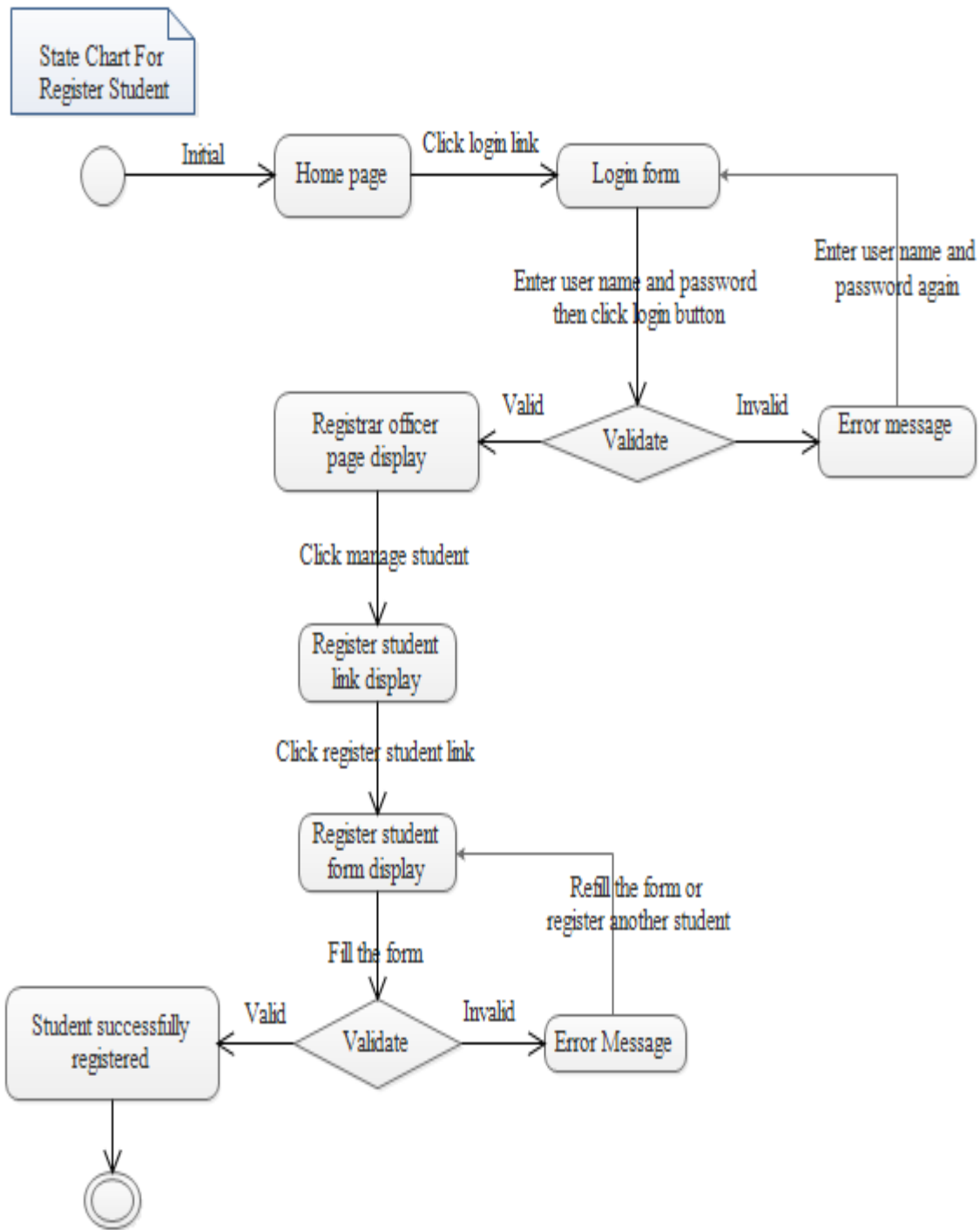


Figure 4. 17 State chart diagram for Register Student

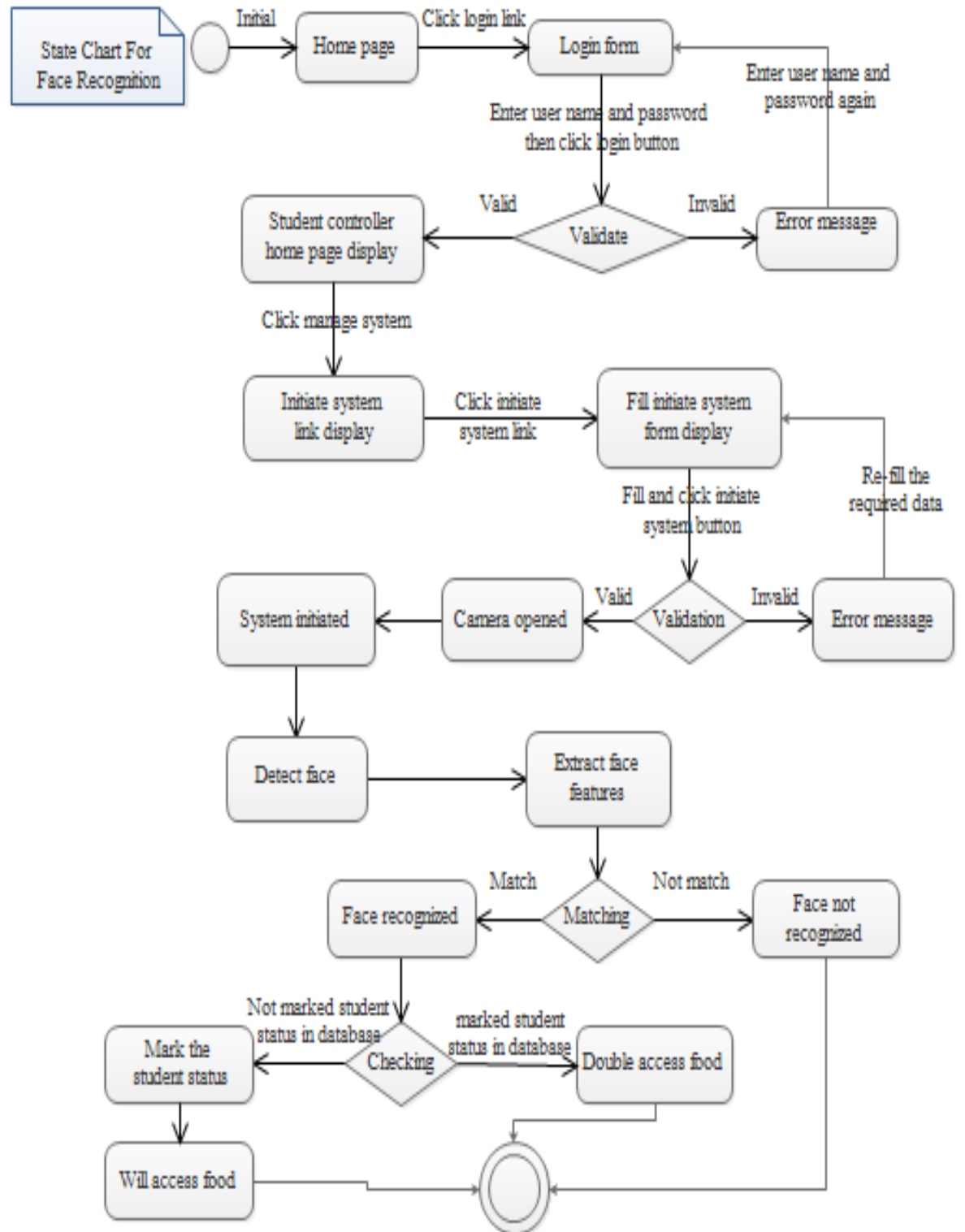


Figure 4. 18 State chart diagram for Face Recognition

CHAPTER FIVE

5. SYSTEM DESIGN

The Systems design of this project defines the elements of the cafeteria management system using face recognition like modules, architecture, components, deployment, persistent, access control, and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing, and designing systems that satisfy the specific needs and requirements of the stakeholder and also defines the coherent and well running system.

5.1. Design Goals

The goal of system design is to manage complexity by dividing the system into smaller and manageable pieces [1].

5.1.1. User Interface and Human Factors

Our system would have an easy and clear user interface. Since the user interface is simple, clear, and has clear user guide links and buttons, users who have little computer basic skills or with little training can use the system. In addition to simple and clear usage, the system user interface is more attractive for users and interactive to use the system. Our system has a simple and clear guide for users on where to click or what to select to perform the desired action.

5.1.2. Hardware Consideration

The hardware requirements of our system are considerably affordable and accessible. The system would use computer devices, cameras, and deployment servers. These devices are already available. No need for a supercomputer or complex electronics devices.

5.1.3. Security Issues

The user of our system would be only allowed, users. Users must be registered into the system unless and otherwise, the system would not allow users for security purposes. Data editing or removal process is also performed by only users who have administrative roles. Username and password validation rules are also part of the security policies of the system. Our system would strongly withstand malicious attacks.

5.1.4. Performance Consideration

The performance of our system would be efficient. The response time and memory requirements are less to perform its tasks. The system would handle as many as possible users at a time. It required considerable response time. It can provide the desired service within a short period and also, the Django framework is fully loaded with extras and scalable, the proposed system has a better response time and it can support many users concurrently.

5.1.5. Error Handling and Validation

The systems can ably handle exceptions that may happen while the user uses the system. When a user interacts with the system errors may occur. To control this kind of inaccuracy the system would generate different user-friendly messages. It handles exceptions of data duplication to save memory space errors related to the cafeteria management system such as one student cannot register twice and cannot take the image of the student twice.

5.1.6. Quality Issues

Our system would be reliable by handling exceptions when the user uses the system as mentioned before and by performing its function consistently. In the quality assessment, the users would be involved by a feedback mechanism in which they can give comments on the system.

5.1.7. Backup and Recovery

The system would use physical (copy of physical database files that stores database information in another location and forms the foundation of the database recovery mechanism).

5.1.8. Physical Environment

The system is deployed on the university's main server and any authorized user can access the system on the computer by using a web browser.

5.1.9. Resource Issues

The resource issue would not be a problem in the system. Users can use the system by using a computer, camera, and the deployment of the system would be in the university's main server so there would be no resource issue to implement and use the system.

5.2. Current System Process

Current system architecture describes the architecture of the system being replaced and briefly describe the assignment of functionality of the existing system.

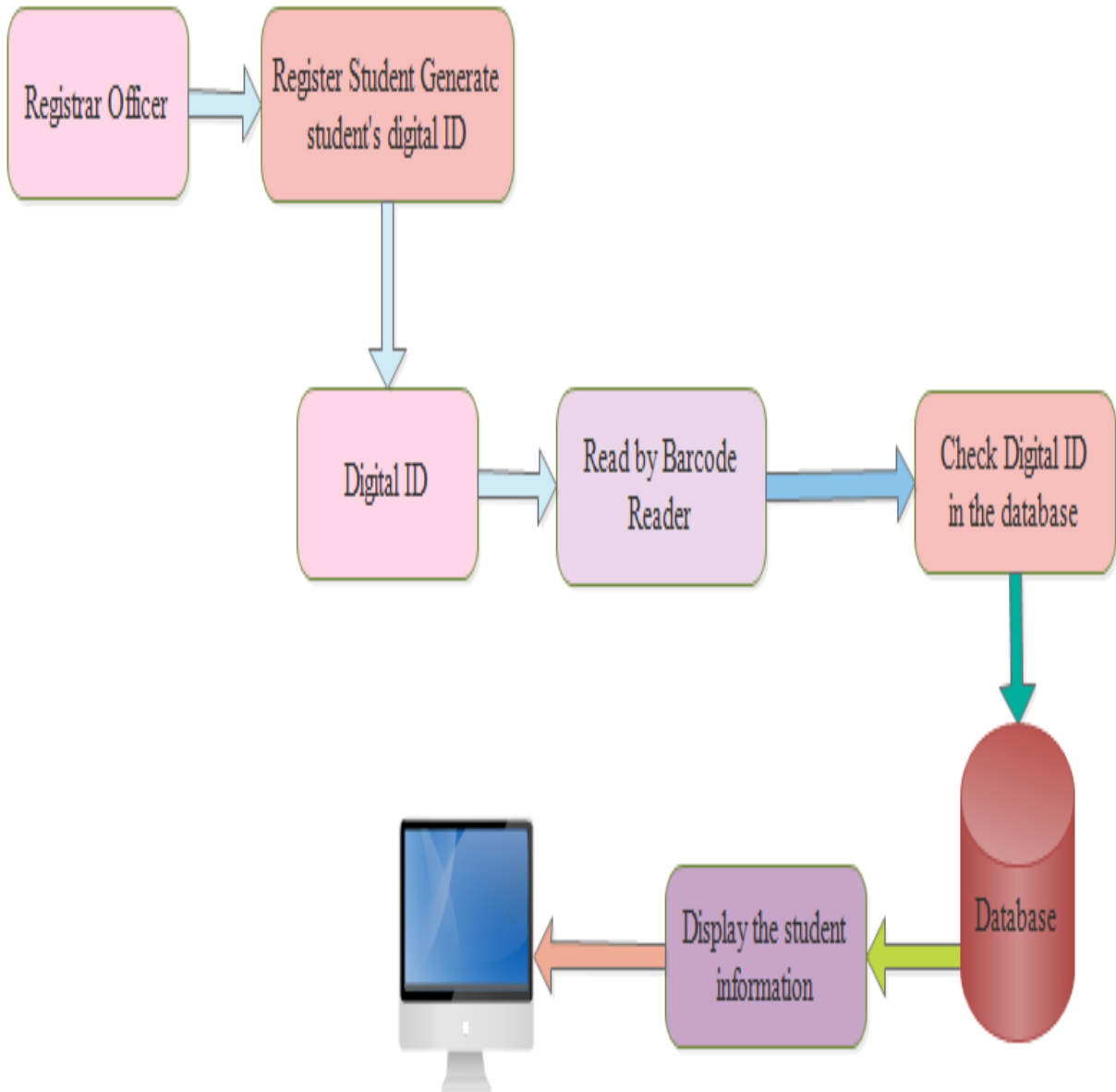


Figure 5. 1 Current system process

5.3. Proposed System Architecture

This section presents a general view of our system architecture and briefly describes the assignment of functionality to each subsystem.

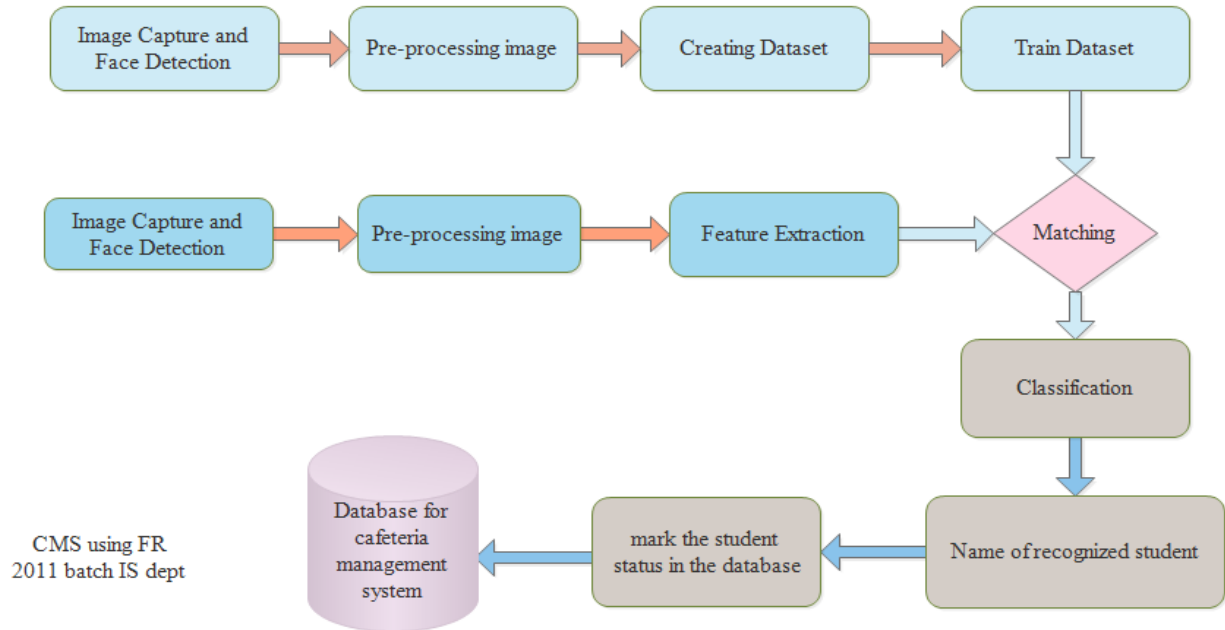


Figure 5. 2 Proposed system Process

We would use multi-tier architecture in our system: presentation tier, application tier, and database tier.

Presentation tier

The presentation tier is the user interface and communication layer of the application, where the end-user interacts with the application. Its main purpose is to display information and collect information from the user. Moreover, the authorized user captures the frames or student's face image using a camera or webcam.

Application tier

The application tier, also known as the logic tier or middle tier, is the heart of the application. This is the tier where processing such as face detection, preprocessing image, face recognition, and marking student status, is performed. The application tier can also add, delete or modify data in the data tier.

Data-tier

The data tier, sometimes called the database tier, data access tier, or back-end is where the information processed by the application is stored and managed. The students' face images are stored in the database after the face detection and preprocessing image. The stored images are used for training the face recognition algorithm. After successful recognition of the face, student status is marked to the database.

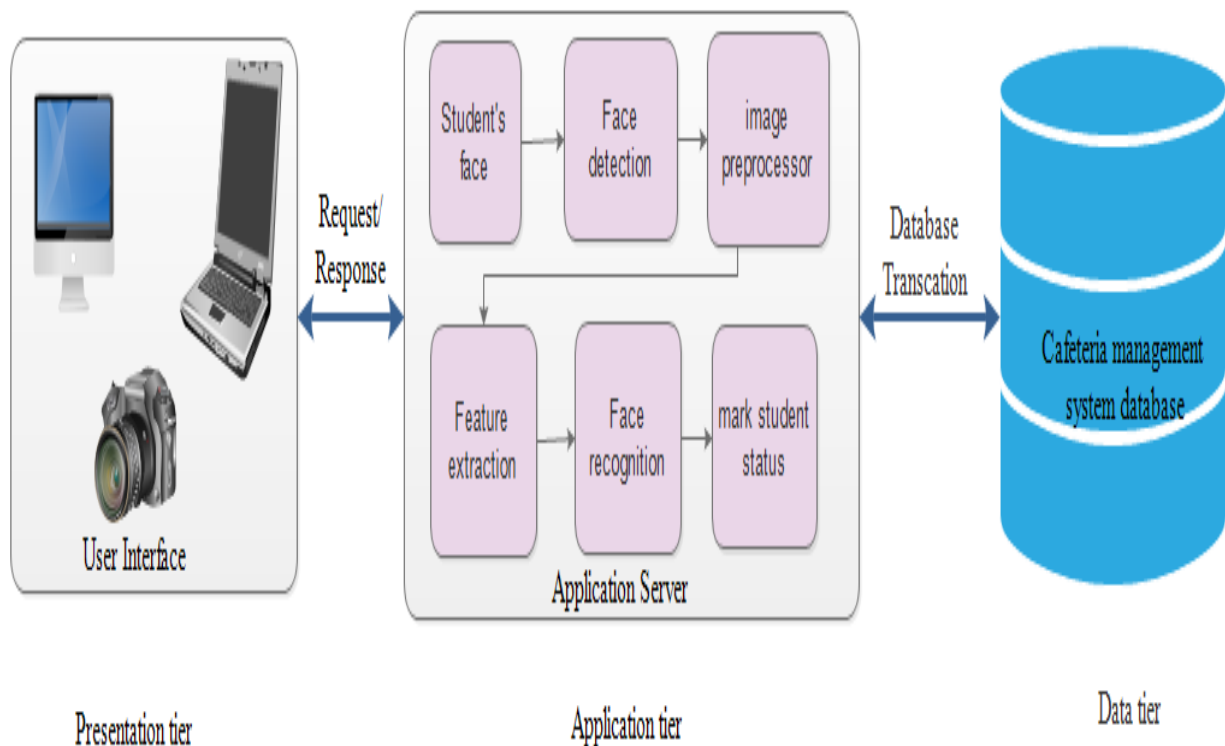


Figure 5. 3 General proposed architecture

5.3.1. Subsystem Decomposition and Description

Decomposing the system enables us to minimize the complexity of the system by decomposing it into subsystems. Subsystems identified from our system are: -

User Subsystem: - allow us to manage the user's information with username and password.

- ✓ Add user
- ✓ Update user
- ✓ Delete user
- ✓ View user

- ✓ Activate user
- ✓ Deactivate user
- ✓ Manage user role

Student Subsystem: - allow us to manage the student's information.

- ✓ Register student
- ✓ Update student
- ✓ Delete student
- ✓ View student
- ✓ Make cafe student
- ✓ Make non-cafe student

Face Recognition process Subsystem: - allow us to recognize a student for identification.

- ✓ Create dataset
- ✓ Train model
- ✓ Face detection and extraction
- ✓ Face recognition
- ✓ Marking student status

Manage System Subsystem: - allow us to initiate, and cease the system for performing the face recognition to identify the students.

- ✓ Initiate system
- ✓ Cease system

Student Status Subsystem: - allow to view the student status taken through face recognition.

- ✓ View student status
- ✓ Mark student status

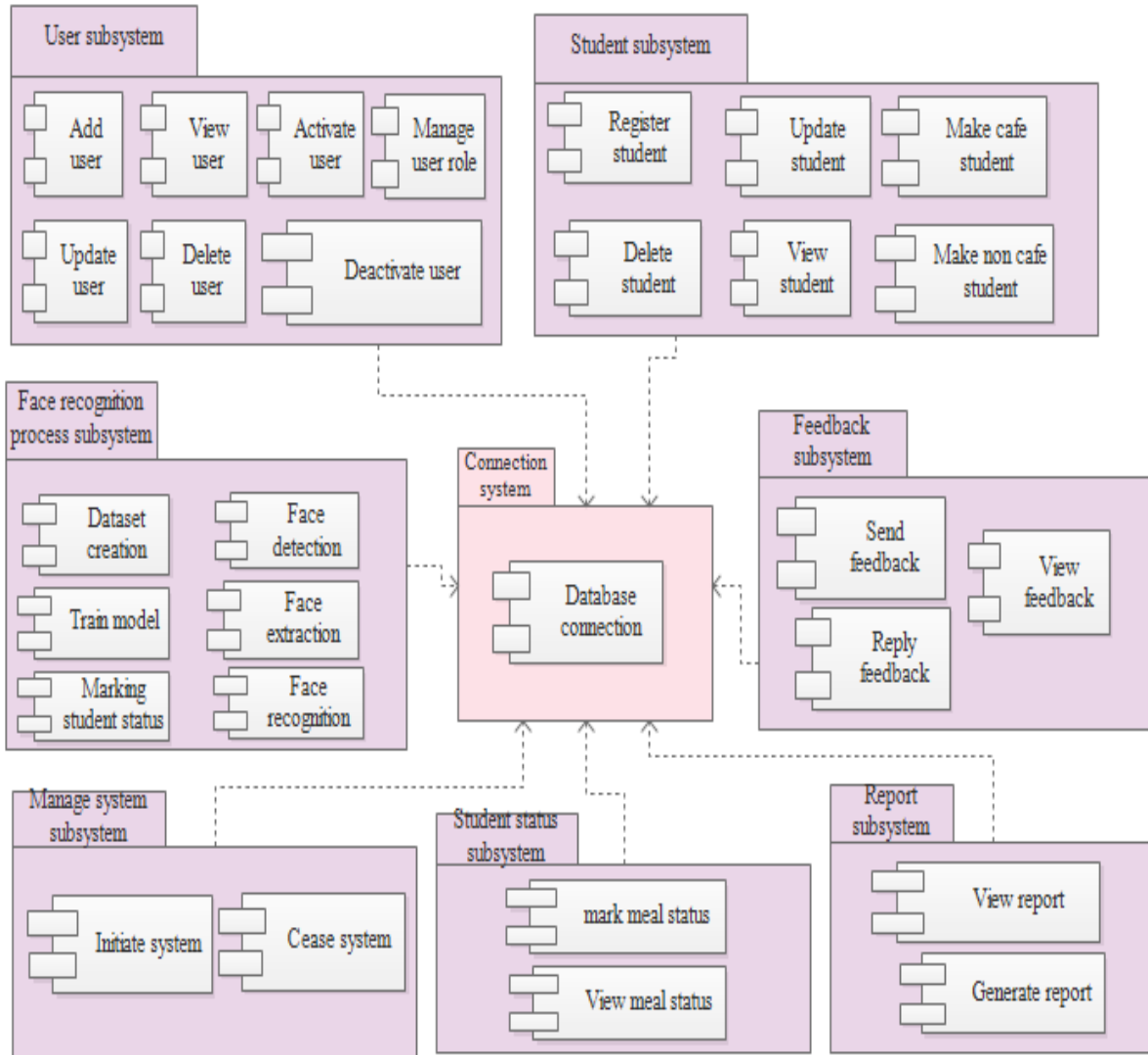
Report Subsystem: - allow the system to generate report and view report from student status.

- ✓ Generate report
- ✓ View report

Feedback Subsystem: - allow us to send, reply, and view feedback.

- ✓ Send feedback
- ✓ Reply feedback
- ✓ View feedback

The following figure describes the above description in the diagram.



Industrial project group-1 2011 batch IS dept

Figure 5. 4 Subsystem decomposing

5.3.2. Hardware/Software Mapping

Hardware or software mapping describes how subsystems are assigned to hardware and customized components. We use a UML deployment diagram to diagrammatically illustrate our proposed system's hardware/software mapping.

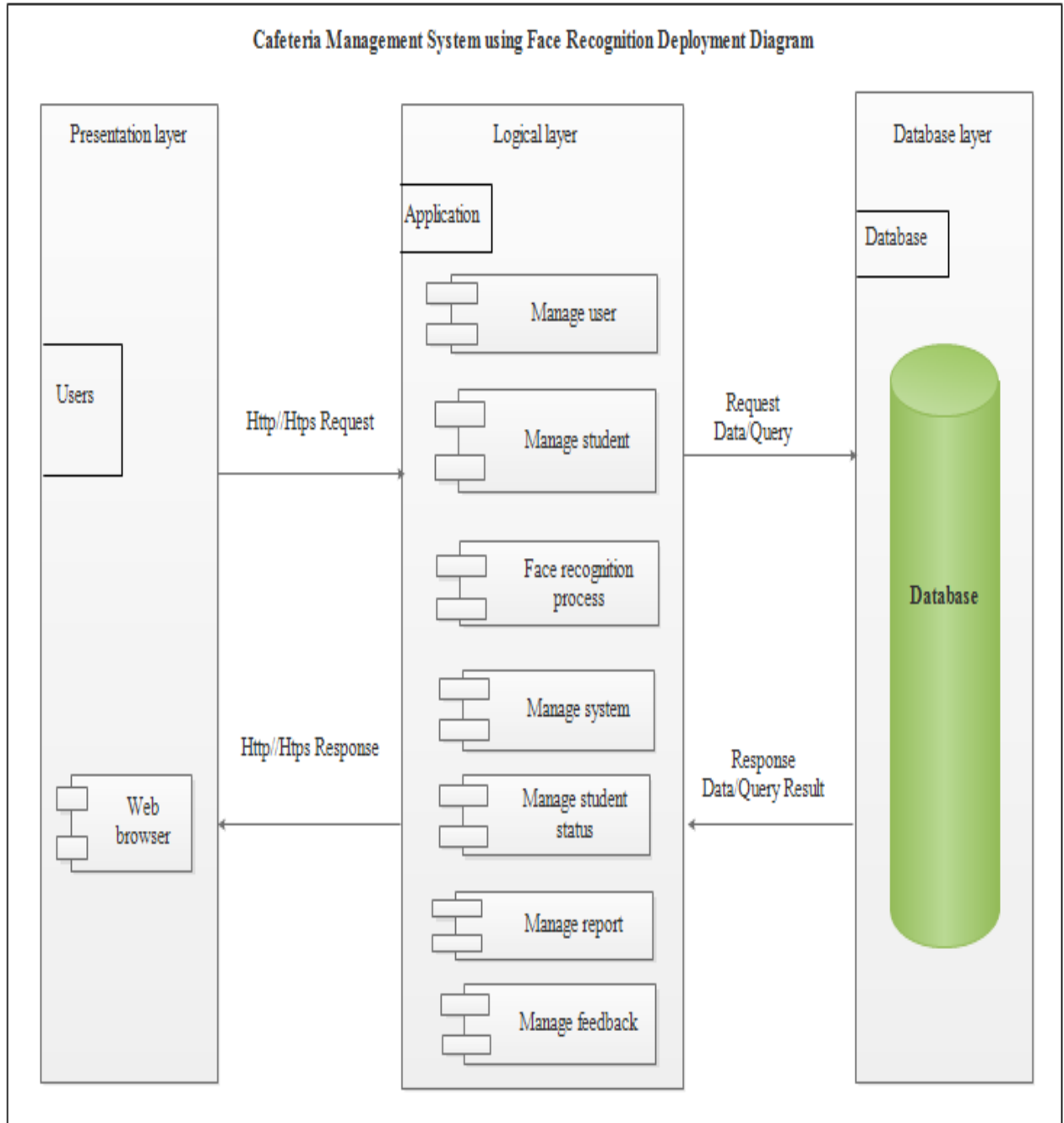


Figure 5. 5 Hardware/Software diagram

5.3.3. Detailed Class Diagram

This detailed class diagram shows classes, attributes, methods (operations), data types, visibility ((public (+), private (-), protected (#)) of the attributes and methods), inheritances, associations, aggregation, composition, dependencies, and multiplicities. Below we use the detailed class UML diagram to show how the aforementioned concepts are organized and designed for a better understanding of the system's detailed class diagram.

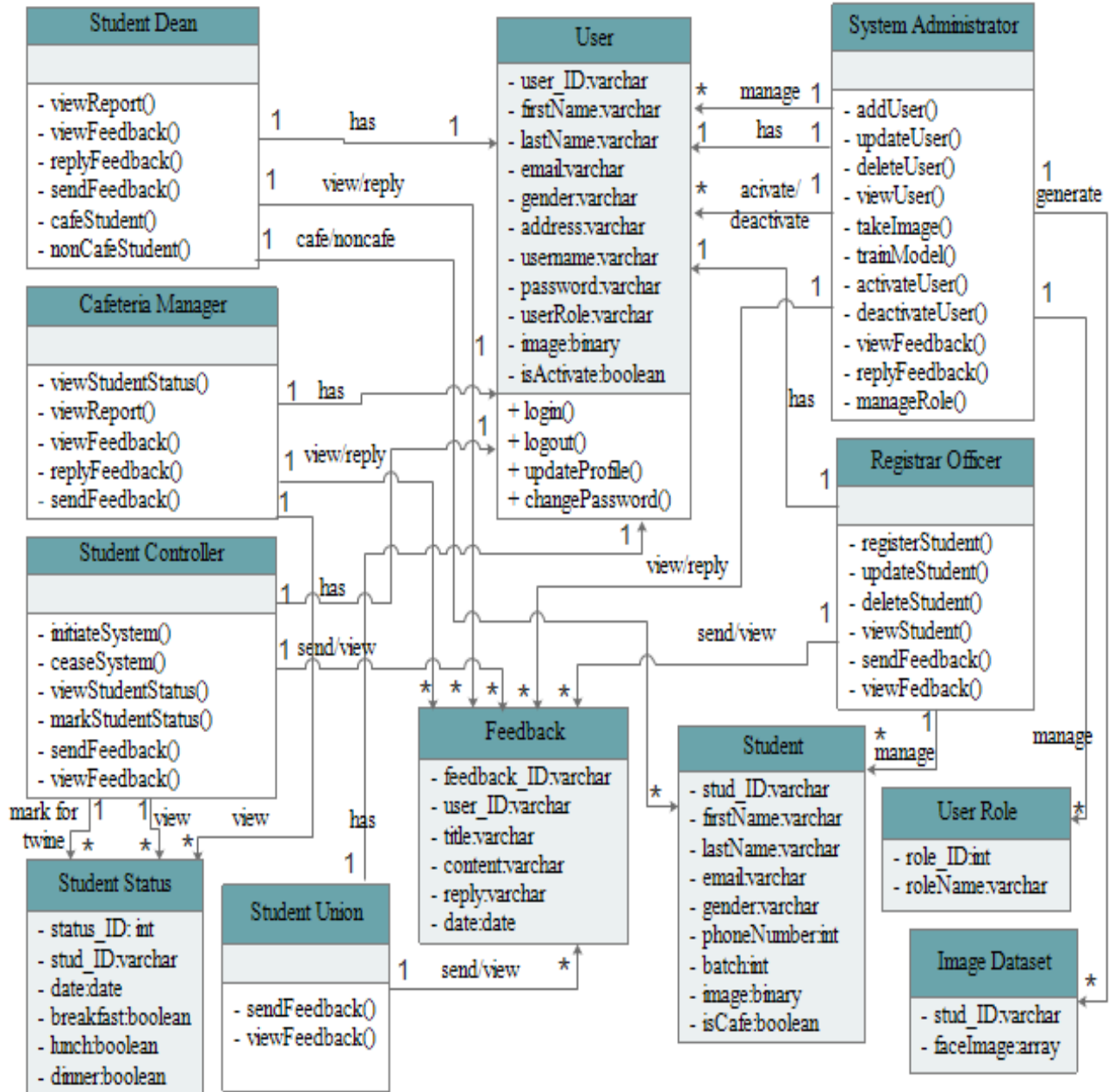
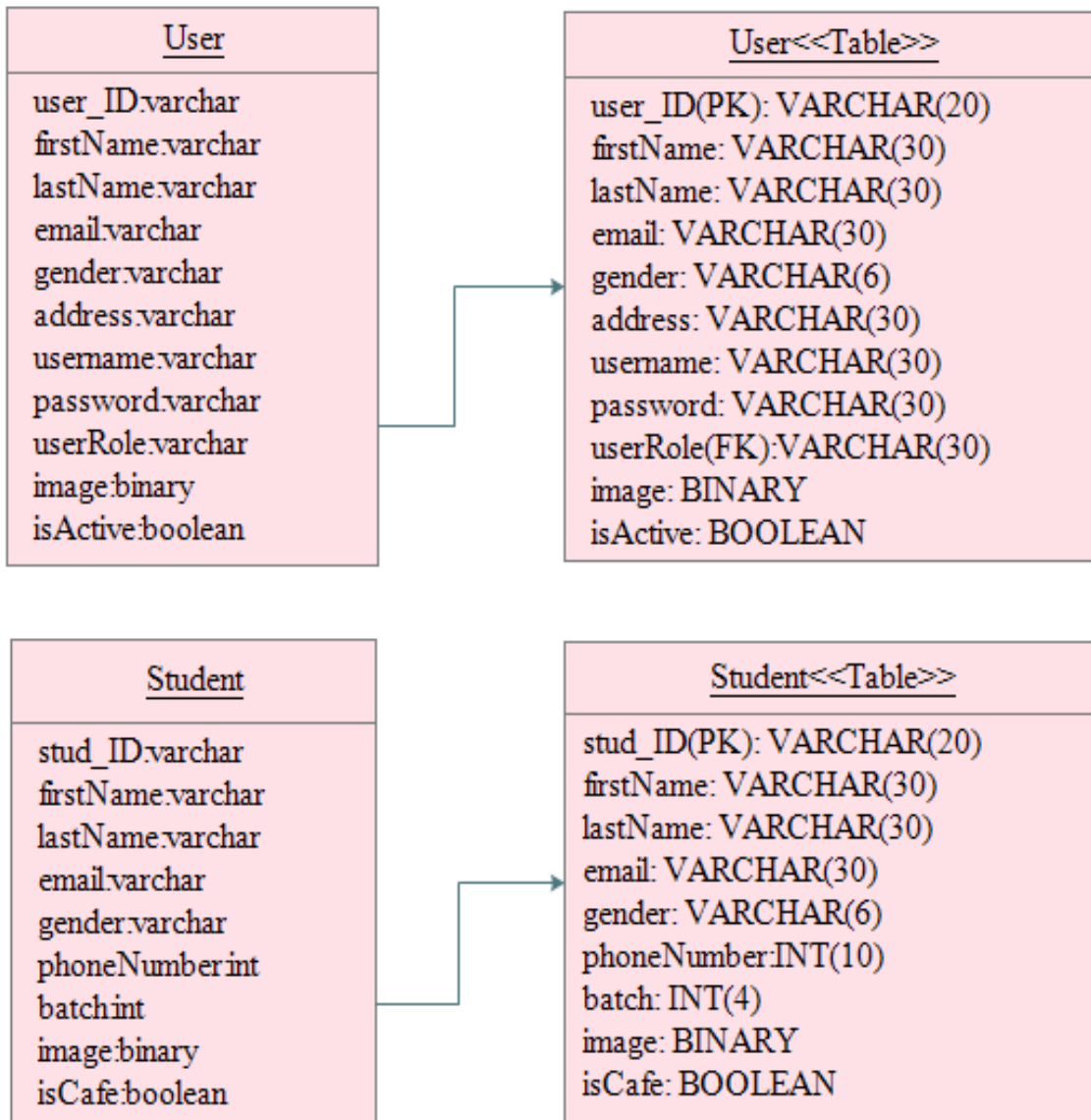


Figure 5. 6 Detail class diagram

5.3.4. Persistent Data Management

Persistence data encapsulate the capability to store, retrieve, and delete objects/data permanently. In the current database system, we have used different tables as objects and each object is related to each other. This schema enables data manipulation activity such as select, search, delete, and update on the database.

The following figure indicates the persistent data management of the system.



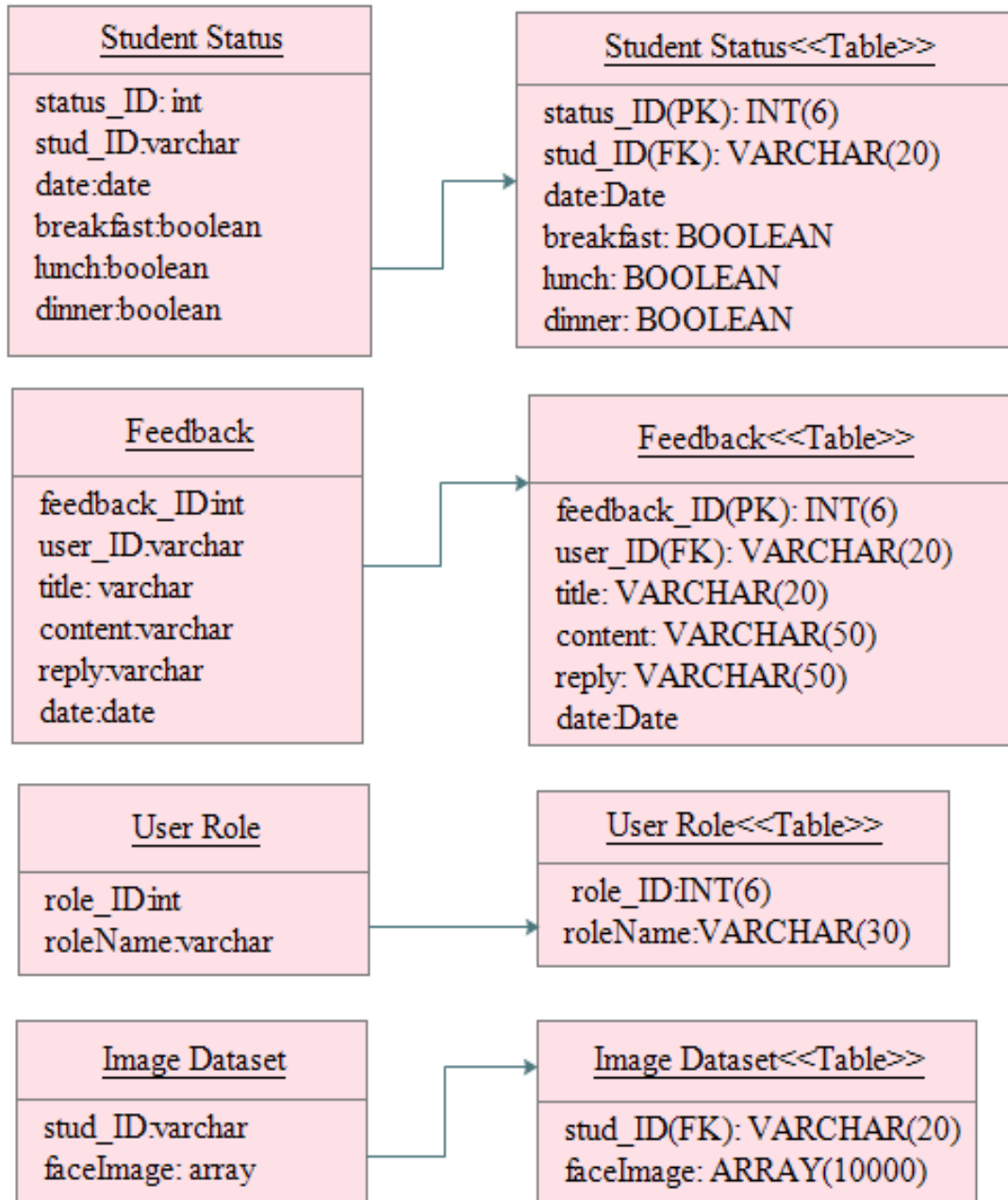


Figure 5. 7 Persistent data management

5.3.5. Access Control and Security

In our system, different actors have access to different information and data. Access control and security specify what the user can access or what cannot perform by some users. This access control is verified by a username and password. The system admin represents an authenticated user. The system follows a multi-user system. In a multi-user system, different actors have access to different functions and data.

Then it must be having: -

Table 5. 1 Access control and security

Functionality	System Administrator	Student Dean	Registrar Officer	Cafeteria Manager	Student Controller	Student Union
Login	✓	✓	✓	✓	✓	✓
Add User	✓					
Update User	✓					
View User	✓					
Delete User	✓					
Activate User	✓					
Deactivate User	✓					
Manage user role	✓					
Take Student's Image	✓					
Train Model	✓					
Register Student			✓			
Update Student			✓			
View Student			✓			
Delete Student			✓			
Make Café Student		✓				
Make Non-Café Student		✓				
Initiate System					✓	
Cease System					✓	
View Student Status				✓	✓	

Mark Student Staus					✓	
View Report		✓		✓		
Send Feedback		✓	✓	✓	✓	✓
Reply Feedback	✓	✓		✓		
View Feedback	✓	✓	✓	✓	✓	✓
Update Profile	✓	✓	✓	✓	✓	✓
Change Password	✓	✓	✓	✓	✓	✓
Logout	✓	✓	✓	✓	✓	✓

5.4. Packages

A package diagram is a kind of structural diagram that shows the arrangement and organization of model elements in the system and shows the dependencies between sub-systems.

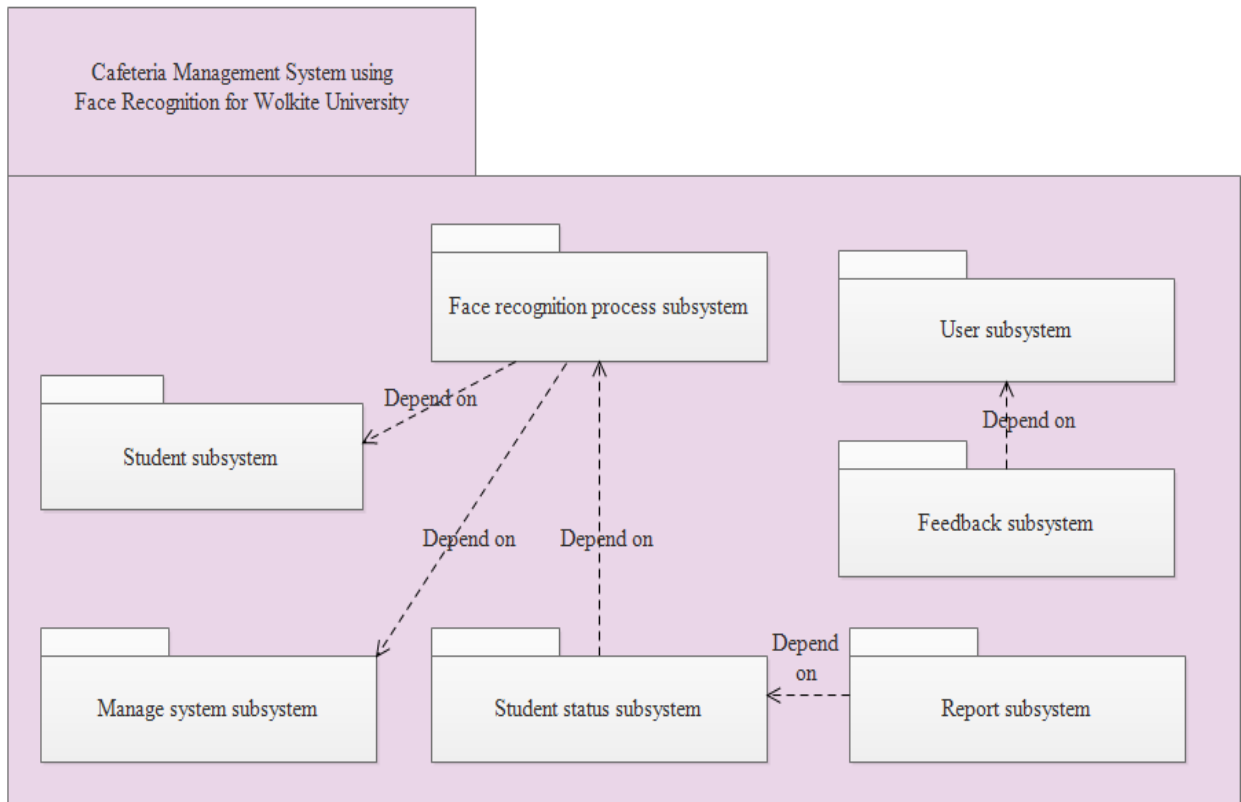


Figure 5. 8 Packages diagram

5.5. Algorithm Design

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.

Algorithm1: Login ()

BEGIN

 Login (username, password)

 INPUT: Username and Password

 IF (User exist)

 READ Password FROM database

 IF (Password == Entered Password)

 Login successful

 Redirect to an authenticated page

 ELSE

 PRINT “incorrect password “

 END IF

 ELSE

 PRINT “incorrect Username or password “

 END IF

END

Algorithm2: addUser ()

BEGIN

addUser (User_ID, First Name, Last Name, Email, Gender, Address, Username, Password, Image)

 INPUT: User_ID, First Name, Last Name, Email, Gender, Address, Username, Password, Image

 IF (User exist)

 PRINT “The user is existing in the database”

 ELSE

```

        PRINT "User added successfully!"
    END IF
END
Algorithm3: registerStudent ()
BEGIN
    registerStudent (Stud_ID, First Name, Last Name, Email, Gender, Phone_no, Batch,
Image)
    INPUT: Stud_ID, First Name, Last Name, Email, Gender, Phone_no, Batch, Image
    IF (Student exist)
        PRINT "The student is existing in the database"
    ELSE
        PRINT "Student registered successfully!"
    END IF
END

```

Algorithm4: takeImage ()

This method enables the user to log in to the system:

```

BEGIN
    takeImage (Label)
    INPUT: Label
    IF (Label exist)
        PRINT "The image of this label was exist in a dataset"
    ELSE
        Camera open
        Detect face image
        Perform preprocessing image
        Create a dataset
        PRINT "Image captured successfully!"
    END IF
END

```

5.6. User Interface Design

The proposed system has a graphic user interface to interact with the user. Below different user interfaces that are visible to all users of the system are described. In addition, can get information posted and different links to other pages.

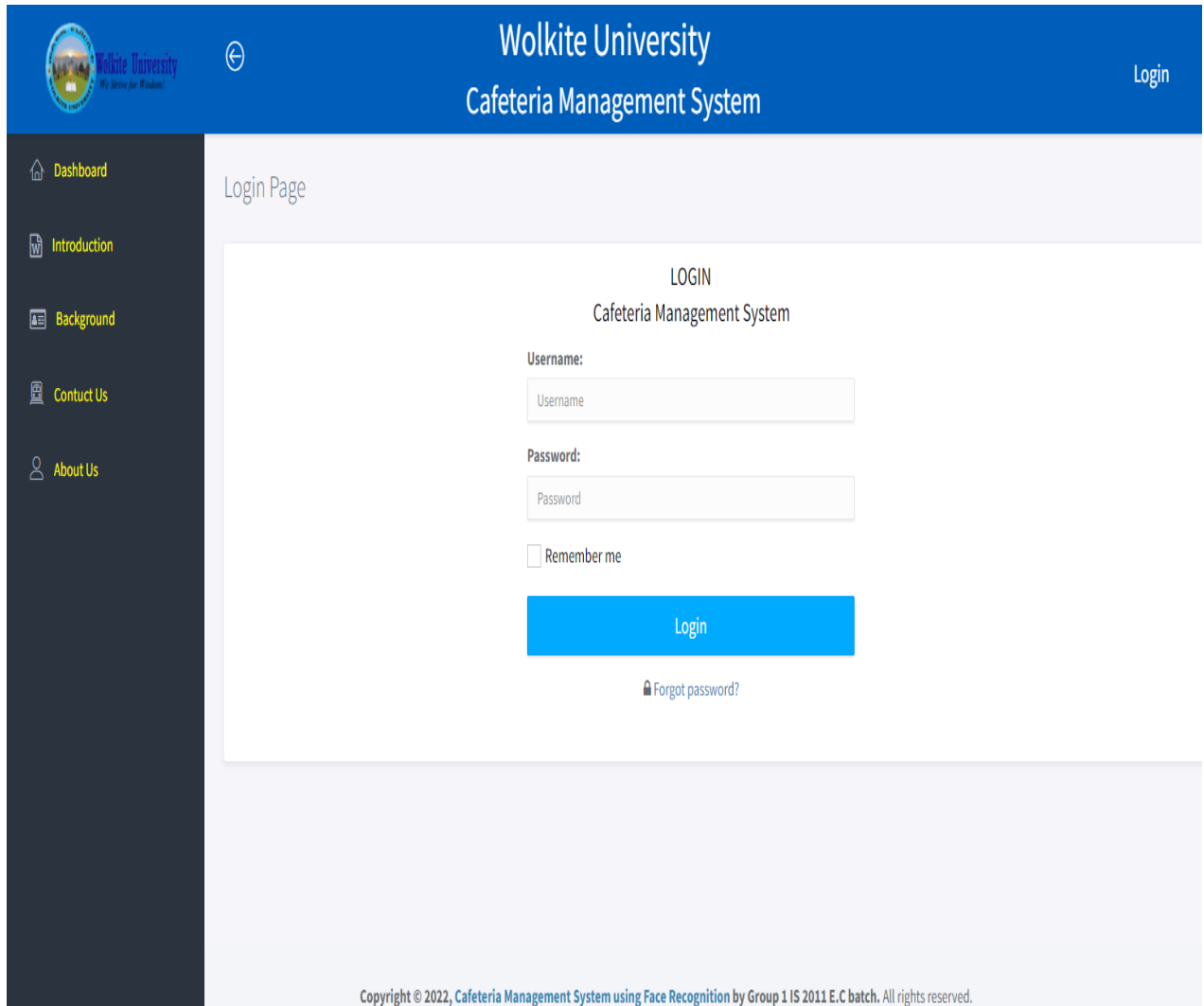


Figure 5. 9 Login page

Wolkite University
Cafeteria Management System

Dejene Tolera

System Admin Dashboard

Role	Count
Student Dean	3
Cafeteria Manager	5
Registrar Director	4
Student Controller	6
Student Union	10
Total User	28

Copyright © 2022, Cafeteria Management System using Face Recognition by Group 1 IS 2011 E.C batch. All rights reserved.

Figure 5. 10 Admin home page

Wolkite University
Cafeteria Management System

Dejene Tolera

Dashboard
Manage User
Add User
List of User
Take Student's Image
Train Model
Manage Feedback

Add User

[Back to List](#)

User ID:

First Name: **Last Name:**

Email: **Gender:**

Address: **Username:**

Password: **Confirm Password:**

User Role: **Profile Picture:** No file chosen

Copyright © 2022, Cafeteria Management System using Face Recognition by Group 1 IS 2011 E.C batch. All rights reserved.

Figure 5. 11 Add user page

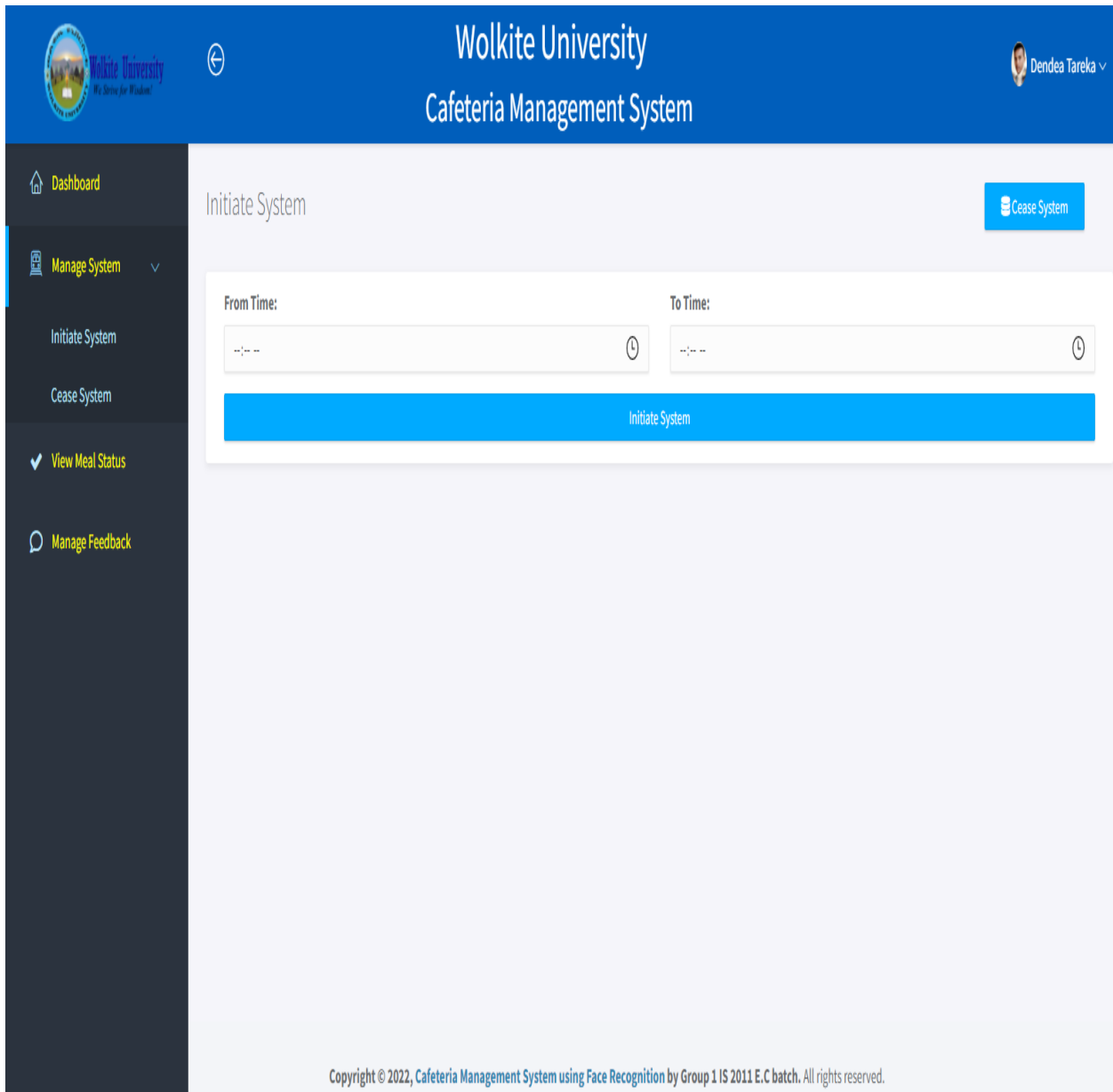


Figure 5. 12 Initiate system page

Wolkite University
Cafeteria Management System

Dheresa Amante

Dashboard
Manage Cafe Stud
View Report
Manage Feedback

Student Dean Feedback

+ Send Feedback

Feedback ID	User ID	Title	Content	Sended On	Reply
1	Manager/9	about account	when i need to change our password, it response error to me	2/25/2022	Reply
2	Stud_union/13	about account	when i need to update our profile, it response error to me	2/26/2022	It may be corrected at moment.
3	Manager/16	about account	when i need to change our password, it response error to me	2/25/2022	Reply

Copyright © 2022, Cafeteria Management System using Face Recognition by Group 1 IS 2011 E.C batch. All rights reserved.

Figure 5. 13 Manage feedback page

CHAPTER SIX

6. IMPLEMENTATION AND TESTING

Implementation phase is the most essential stage to develop the required system. It is the phase in which the design and analysis of the system are transformed into a tangible system. This is done by writing code and makes it operational.

6.1. Implementation of the Database

We use PostgreSQL Database PostgreSQL is an advanced, enterprise-class, and open-source relational database system. PostgreSQL supports both SQL (relational) and JSON (non-relational) querying. PostgreSQL is primarily used as a robust back-end database that powers many dynamic websites and web applications. It allows to store large and sophisticated data safely and help developers to build the most complex applications, run administrative tasks and create integral environments. It is more secure, cheap and scalable.

6.2. Implementation of the Class Diagram

The implementation of the system is based on the design of the class diagram and the business logic with the targeted programming language. The attributes are properly defined with their data types and access visibilities (modifiers). Methods are also defined appropriately with their return types, parameters, and data types including access visibility.

6.3. Configuration of the application server

In this project, Django is used as an application server because it is a free, open-source framework and provides speed and performance as well as good security for developers. It offers interactive shortcuts for full integration with the system databases.

- ✓ Folders and files are properly organized.
- ✓ The application server properly starts and shutdowns.
- ✓ Its configuration enables remote access to the system.

6.4. Configuration of Application Security

We used multi-level Authentication. Multi-level authentication implies different levels of permissions for different users. Each authenticated user is leveled differently and we set up users as System Admin, Cafeteria Manager, Student Dean, Student Controller, Registrar

Officer, and Student Union. We have implemented all input validations properly in order to secure our system. When the user inputs invalid data or leave it empty, the system notifies the user to enter valid input.

Django allows many application security mechanisms. From those application securities, some of them are discussed below.

a) Cross-site scripting (XSS) protection

Django's template system protects you against the majority of XSS attacks by escaping specific characters that are "dangerous" in HTML. When you give input, Django forcefully encodes it and then escapes dangerous characters.

b) Cross-site request forgery (CSRF) protection

The most robust way to defend against CSRF attacks is to include a CSRF token within relevant requests. The token should be tied to the user's session and strictly validated in every case before the relevant action is executed. It works by checking for a secret in each POST request.

6.5. Implementation of User Interface

The user interface of this system is implemented based on the design. It is clear and interactive; as much as possible complexity is handled in this user interface:

- ✓ Consistent and stable which does not create any confusion for users.
- ✓ Can be easily understandable with clear and steady navigation.
- ✓ Consistent and dependable.
- ✓ Attractive and eye-catching for users.
- ✓ Is compatible, well-matched, and friendly.
- ✓ It reduces the memory loads of the users.

6.6. Testing the System

The functionality of the developed system must be tested because it has to provide its intended goal or functions. The following features are tested:-

- ✓ Input and output function.

- ✓ Subsystem communication.
- ✓ Database and application server communication.
- ✓ Databases and user interface interactions.
- ✓ Graphical user interface.
- ✓ Validations and authentication.
- ✓ Security mechanisms were only authorized users are allowed to access our system.

6.6.1. Test Case

Sample Test Cases

Face Recognition needs the following:

- ✓ Student's face image

Test case 1: check student identification and recognition.

Test case objective: to predicate and recognize.

Test case description: first student controller enters the cafeteria food service schedule, then presses the initiate system button, then the webcam is opened automatically. Then the system detects and predicates the student's faces. After predicting the student's face, it gets the student's name from the image dataset, then searches by student's name from the PostgreSQL, then mark the single cafeteria food service schedule (breakfast, lunch, or dinner). Finally, the student is authorized to access the cafeteria food service.

Requirements Verified: Yes

Test Environment: PostgreSQL, Webcam, and Django server must be in running state, Database Should contain appropriate table and link must be established between server and client program.

6.6.2. Testing tools and Environments

In this project different testing tools are used for testing the system development process progress and for debugging purposes.

Software testing tools:

- ✓ Windows 10 operating system.
- ✓ Django Framework.
- ✓ Python programming language.
- ✓ Python Open-CV.
- ✓ Webcam.
- ✓ Web browsers (Chrome, Microsoft edge).

Hardware testing tools:

- ✓ Computer(Dell desktop, HP laptop)
- ✓ Laptop Camera
- ✓ Internet cable

6.6.3. Unit testing

We did unit testing at the coding level to test the functional correctness of the independent modules. We used it to isolate a section of codes, verify the correctness of the code, the functions, and procedures of the system, fix and debug in the development life cycle , and helped us to understand the code base, reuse code, and enabled us to make quick changes in quality to our project. Generally, we used unit testing to correct code errors at the right time.

6.6.4. System testing

We applied System testing at a high-level design to ensure the functional and non-functional requirements' correctness, performance, accuracy, functionality validations, and security have been met correctly. We tested the complete and fully integrated software product. We examined how the whole subsystems work together to achieve the desired goal of the user's requirements from the system.

6.6.5. Integration testing

At low-level design, we used integration testing to examine how the different procedures work together to achieve the goal of the system. Check the interaction between individual functionality which performs the specific task. All the screen functions, store data tables, and other modules were connected with seamless interfacing. All the required outputs were

produced successfully as expected from the systems and inputs were validated and stored in the correct formats. In this second-level testing technique, we tested the interfacing issues between the modules and we make sure that this project has no interfacing issues between the related modules.

6.7. Model Development

Developing tensor flow models that can classify the Students who get food service from the Cafeteria.

6.7.1. Convolutional Layer

Convolutional layers are the main building blocks used in a convolutional neural network. A convolution is a mathematical term that describes a dot product multiplication between two sets of elements. Within deep learning the convolution operation acts on the filters/kernels and image data array within the convolutional layer [5].

Therefore a convolutional layer is simply a layer that houses the convolution operation that occurs between the filters and the images passed through a convolutional neural network. Convolutional neural networks apply a filter to input to create a feature map that summarizes the presence of detected features in the input. “Invariance to local translation can be a very useful property if we care more about whether some feature is present than exactly where it is.

We use conv2d with the same padding and relu activation Conv 2d filter is two- dimensional. In our model definition context, we have 6 convolutional layers.

```
#Initializing the model
```

```
model = Sequential()
```

```
# Convolution Step 1
```

```
model.add(Convolution2D(32, 3, padding='same', activation='relu', input_shape=(224,224, 3))
```

6.7.2. Max Pooling Layer

Max pooling is a variant of sub-sampling where the maximum pixel value of pixels that fall within the receptive field of a unit within a sub-sampling layer is taken as the output. Max

pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter above it the max-pooling operation below has a window of 2x2 and slides across the input data, outputting an average of the pixels within the receptive field of the kernel.

```
# Max Pooling Step 1
```

```
model.add(MaxPooling2D(pool_size = (2, 2)))
```

6.7.3. Flatten layer

Flatten layer Takes an input shape and flattens the input image data into a one-dimensional array. After the execution of all the convolutional layers we have to flatten it into one-dimensional array to create the pooled feature map to a single long feature vector after doing the convolutional layers and max-pooling and dropouts for handling over-fit flattening the feature comes next to our TensorFlow model of classification.

```
# Flattening Step1
```

```
model.add(Flatten())
```

6.7.4. Dense Layer (Fully connected layer)

On the dense layer each neuron in the dense layer receives input from all neurons of its previous layer this helps the model to learn features from all the combination of the features of the previous layers we have a dense layer with softmax activation and at last, we have a dense layer with the class size of the dataset. A dense layer has an embedded number of arbitrary units/neurons within each neuron is a perceptron.

```
# Full Connection Step1
```

```
No_class = 3
```

```
model.add(Dense(256, activation = 'relu'))
```

```
model.add(Dense(No_class, activation = 'softmax'))
```

6.7.5. Train the network compilation processes

To train the network, we have to compile it. The compilation processes involve specifying the following parameters:

1. Define Network.

The first step is to define a neural network. Neural networks are defined in Keras as a sequence of layers. The container for these layers is the Sequential class. In our case, we use a sequential model from the TensorFlow library.

The first layer in the network must define the number of inputs to expect. Our model network has 1 input layer, 5 hidden layers, and 1 output layer. The input layer takes the dataset and changes it to the shape we gave it and transforms its color range from 0-255 to 0-1. After the hidden layers, we flatten images from 2d-array to a 1d- array.

2. Compile Network.

Once we have defined our network, it must compile. The compilation is an efficient step. It transforms the simple sequence of layers that we defined into a highly efficient series of matrix transforms in a format intended to be executed depending on how Keras is configured. Whether it's CPU or GPU.

Compilation requires several parameters to be specified, specifically tailored to training your network. Specifically, the optimization algorithm to use to train the network, and the loss function is used to evaluate the network that is minimized by the optimization algorithm. In our case, we use the Adam optimization algorithm that requires the tuning of the learning rate.

3. Fit Network.

Once the network is compiled, it can be fit, which means adapting the weights on a training dataset. The network is trained using the back propagation algorithm and optimized according to the optimization algorithm and loss function specified when compiling the model [6]. The back propagation algorithm requires that the network be trained for a specified number of epochs or exposures to the training dataset.

4. Evaluate Network.

We can evaluate the performance of the network on a separate dataset, unseen during testing. This would provide an estimate of the performance of the network in making predictions for unseen data in the future [6].

5. Make Predictions.

Finally, once we are satisfied with the performance of our fit model, we can use it to make predictions on new data.

Deployment is the method by which we integrate a machine learning model into an existing production environment to make practical identification based on data. It is one of the last stages in the machine learning life cycle and can be one of the most. To start using a model for practical prediction, it needs to be effectively deployed into production.

6.7. Activation function

Rectifier Linear Function ReLU activation functions that transform a summed signal from each neuron in a layer can be extracted and added to the Sequential as a layer-like object called Activation [5].

6.8. Loss function

A method that quantifies „how well“ a machine learning model performs. The quantification is an output (cost) based on a set of inputs, which are referred to as parameter values. The parameter values are used to estimate a prediction, and the “loss” is the difference between the predictions and the actual values [5]. Since we have many classes its Multi classification so we use categorical_crossentropy from TensorFlow, and Keras library.

6.9. Optimization Algorithm

For optimizer we used Adam instead of SGD. Adam is faster than SGD, even though SGD is more accurate than Adam [5].

6.10. Learning Rate

The amount that the weights are updated during training is referred to as the step size or the learning rate. Learning rate is a configurable hyper parameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0.

The learning rate controls how quickly the model is adapted to the problem. “The learning rate is perhaps the most important hyper parameter. If you have time to tune only one hyper parameter, tune the learning rate. The optimization initial learning rate is 0.00001 because if the learning rate is small it can fall to the local minimum more precisely.

6.11. Summary () function

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
conv2d_1 (Conv2D)	(None, 220, 220, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 110, 110, 32)	0
conv2d_2 (Conv2D)	(None, 108, 108, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_3 (Conv2D)	(None, 52, 52, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
conv2d_4 (Conv2D)	(None, 24, 24, 256)	295168
max_pooling2d_3 (MaxPooling2D)	(None, 12, 12, 256)	0
conv2d_5 (Conv2D)	(None, 10, 10, 512)	1180160
max_pooling2d_4 (MaxPooling2D)	(None, 5, 5, 512)	0
flatten (Flatten)	(None, 12800)	0
dense (Dense)	(None, 512)	6554112
dense_1 (Dense)	(None, 256)	131328

dense_2 (Dense)	(None, 128)	32896
dense_3 (Dense)	(None, 64)	8256
dense_4 (Dense)	(None, 32)	2080
dense_5 (Dense)	(None, 16)	528
dense_6 (Dense)	(None, 3)	51

```

=====
Total params: 8,307,075
Trainable params: 8,307,075
Non-trainable params: 0

```

A total of 8,307,075 params were found from our tensor flow sequential classification model and all are trainable.

6.12. Epoch

The number of epochs is a hyper parameter that defines the number of times that the learning algorithm would work through the entire training dataset. 100 epoch is used to train this classification model.

6.13. Discussion of the Result

After preprocessing the dataset, the final dataset consisted of a total of 5430 images for three students. For training, validation, and testing the proposed CNN, the dataset was partitioned into three subsets. The training dataset contained 3900 students' images. The testing dataset similarly contained 1080 images and the validation dataset contained 450 images.

As mentioned in the proposed architecture of the CNN model, it consisted of 19 layers in which 1 input layer, 5 are convolutional, 5 max-pooling layers, 1 flattening layer, 6 fully connected layers (dense), and 1 output layer. The CNN model thus achieved an extraordinary performance with an accuracy of 92% with the test data subset used from the processed dataset.

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

7.1. CONCLUSION

Deep learning methods have recently become popular for computer vision and image identification. Computer vision is a field that deals with methods for acquiring, processing, analyzing, and understanding images. Deep convolutional neural networks (CNNs) have seen great success in the past few years on a variety of machine learning problems. Deep Convolutional Neural Networks have been applied for an end to end learning.

In this project, we build Cafeteria Management System using Face Recognition to recognize and identify a face of a student together with a web feature at the cafeteria entrance. This allows a student to get food services only using his/her physiological features (his face in this case) which do not have the possibility of getting lost.

In this project, we used a deep learning approach for the identification and recognition of the students. The complete procedure was described, from collecting the images used for training, testing, and validation to image preprocessing and augmentation, and finally the procedure of training the deep CNN.

The image data was collected from the student's face purposively. The dataset was preprocessed by image resizing. The model was developed by using Tensor Flow, Keras, Matplotlib, Numpy, and OpenCV tools. Furthermore, the input image size was necessarily resized to 224 x 224 for Deep Convolutional Neural Network (CNN) due to hardware limitations. Deep Learning models were trained by using Adam deep learning optimizers.

Finally, we developed a Machine Learning System that identifies and recognizes the students at the cafeteria entrance. We expect that our Machine Learning system will make a significant contribution to the Cafeteria Management System at the cafeteria entrance.

7.2 RECOMMENDATION

Our project team developed a web-based Cafeteria management system integrated with Deep Learning to be used at the cafeteria entrance for the students who access the food service from the cafeteria.

We have used our maximum potential to include the basic functionalities in the problem area. But problems are broad in nature and are constrained by time. Due to this reason, we may have missed out some other functionalities such as adding non café students into the system, recording, managing the materials needed in cafeterias, and managing the type and way of food being served in the cafeterias So, we recommend future developers to include some or all of these functionalities on top of what we already did in this project.

8. REFERENCES


- [1] R. G. Booch, Object-oriented Analysis and Design with Application, Boston: Pearson Education, 2007.
- [2] "Face detection with MTCNN," [Online]. Available: <https://towardsdatascience.com/robust-face-detection-with-mtcnn-400fa81adc2e>.
- [3] "Face detection using MTCNN," [Online]. Available: <https://medium.com/swlh/face-detection-using-mtcnn-part-2-d3db8abf3047>.
- [4] D. Scragg, The Object Primer, Cambridge: Cambridge university press, 2004.
- [5] "Image classification using CNN," [Online]. Available: <https://learnopencv.com/image-classification-using-convolutional-neural-networks-in-keras/>.
- [6] "Train model using tensorflow CNN," [Online]. Available: <https://medium.com/analytics-vidhya/step-by-step-train-model-using-tensorflow-cnn-2f243b847c8>.



9. APPENDICES

Appendix I: Manual books of the fresh student.



Appendix II: Digital ID of the senior student.

Wolkite University  **Registrar Office**

 +251113220135 fax:+251113220041
 07 Wolkite <http://www.wku.edu.et>

Regular Student Identification Card

Name: DEJENE TOLERA FEYISA

College: College of Computing and Informatics

Department: Information Systems

Address: Wolkite ID.NO: CIR/136/11

