



**WOLKITE UNIVERSITY**  
**COLLEGE OF COMPUTING AND INFORMATICS**  
**DEPARTMENT OF SOFTWARE ENGINEERING**  
**WOLKITE UNIVERSITY EMPLOYEE PERFORMANCE**  
**APPRAISAL SYSTEM**  
**INDUSTRIAL PROJECT**

BY

<b>NAME OF THE STUDENTS</b>	<b>ID NO</b>
BIRUK BIRHANU	NSR/0374/12
BISRAT JENBERE	NSR/0385/12
EBENEZER MULU	NSR/0473/12

PROJECT ADVISOR: Mr. Debebe H.

Wolkite University, Wolkite, Ethiopia

Jan 25, 2024

**WOLKITE UNIVERSITY**  
**COLLEGE OF COMPUTING AND INFORMATICS**  
**DEPARTMENT OF SOFTWARE ENGINEERING**  
**WOLKITE UNIVERSITY EMPLOYEE PERFORMANCE**  
**APPRAISAL SYSTEM**

SUBMITTED TO DEPARTMENT OF SOFTWARE ENGINEERING IN  
PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE  
DEGREE OF BACHELOR OF SCIENCE IN SOFTWARE  
ENGINEERING

BY

<b>NAME OF THE STUDENTS</b>	<b>ID NO</b>
BIRUK BIRHANU	NSR/0374/12
BISRAT JENBERE	NSR/0385/12
EBENEZER MULU	NSR/0473/12

PROJECT ADVISOR: Mr. Debebe H.

Wolkite University, Wolkite, Ethiopia

Jan 25, 2024

# DECLARATION

This is to declare that this project work which is done under the supervision of Mr. Debebe H and having the title Employee Performance Appraisal System is the sole contribution of:

- ❖ Bisrat Jenbere
- ❖ Biruk Birhanu
- ❖ Ebenezer Mulu

No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven. Date: Jan 25, 2024

## Group Members

Full Name

Signature

_____	_____
_____	_____
_____	_____

## Approval Form

This is to confirm that the project report entitled Web Based Internship Management System submitted to Wolkite University, College of Computing and Informatics Department of Software Engineering by: Bruk Birhanu, Bisrat Jenbere, and Ebenezer Mulu is approved for submission.

Advisor Name	Signature	Date

Department Head Name	Signature	Date

Examiner 1 Name	Signature	Date

Examiner 2 Name	Signature	Date

Examiner 3 Name	Signature	Date

# Table of Contents

DECLARATION .....	i
Approval Form.....	ii
ACKNOWLEDGEMENT .....	ix
ABBREVIATION.....	x
ABSTRACT.....	xi
CHAPTER ONE .....	1
INTRODUCTION .....	1
1.1 Background of the Organization .....	2
1.1.1. Mission.....	2
1.1.2. Vision.....	2
1.1.3 Core Values.....	2
1.2 Statements of problem .....	3
1.3 Objective .....	4
1.3.1 General Objective .....	4
1.3.2 Specific Objective .....	4
1.4 Feasibility Study .....	4
1.4.1 Ethical and Legal Feasibility.....	4
1.4.2 Technical Feasibility .....	4
1.4.4 Economic Feasibility.....	5
1.5 Scope and Limitation .....	5
1.5.1 Scope.....	5
1.5.2 Limitations .....	6
1.6 Significance of the Project .....	6
1.7 Beneficiary of the Project .....	6
1.8 Methodology .....	7
1.8.1 Data collection .....	7
1.8.2 Process Model .....	9
1.8.3 Analysis and Design .....	9
1.8.4 Development tools and technologies .....	10
1.8.5 System Testing Methodology .....	10
CHAPTER TWO .....	13
DESCRIPTION OF THE EXISTING SYSTEM.....	13
2.1 Introduction of Existing System .....	13
2.2 Users of Existing System .....	14

2.3 Major Functions of the Existing System .....	14
2.4. Forms and Other Documents of the Existing Systems .....	16
2.5. Drawbacks of the Existing System .....	22
2.6. Business rules of the existing system.....	23
PROPOSED SYSTEM .....	24
3.1 Functional Requirements .....	24
3.4. Non-functional Requirements .....	26
3.4.1 User Interface and Human Factors.....	26
3.4.2 Performance Consideration.....	26
3.4.3 Security Issues.....	27
3.4.4 Error Handling and Validation.....	27
3.4.5 Quality Issues .....	28
3.4.6 Resource Issues .....	28
3.4.7. Physical Environment .....	29
3.4.8. Backup and Recovery .....	29
3.4.9. Documentation .....	29
CHAPTER FOUR.....	30
SYSTEM ANALYSIS .....	30
4.1 System Model .....	30
4.1.1 Use Case Model .....	31
4.1.1.1 Use Case Diagrams .....	34
4.1.1.2 Use Case Description .....	35
4.1.1.3 Use case Scenario .....	48
4.2. Object Model.....	52
4.2.1. Class Diagram .....	53
4.2.2. Data Dictionary .....	55
4.3. Dynamic Model.....	59
4.3.1. Sequence Diagram .....	59
4.3.2 Activity Diagram.....	61
4.3.3 State Chart Diagram.....	65
CHAPTER FIVE.....	68
SYSTEM DESIGN .....	68
5.1. Design Goals .....	68
5.1.1. Performance Consideration.....	68
5.1.2 Security Issues.....	69

5.1.3 Quality Considerations.....	69
5.2 Proposed System Architecture .....	70
5.2.1. Subsystem Decomposition and Description .....	71
5.2.2. Hardware/Software Mapping.....	73
5.2.3. Detailed Class Diagram .....	74
5.2.4. Persistent Data Management.....	75
5.2.5. Access control and Security .....	77
5.3. Packages.....	79
5.4. Algorithm Design.....	80
5.5. User Interface Design.....	82
CHAPTER SIX.....	84
IMPLEMENTATION AND TESTING.....	84
6.1. Implementation of the Database .....	84
6.1.1 Table Creation and Constraints.....	84
6.1.2. Normalization.....	87
6.1.3. Database Level Security.....	87
6.2. Implementation of the Class Diagram .....	88
6.3. Configuration of the Application Server.....	96
6.4 Configuration of Application Security.....	96
6.7 Testing.....	98
6.7.1. Test Cases .....	98
CHAPTER SEVEN.....	104
CONCLUSION AND RECOMMENDATION .....	104
7.1 Introduction.....	104
7.2 Conclusion .....	104
7.3 Recommendations .....	105
References.....	106
APPENDICES .....	107
Appendix I: sample source code .....	107
Appendix II: sample Test Script .....	112
Appendix III: Interview question prepared .....	114
Appendix IV: Existing System Forms and Reports .....	117

## List of Tables

Table 4. 1 Define evaluation criteria .....	35
---	----

Table 4. 2 Request Evaluation .....	35
Table 4. 3 Evaluate Self.....	36
Table 4. 4 Evaluate Peers.....	37
Table 4. 5 Evaluate Subordinate Employee.....	38
Table 4. 6 Evaluate Instructors .....	39
Table 4. 7 Approve the Evaluation Result.....	40
Table 4. 8 Assign Course.....	41
Table 4. 9 View Appraisal Result.....	41
Table 4. 10 Access Employee Information.....	42
Table 4. 11 Manage User.....	43
Table 4. 12 Manage Instructor.....	44
Table 4. 13 Manage Departments .....	45
Table 4. 14 Login use case description.....	46
Table 4. 15 Logout.....	47
Table 4. 16 Users .....	55
Table 4. 17 Department .....	56
Table 4. 18 College.....	56
Table 4. 19 Course .....	56
Table 4. 20 Appraisal Cycle .....	57
Table 4. 21 Evaluation Template.....	57
Table 4. 22 Appraisal.....	58
Table 5. 1 Access control	77
Table 6 1 Login Test case (Valid credentials)	98
Table 6 2 Login Test case (Invalid credentials).....	99
Table 6 3 Add User Test Case (Correct Input) .....	99
Table 6 4 6 Evaluation Test Case (Rating Left) .....	100
Table 6 5 Evaluation Test Case (No Active Cycle).....	101
Table 6 6 Evaluation Test Case (deadline passed) .....	102
Table 6 7 View Result Test Case (without Evaluating All His Peer).....	102
Table 6 8 View Result Test Case.....	103

## List of Figures

Figure 2. 1 Instructor Evaluation Form (By Student).....	17
Figure 2. 2 Instructor Evaluation Form (By Head).....	18
Figure 2. 3 Instructor Evaluation Form .....	19
Figure 2. 4 Administrative Employee Evaluations form (By Head and Peer) .....	20
Figure 2. 5 Administrative Employee Evaluations form (By minute).....	22
Figure 4. 1 Use Case Diagram .....	34
Figure 4. 2 high level class diagram .....	54
Figure 4. 3 Sequence Diagram (Self-Evaluation) .....	59
Figure 4. 4 Sequence Diagram (Peer Evaluation).....	60
Figure 4. 5 Activity Diagram (Define Evaluation rules.....	61
Figure 4. 6 Activity Diagram (Evaluation) .....	62
Figure 4. 7 Activity Diagram (View Appraisal Result).....	63
Figure 4. 8 Activity Diagram (Manage user).....	64
Figure 4. 9 State Diagram (Access Employee Information).....	65
Figure 4. 10 State Diagram (Assign Course).....	66
Figure 4. 11 State Diagram (evaluation Process) .....	67
Figure 5. 1 System architecture .....	70
Figure 5. 2 Component Diagram .....	72
Figure 5. 3 Hardware /Software mapping.....	73
Figure 5. 4 Detailed Class Diagram.....	74
Figure 5. 5 Persistence Data Model .....	76
Figure 5. 6 Package diagram.....	79
Figure 5. 7 View Appraisal Result.....	80
Figure 5. 8 Perform Evaluation.....	81
Figure 5. 9 Login page design .....	82
Figure 5. 10 Dashboard design .....	83

Figure 6 1 AppraisalCycle Model .....	88
Figure 6 2 Appraisal Model .....	89
Figure 6 3 User Model .....	90
Figure 6 4 Complaint Model.....	91
Figure 6 5 Course Model .....	92
Figure 6 6 College Model .....	93
Figure 6 7 Appraisal Template Model .....	94
Figure 6 8 Department Model.....	95

## **ACKNOWLEDGEMENT**

We express our profound gratitude to God for His continuous support, guiding us through the completion of our proposal and the comprehensive documentation of our project. Without His divine assistance, the realization of WKU Performance Appraisal System would not have been achievable.

We extend our heartfelt thanks to our dedicated advisor, Instructor Mr. Debebe H, for his invaluable support and guidance throughout this journey. His insights, feedback, and commitment have been instrumental in shaping the outcome of our project.

Additionally, we would like to acknowledge the pivotal role played by the instructors of the Software Engineering department, who provided the framework for the successful execution of our project. Their expertise and dedication have been a source of inspiration. Special thanks are due to our friends who generously shared their advice and corrections, contributing to the refinement of our work.

We are grateful for the collaborative spirit within the Software Engineering community that has made this project possible.

Thank you.

## **ABBREVIATION**

API: Application Programming Interface

BR: Business Rules

CRUD: Create, Read, Update, Delete

CPU: Central Processing Unit

GUI: Graphical User Interface

HR: Human Resources

HTTP: Hypertext Transfer Protocol Secure

ICT: Information Communication and Technology

OOA: Object-Oriented Analysis

OOD: Object-Oriented Design

OOSAD: Object-Oriented System Analysis and Design

OO: Object-Oriented

PAS: Performance Appraisal System

RAM: Random Access Memory

UML: Unified Modeling Language

WKU: Wolkite University

3-tier architecture: Three-tier architecture

## **ABSTRACT**

The proposed Performance Appraisal System (PAS) for Wolkite University represents a significant advancement in evaluating employee performance. This system aims to enhance the current appraisal process by meticulously gathering and documenting user requirements from employees, managers, and HR professionals. By addressing organizational needs, the PAS will introduce innovative solutions, including customized evaluation criteria and automated notifications.

In the development phase, the system will incorporate modules for data management, user access, and role management. This encompasses the creation of essential components such as data storage, user login capabilities, and performance report accessibility. The proposed system expands both in functionality and scope, introducing novel features absent in the existing system.

Key features of the PAS include a complaint submission feature. This inclusion ensures a participatory approach, emphasizing employee feedback and concerns. Additionally, administrative employees, previously excluded, are now incorporated into the performance appraisal process, fostering a more comprehensive evaluation of all roles within the organization.

One notable advancement is the system's accessibility to both HR professionals and employees. This democratization of access empowers employees to take ownership of their performance and development goals, fostering a culture of accountability and active participation.

# **CHAPTER ONE**

## **INTRODUCTION**

An appraisal is a systematic process through which an employee's performance is methodically documented and assessed within an organizational context. This evaluation typically involves a comprehensive examination of an employee's job-related achievements, skills, and behaviors, often over a set period, and is often used to gauge their effectiveness in meeting job expectations and contributing to the organization's goals. Appraisals serve as a valuable tool for both employees and employers, offering constructive feedback, identifying areas for improvement, recognizing accomplishments, and informing decisions related to promotions, compensation, and career development. This process is critical for maintaining accountability and fostering professional growth within the workplace.

Performance appraisal system software can revolutionize the performance appraisal process by automating many of the tasks involved, providing a centralized platform for all of the activities involved, and providing a standardized and data-driven approach to performance evaluation. The main purpose of an EPAS is to provide employees with feedback on their performance, identify areas for improvement.

Wolkite University is a higher educational institution that consists of many manual and automated processes. Performance appraisal system is a manual system. The current performance appraisal process was largely manual and paper-based. This could be a time consuming and inefficient process, and it could also be prone to errors. Additionally, manual performance appraisals were often subjective and could be biased. Owing to this, we have selected the Wolkite University (WKU) performance appraisal system to automate the system and this automated performance appraisal System is a system used to perform appraisal activity in a computerized way to overcome the above listed problems. This results in reducing misuse of manpower, avoiding errors, saving time and providing comfort in performance evaluation system

## **1.1 Background of the Organization**

### **1.1.1. Mission**

The University as indicated in the proclamation No. 650/2009, has the following Missions

- ❖ To produce graduates who are Knowledgeable, attitudinally mature and practical innovative.
- ❖ To support relevant and demanded technologies and knowledge that address nation and community level development problems to help make operation of the government and non-governmental organizations efficient, effective and competitive.
- ❖ To provide training and consulting services to the community and the government.

### **1.1.2. Vision**

WKU aspires to be one of the leading applied science universities in Ethiopia by 2022E.c

### **1.1.3 Core Values**

- ❖ Excellence
- ❖ Inclusiveness
- ❖ Truth of integrity
- ❖ Being a learning organization
- ❖ Accountability
- ❖ Academic freedom

## 1.2 Statements of problem

Due to various factors, including manual data entry processes, inadequate data management tools, and a lack of streamlined integration between the evaluation system and employee profiles, a lot of problems are encountered which include issues of inclusiveness, efficiency, and resource management.

- ❖ **Inclusiveness:** The present system lacks inclusiveness, as it does not include the evaluation of all roles and employees effectively. For instance, the evaluation of administrative employees remains a manual process
- ❖ **Difficulty in Analyzing Employee Performance:** A significant obstacle is encountered when HR and other administrators seek the need to analyze employee performance, especially in cases of promotions or role transitions. The current system lacks the capability to provide an efficient and comprehensive overview of employee performance, thus impeding informed decision-making.
- ❖ **Time-Consuming Distribution and Collection:** The distribution and collection of performance criteria and evaluation results pose significant time constraints and inefficiencies. Given the need to encompass all roles from HR professionals to individual employees, the process becomes laborious and time-intensive.
- ❖ **Manual Calculation for HR:** Human resources faces considerable challenges when attempting to manually calculate performance evaluations based on predefined criteria. This manual approach is error-prone, time-consuming, and resource intensive.
- ❖ **High Costs:** The existing system relies heavily on paper-based documentation, resulting in excessive resource consumption and high operational costs.
- ❖ **Manual Data Entry:** The current process requires manual data entry to input employee evaluation results into their individual profiles. This manual approach is time-consuming and error-prone, leading to delays and potential inaccuracies.

## **1.3 Objective**

### **1.3.1 General Objective**

The general objective of this project is to develop a software that will automate the Performance Evaluation Process at WKU.

### **1.3.2 Specific Objective**

- ❖ To gather and document user requirements for the Performance Appraisal System
- ❖ To identify potential challenges and areas for improvement in the existing appraisal process
- ❖ To encode the selected proposed solutions into the system
- ❖ To develop a data definition and manipulation module for employee performance data.
- ❖ To develop a user management module for system access
- ❖ To develop an account and role management system for role-based access
- ❖ To conduct comprehensive testing using various testing techniques

## **1.4 Feasibility Study**

To ensure our project's viability, we have conducted a thorough feasibility study encompassing various key aspects.

### **1.4.1 Ethical and Legal Feasibility**

Our project adheres to high ethical standards and legal compliance. It has been designed with utmost care to avoid any negative impact on the moral values of the community. We have also taken extensive measures to ensure that the project does not pose safety issues or any social and psychological threats to the well-being of the society at large.

### **1.4.2 Technical Feasibility**

In terms of technical feasibility, all team members possess the necessary skills to analyze and design complex systems. Moreover, they are well-versed in utilizing both front-end and back-end frameworks effectively, ensuring the successful completion of the project.

Operational feasibility revolves around whether the organization has the capacity to operate the system efficiently. WKU is well-equipped to successfully manage and operate the system. The university boasts a comfortable environment, including full ICT coverage,

which can seamlessly host and configure our project for optimal performance and service delivery.

#### **1.4.4 Economic Feasibility**

Our project is economically feasible, and it promises to be profitable for the organization. This profitability arises from its ability to significantly reduce the consumption of valuable resources and minimize manual processes. Moreover, it saves valuable time, making it a cost-effective solution. The project aligns with the organization's goal of sustainability and cost-efficiency, making it a financially viable endeavor.

### **1.5 Scope and Limitation**

#### **1.5.1 Scope**

##### **User Management and Role-Based Permission System**

The appraisal system will include a user management and role-based permission system to control access to the system and its features. Administrators will be able to create, edit, and delete user accounts, as well as assign users to different roles and groups. Roles will be defined by the specific actions that users are allowed to perform within the system. For example, HR role have the ability to create and edit appraisal templates and settings

##### **Peer Evaluation, Supervisor Evaluation, and Self-Evaluation**

The appraisal system will support peer evaluation, supervisor evaluation, and self-evaluation. Employees will be able to provide feedback on their peers' performance, and supervisors will be able to assess their employees under their management. Employees will also have the opportunity to self-assess their own performance.

##### **Student Evaluation for Academic Employees**

For academic employees, the appraisal system will also support student evaluation. Students will be able to provide feedback on the academic performance and teaching effectiveness of their professors.

## **Required Report Generation**

The appraisal system will generate a variety of required reports, including individual appraisal reports, team appraisal reports, and department appraisal reports. These reports will provide valuable insights into employee performance and areas for improvement.

## **Attaching the Result Calculated to Employee Profile**

The appraisal system will automatically attach the result calculated for each employee to their employee profile. This will make it easy for employees to view their appraisal results and track their progress over time.

## **Dynamic Appraisal Template Preparation**

The appraisal system will allow administrators to create and edit dynamic appraisal templates. This will enable administrators to tailor the appraisal process to the specific needs of their organization

### **1.5.2 Limitations**

- ❖ The appraisal system will not be integrated with other HR management systems.
- ❖ The appraisal system will not provide data analysis capabilities.

## **1.6 Significance of the Project**

The significance of this project lies in its capacity to address the existing problems in the current system. Due to the ongoing shift from manual systems to a computerized approach. The impact of this project extends to a range of stakeholders, benefiting various parties within our organization.

## **1.7 Beneficiary of the Project**

### **Employees**

- ❖ Self-assess their achievements and contributions.
- ❖ Understand their career growth prospects based on their performance.
- ❖ Enhance their motivation and job satisfaction through recognition and rewards.

### **Head and Team Leader**

- ❖ Simplify distribution and collection of evaluation criteria to Department Employees.
- ❖ Streamline report preparation after the appraisal period.
- ❖ Simplify overall appraisal process management, including participant selection and gathering student feedback on instructors.
- ❖ Facilitate a seamless approval or rejection of evaluation results through a single-button click.

### **Dean and Director**

- ❖ Streamline overseeing the college-level evaluation process.
- ❖ Simplify the process of generating reports at the college level.
- ❖ Simplify the management of department heads and team leaders.

### **Human Resources (HR)**

- ❖ Standardizing performance evaluation processes across the organization.
- ❖ Managing and storing performance-related data.
- ❖ Ensuring compliance with performance-related policies and regulations.
- ❖ Facilitate the seamless incorporation of dynamic criteria with ease.
- ❖ Enable easy access to evaluation results for efficient decision-making.

### **Organizational Leadership and Decision-Makers**

- ❖ Make informed decisions about resource allocation, promotions, and workforce planning.
- ❖ Identify patterns and trends in employee performance across the organization.

## **1.8 Methodology**

### **1.8.1 Data collection**

Accurate and comprehensive data gathering is the cornerstone of any successful project, especially when it comes to understanding the project's requirements. To ensure that we have a deep and detailed understanding of the project's needs, the project team will invest substantial effort in various activities, including interviews, observations, document reviews, and brainstorming sessions. These conventional techniques are crucial for

obtaining a holistic perspective on the project's requirements, ultimately setting the stage for a successful project execution.

## **Interviews**

The project team conducted interviews with a diverse group of stakeholders, including

- ❖ ICT Director: Valuable insights were gained to understand the technical and infrastructure needs of the project.
- ❖ HR Official: Essential information regarding staffing, skill requirements, and compliance issues related to the project was obtained.
- ❖ Quality Coordinator: The Quality Coordinator's involvement has already played a pivotal role in maintaining and enhancing the quality of the appraisal system.
- ❖ Department Head: Valuable insights were gathered to understand the unique challenges and expectations within their department.

## **Observation**

### **During this phase, the project team:**

- ❖ Observed the existing system: Analysis of how the current system operates, its strengths, weaknesses, and potential areas for improvement has been completed.

Identified limitations: Observations helped pinpoint any shortcomings, bottlenecks, or inefficiencies in the current processes, which have been addressed in the project's requirements.

## **Document Review**

The project team reviewed the following types of documents

- ❖ Employee History Documents: Examination of documents that track an employee's history within the organization, including performance evaluations and progress, has been completed.
- ❖ Final Evaluation Calculations: Analysis of how final evaluations are currently calculated has assisted in determining specific requirements related to performance measurement.

- ❖ Employee Evaluation Documents: Scrutiny of documents used for employee evaluations, such as self-evaluation forms, peer evaluations, and assessments from team leaders, has been completed.

## **Brainstorming**

The project team organized brainstorming sessions to

- ❖ Generate new ideas: Team members, along with relevant stakeholders, brainstormed to develop innovative solutions and ideas to meet project requirements.
- ❖ Problem-solving: Brainstorming was also used to identify potential challenges and find effective ways to address them within the project.

### **1.8.2 Process Model**

The project team will employ the Agile methodology, which promotes frequent collaboration and adaptability.

It plays a central role in shaping the solution because:

- ❖ It fosters creative problem-solving
- ❖ Encourages continuous improvement,
- ❖ Allows for real-time adjustments based on emerging insights and changing requirements.
- ❖ This dynamic approach enhances the project's responsiveness to the needs and expectations of stakeholders

### **1.8.3 Analysis and Design**

In This project, the team will employ the Object-Oriented System Analysis and OOSAD because it offers several advantages over other methodologies, including:

- ❖ Improved software quality: OOAD systems are typically more reliable, secure, and efficient than systems developed using other methodologies.
- ❖ Reduced development time and cost: OOAD can help to reduce development time and cost by promoting reuse and modularity.

- ❖ Increased maintainability: OOAD systems are easier to maintain and update than systems developed using other methodologies.
- ❖ Enhanced flexibility and adaptability: OOAD systems are more flexible and adaptable to change than systems developed using other methodologies.
- ❖ Object-Oriented Analysis OOA: In this initial phase, the team will focus on modeling the functions of the system. This involves creating a use case model to outline how the system functions. The team will also identify the various business objects involved, arrange them in an organized manner, and establish the relationships between these objects. Finally, the behavior of each object will be modeled, helping us understand how they interact within the system.
- ❖ OOD: Following the OOA phase, the team will move on to OOD. Here, the use case model will be refined to better align with the implementation environment. This phase involves modeling the interactions and behaviors of objects to support the use case scenarios effectively. The object model will be updated to reflect these refinements.

#### **1.8.4 Development tools and technologies**

##### **Front-end Technologies**

For our interface design, we will harness the power of React. These choices are rooted in their ability to create visually appealing, user-friendly interfaces that enhance the overall user experience.

##### **Backend Technologies**

In the backend development, we will rely on a stack of cutting-edge technologies. This stack includes Node.js, Express, Mongoose, and MongoDB. These technologies are selected for their robust capabilities in building a dynamic and efficient backend system that drives the functionality of our project.

#### **1.8.5 System Testing Methodology**

Before deploying WKU's performance appraisal system, our development team will conduct rigorous testing, ensuring both functional and non-functional aspects are thoroughly assessed. The project is built using Node.js and Express, and we will leverage specific tools associated with these technologies, along with a free testing tool where applicable.

### **Unit Testing**

- ❖ Evaluate code quality for individual components within the WKU performance appraisal system using Mocha (for Node.js).
- ❖ Verify the correct implementation of sub-procedures and functions.
- ❖ Identify and address errors early in the development cycle.

### **Integration Testing**

- ❖ Automate tests for integrated components within the WKU performance appraisal system using Supertest (for Express).
- ❖ Assess how different procedures collaborate to achieve the system's goals.
- ❖ Identify and address potential issues in component interactions.

### **System Testing**

- ❖ Conduct comprehensive testing of the entire WKU performance appraisal system using Apache JMeter.
- ❖ Identify and rectify errors and bugs unique to the WKU context.
- ❖ Ensure all system components function harmoniously.

### **Performance Testing**

- ❖ Measure the behavior of the WKU performance appraisal system under increasing loads using Apache JMeter.
- ❖ Identify bottlenecks and assess response times within the WKU environment.
- ❖ Ensure the system can handle expected user loads during peak times at WKU.

### **Compatibility Testing**

- ❖ Test the compatibility of the WKU performance appraisal system with specified operating systems, hardware platforms, web browsers, mobile devices, and other designated third-party programs using BrowserStack (free trial available).

## **CHAPTER TWO**

### **DESCRIPTION OF THE EXISTING SYSTEM**

#### **2.1 Introduction of Existing System**

Wolkite University's current performance appraisal system operates through integrated systems that combine the best of both manual and automated approaches, involving various stakeholders such as employees and multiple university departments. When an evaluation period commences, employees are required to engage in a series of steps to undergo performance appraisal.

Throughout the appraisal period, employees engage in assessment, reflecting on their achievements, challenges, and professional development needs. Simultaneously, Team Leaders conduct evaluations based on established criteria, considering factors like job specific tasks, collaboration, and adherence to organizational values. The system also incorporates assessment from peers and subordinates, contributing to a holistic view of an employee's performance.

However, the system relies heavily on manual documentation. This manual process poses challenges, making it time-consuming and potentially stressful for both employees and Team Leaders. Recognizing the complexity, there is a need for a more streamlined and technologically advanced performance appraisal system at Wolkite University. Such a system could automate data management and enhance the overall efficiency of the performance evaluation process. This evolution would contribute to a more transparent, fair, and supportive approach to employee development within the university.

## **2.2 Users of Existing System**

In WKU employee performance appraisal system, users typically refer to individuals involved in the evaluation and assessment process. These users play distinct roles in ensuring that the performance of employees is systematically and fairly assessed. Here are key user roles in WKU employee performance appraisal system:

### **Employees**

Integral to the appraisal process is the evaluation of employees against the backdrop of organizational values. Employees actively embody these values in their daily work, fostering a positive and values-driven work culture.

Employees at WKU are not passive recipients of the appraisal process; they are vital contributors and collaborators, actively shaping the success and effectiveness of the Employee Performance Appraisal System.

### **Students**

Student feedback is a valuable component of the appraisal system. Students often contribute to the evaluation process by providing input on the effectiveness of teaching methods, clarity of instruction, and overall learning experience. Their perspectives can offer valuable insights into the impact of an instructor's teaching style on the student learning outcomes.

### **Human Resources (HR)**

HR professionals may be involved in the administrative aspects of the appraisal system manually, such as coordinating the scheduling of appraisal meetings, maintaining records.

### **System Administrators**

Individuals responsible for the technical aspects of the appraisal system, including maintaining the software or platform used for documentation, tracking appraisal timelines, and ensuring data security and privacy.

## **2.3 Major Functions of the Existing System**

The major functions of the existing WKU employee appraisal system is designed to assess, enhance, and support the performance of the employee within the university.

Here are key functions typically associated with in the appraisal system:

**Student Feedback**

Gathering feedback from students through the existing appraisal system to assess the quality of instruction, communication, and the learning experience. Student feedback is valuable for understanding the impact of teaching on learners.

**Peer Reviews**

Facilitating peer evaluations where employees receive constructive feedback from colleagues. Peer reviews foster collaboration, the sharing of best practices, and the development of a supportive professional community.

**Administrative Assessments**

Involving academic administrators or Team leaders in the appraisal process. They may assess the employee contributions to overall goals, adherence to institutional policies, and alignment with the overall mission of the institution.

**Self-Assessment**

Providing the employee with the opportunity to reflect on their own working practices, strengths, and areas for improvement. Self-assessment encourages a sense of ownership and helps the employee to identify professional development needs.

**Documentation and Record Keeping:**

Maintaining accurate and confidential records of appraisal results manually. Documentation includes a historical record of an instructor's performance result.

## 2.4. Forms and Other Documents of the Existing Systems

ቀን-----

**የወልቀጤ ዩኒቨርሲቲ የመምህራን መገምገሚያ መስፈርት (በተማሪዎች የሚሞላ)**

የተገምጋሚው መምህር ስም \_\_\_\_\_  
 የተገምጋሚው መ/ር ኮሌጅ-----  
 የተገምጋሚው መ/ር የትምህርት ክፍል-----  
 የኮረሱ ስም \_\_\_\_\_ ኮረስ ኮድ \_\_\_\_\_  
 የትምህርት ዘመን \_\_\_\_\_ ሴሚስትር \_\_\_\_\_

- መግለጫ:**
1. እያንዳንዱን የመገምገሚያ ነጥብ በጥንቃቄ አንብብ/ቢ።
  2. ከነበብከ/ሽው የመገምገሚያ ነጥብ አኳያ የመምህርህ/ሽን
    - 2.1 የማስተማር ዝግጅት እና ብቃት፣
    - 2.2 የትምህርት አቀራረብና
    - 2.3 አጠቃላይ ባህሪ በሚዘገቡት መዝገን
  3. ከነበብከው/ሽው የመገምገሚያ ነጥብ አኳያ የመምህሩ/ሯን የሥራ እፈጻጸም **ከበጣም ዝቅተኛ** እስከ **በጣም ከፍተኛ** በቁጥር ተመንዝሮ ከተቀመጠው ውስጥ አንዱን በማክበብ አመልክት። መምህሩን/ሯን የማይመለከት ነጥብ ካለ **'አይመለከተውም'** በሚል አመልክት/ኛ።

ተረቁ.	የመገምገሚያ መስፈርት	በጣም ዝቅተኛ	ዝቅተኛ	መካከኛ	ከፍተኛ	በጣም ከፍተኛ	አይመለከተውም
<b>Core Competency</b>							
1	የትምህርቱን ዝርዝር ይዘት /ኮርስአውቶላይን/ በግልፅ ያሰውቃል በጊዜ ሰሌዳ መጥፍ ያዘጋጃል፣የቀርባል፣ለተማሪው አባዝቶይሰጣል	1	2	3	4	5	
2	የትምህርቱ አጠቃላይ ዓላማ በግልጽ ለተማሪዎች ያሰውቃል፣	1	2	3	4	5	
3	የትምህርቱን ዝርዝር ዓላማ በጊዜው ለተማሪዎች ግልጽ ያደርጋል /ለምሳሌ በየምዕራፍ በየሮዕሱ.....ውዘተ፣	1	2	3	4	5	
4	በመጀመሪያ ቀን የመጀመሪያ ክለስ ያስተምራል፣ <b>ይመዝናል።</b>	1	2	3	4	5	
5	ስለሚያስተምረው ትምህርት ተገቢ ዝግጅት አድርጎ ያቀርባል፣	1	2	3	4	5	
6	በትምህርቱ ዝርዝር ይዘት መሠረት ትምህርቱን ያቀርባል፣	1	2	3	4	5	
7	ለሚያስተምረው ትምህርት አስፈላጊ ጽሁፎችን አዘጋጅቶ ይሰጣል።	1	2	3	4	5	
8	በቤተ-መጻሕፍት ውስጥ ያሉ ለሚያስተምረው ትምህርት ጠቃሚ የሆኑትን ማጣቀሻ መፃሕፍት ዝርዝር (Reference) አዘጋጅቶ ይሰጣል፣	1	2	3	4	5	
9	እንደየአስፈላጊነቱ የተግባር ልምምድ ይሰጣል(ምርክሻ፣ ለበራቶሪ፣ የመስክጉብኝት፣case study ወዘተ..)።	1	2	3	4	5	
<b>Professional Competency</b>							
10	የማስተማሪያ ቋንቋን በመጠቀም ትምህርቱን በግልፅ ያቀርባል፣	1	2	3	4	5	

11	የተለያዩ የትምህርት መርጃ መሣሪያዎችን በመጠቀም ያስተምራል	1	2	3	4	5	
12	በክፍል ውስጥ የተማሪዎችን ጥያቄ ተቀብሎ በአግባቡ ያስተናግዳል፤	1	2	3	4	5	
13	ለትምህርቱ ጠቃሚ የሆኑ መልመጃዎችን ይሰጣል ያርማል/Class work, Homework, queez,...etc፤	1	2	3	4	5	
14	በ1ለ5 አደረጃጀት ወይም የቡድን ስራ በክፍል ውስጥ በአሳታፊ የስነ-ምስተማር ዘዴ ተጠቅሞ ያስተምራል፤	1	2	3	4	5	
15	የተከተታታይ ምዘናን በትክክል ይተገብራል፡፡	1	2	3	4	5	
16	የምዘና ዉጤት መሰረት በማድረግ በዉጤታቸው ዝቅ ላሉ ተማሪዎች ቲቶሪያል ይሰጣል፡፡	1	2	3	4	5	
17	የሚጠቀሙባቸውን የምዘና ዘዴዎች ከነዋጋቸው በቅድሚያ ለተማሪዎች ይገልጻል	1	2	3	4	5	
18	የፈተና ጥያቄዎችን ካስተማረው ትምህርት ጋር አዛምዶ ያወጣል፤	1	2	3	4	5	
19	በሚሰጣቸው ፈተናዎች ለሚጠቃለሉ ጥያቄዎች ሚዘናዊ የሆነ ዋጋ ይሰጣል፤	1	2	3	4	5	
20	ለሚያወጣቸው ፈተናዎች የተመጣጠነ ጊዜ ይመድባል፤	1	2	3	4	5	
21	በርዕሶች ስፋት መሰረት የጥያቄዎችን ቁጥር ያመጣጥናል	1	2	3	4	5	
22	የተከተታታይ ምዘና ዉጤት ግብረ-መልስ በአጭር ጊዜ ውስጥ ይሰጣል	1	2	3	4	5	
23	ፈተና ሲያወጣ የተለያዩ የምዘና አይነቶች ያካትታል፤ለምሳሌ ምርጫ፣አዛምድ፣ዳሽ ሙሊ፣ግለጽ፣ወዘተ	1	2	3	4	5	
24	ተከታታይ ምዘና ዉጤት መሰረት በትምህርታቸው ዝቅተኛ ውጤት ለመጡ ተማሪዎች ድጋሚ ፈተና ይሰጣል፤	1	2	3	4	5	
25	ለሴትና በትምህርታቸው ድጋፍ ለሚሹ ተማሪዎች ቲቶሪያል ይሠጣል፤ ይደግፋል፡፡	1	2	3	4	5	
<b>Ethical Competence</b>							
26	ለተማሪዎች ተገቢውን ክብር ይሰጣል፤	1	2	3	4	5	
27	በክፍል ውስጥ ተማሪዎች ስለትምህርቱ ሀሳባቸውንና አስተያየታቸውን በነጻነት እንዲገልጹ ይፈቅዳል፤	1	2	3	4	5	
28	የተማሪዎችን ችግር ለማዳመጥ ፈቃደኛ ነው፤	1	2	3	4	5	
29	ስነ-ስርዓት አክባሪነቱ፣ አለባባሱ የሚያሳየው ጭዋነትና እዉቀቱን ለማስተላለፍ የሚያሳየው ቅንነትና ተነሳሽነት	1	2	3	4	5	
30	በዘር ፣በሀይማኖትና በፆታ ልዩነት አይፈጥርም	1	2	3	4	5	
31	ጾታዊ ትንኮሳን ይጸደባል	1	2	3	4	5	
<b>Time Management</b>							
32	በተመደበለት የትምህርት ክፍለ-ጊዜ ሰዓት አክብሮ ይገኛል፤ሰዓቱን በሚገባ ተጠቅሞ ይወጣል	1	2	3	4	5	
33	ክፍለ-ጊዜውን በአግባቡ ለማስተማር ተግባር ያውላል፤	1	2	3	4	5	
34	በተግባር ትምህርት /laboratory, workshop and internship/ ክፍለ-ጊዜ ሙሉ ጊዜውን በአግባቡ ተጠቅሞ ያስተምራል፡፡	1	2	3	4	5	
35	ተማሪዎችን ለመርዳት የማማከሪያ ሰዓት መድቦ ያሳውቃል፤	1	2	3	4	5	
36	ተማሪዎችን ለመርዳት በመደበው ሰዓት ተማሪዎች ለሚያቀርቡት የአካዳሚክ ችግሮች መፍትሄ ይፈልጋል፤	1	2	3	4	5	
<b>አጠቃላይ ዉጤት ከመቶ</b>							

Figure 2. 1 Instructor Evaluation Form (By Student)

**WOLKITE UNIVERSITY**  
**INSTRUCTORS PERFORMANCE EVALUATION QUESTIONNAIRE**  
*(TO BE COMPLETED BY THE DEPARTMENT CHAIRPERSON)*

Name of Instructor \_\_\_\_\_ College \_\_\_\_\_ Department \_\_\_\_\_

Academic Year 20.../20.. \_\_\_\_\_ Semester \_\_\_\_\_ Date \_\_\_\_\_

Instruction I: Indicate the performance of the instructor by circling one of the options against each statement.

1= Very Low (VL); 2= Low (L); 3= Average (A); 4= High (H); 5= Very High (VH) NA "Not Applicable"

No	A. Evaluation Criteria	Scale					
		VL	L	A	H	VH	NA
<b>Core Competency</b>							
1	Efforts of self development in his/her specialization	1	2	3	4	5	NA
2	Adequacy of subject matter knowledge	1	2	3	4	5	NA
3	Willingness to accept additional teaching assignments when compelling situation arises in the department	1	2	3	4	5	NA
4	Willingness to accept related assignments other than regular teaching in the department (mentorship...)	1	2	3	4	5	NA
5	Effectiveness as a mentor in educational development army, internship etc..	1	2	3	4	5	NA
6	Active participation in improvement of teaching-learning process	1	2	3	4	5	NA
7	Participation in community service affairs	1	2	3	4	5	NA
8	Participating actively in seminars/workshop/symposia	1	2	3	4	5	NA
9	Identifying priority areas in one's discipline and pursuing research in that area	1	2	3	4	5	NA
10	Participation in research project and project proposal development	1	2	3	4	5	NA
11	Performance as an academic advisor	1	2	3	4	5	NA
<b>Professional Competence</b>							
12	Participation in problem identification and solving at department/college/institution	1	2	3	4	5	NA
13	Continuous assessment implementation	1	2	3	4	5	NA
14	Participating actively in departmental/Faculty/institution affairs	1	2	3	4	5	NA
15	Providing and reporting tutorial activities designed for the students	1	2	3	4	5	NA
16	Participation in preparation and reviewing of teaching materials	1	2	3	4	5	NA
17	Updating teaching materials	1	2	3	4	5	NA
18	Willingness to share University resources with other colleagues						
<b>Time Management</b>							
19	Executing assigned classes/invigilation on time.	1	2	3	4	5	NA
20	Notifying and implementing consultation timely	1	2	3	4	5	NA
21	Giving timely feedback to students	1	2	3	4	5	NA
22	Meeting deadlines (in reporting, SIMS result feeding, submission of grade/ documents..etc...)	1	2	3	4	5	NA
<b>Ethical Competence</b>							
23	Having positive attitude to work with others	1	2	3	4	5	NA
24	Showing concern for the use of resources of the department and the University	1	2	3	4	5	NA
25	Willingness and participation in committee works at department /University level	1	2	3	4	5	NA
26	Willingness to take assignments outside the University in his area of specialization	1	2	3	4	5	NA
27	His/her professional ethics (dressing, hair style, personality...)	1	2	3	4	5	NA

**General comments about the instructor**

Strength of the instructor?

\_\_\_\_\_

\_\_\_\_\_

Suggested points/aspects the instructor should improve.

\_\_\_\_\_

*Figure 2. 2 Instructor Evaluation Form (By Head)*

**WOLKITE UNIVERSITY**

**INSTRUCTORS' PERFORMANCE EVALUATION FORM**

**(To be completed by Colleagues)**

Name of Instructor/Colleague to be evaluated -----college----- Department -----

Academic year ----- Semester----- Date-----

Instruction 1: Indicate the performance of your colleague by circling one of the options against each statement.

1= very low (VL) 2=Low(L) 3= Average(A) 4= High(H); 5= Very High (VH) 'Not Applicable'

No	Evaluation Criteria	Scale					
		VL	L	A	H	VH	NA
<b>Core Competency: Subject matter</b>							
1	Contribution in preparing and searching for teaching materials	1	2	3	4	5	NA
2	Continuous update of the subject matter	1	2	3	4	5	NA
3	Attending seminars organized by the department/faculty/ institution	1	2	3	4	5	NA
4	Level of his/her subject matter knowledge and skill	1	2	3	4	5	NA
<b>Core Competency: Research and Community Services</b>							
5	Willingness and level of engagement in community service activities	1	2	3	4	5	NA
6	Participation on seminars/workshops at department/ faculty/ institution level during the year	1	2	3	4	5	NA
7	Identifying priority areas in one's discipline and pursuing research in that area	1	2	3	4	5	NA
8	Willingness to prepare research projects with other colleagues	1	2	3	4	5	NA
9	Willingness to help colleagues in identifying areas of research and proposal development	1	2	3	4	5	NA
<b>Professional Competency</b>							
10	Guidance and counseling role to students	1	2	3	4	5	NA
11	Contributing ideas and activities that improve the teaching learning process.	1	2	3	4	5	NA
12	Participation in problem identification and solving at department/college/institution	1	2	3	4	5	NA
13	Willingness and preparedness to implement change tools	1	2	3	4	5	NA
14	Willing to actively participate in 1-5 grouping(Education development army) activities	1	2	3	4	5	NA
15	Implementation of different teaching methods in his/her discipline.	1	2	3	4	5	NA
<b>Ethical Competence</b>							
16	Willingness to participate and level of commitment in committee works	1	2	3	4	5	NA
17	Participating actively in departmental/faculty/ institute meetings	1	2	3	4	5	NA
18	Willingness to share university resources with other colleagues	1	2	3	4	5	NA
19	Showing cordiality to others	1	2	3	4	5	NA
20	Having positive attitude to work with others	1	2	3	4	5	NA
21	Respecting ideas of others	1	2	3	4	5	NA
22	Level of respect to rules and guidelines of the institution	1	2	3	4	5	NA
23	His/her discipline (dressing, addictions, personality etc...)	1	2	3	4	5	NA
<b>Time Management</b>							
24	Time management in department Affairs	1	2	3	4	5	NA
25	Time utilization for consultation hours	1	2	3	4	5	NA

Overall assessment of the colleague

Strengths of the Instructor?

-----  
 -----  
 -----

Suggested points/aspects the instructor should improve.

-----  
 -----  
 -----

*Figure 2. 3 Instructor Evaluation Form*





**የወልደሙ የኢብሊት የሰነድ ስራተኛ የ2016 ዓ.ም የስራ ስራዎች ምላሽ ቅጽ -1**

የስራ ተክል ስም .....

የስራ ክፍል .....

የስራ ደረጃ .....

የገምገማ ጭነት የሚለካው ጊዜ ነፃ ..... ዓ.ም ስለስነ ..... ዓ.ም

ተ.ቁ	የመመዘኛ ነገር	ስመዘኛ የተሰጠው ከስድስ ክፍያ	ስራተኛው ያገኘው ነገር
1	ገበያ ስነ ስራዎች ስራዎች ማዘጋጀት	7	
2	በስራ ስራዎች ስራዎች ስራዎች ስራዎች	8	
3	የተገቢው የስራ ስራዎች ስራዎች	5	
4	በተገቢው የስራ ስራዎች ስራዎች	10	
5	በስራ ስራዎች ስራዎች	5	
6	የስራ ስራዎች ስራዎች	5	
7	የስራ ስራዎች ስራዎች	5	
8	የስራ ስራዎች ስራዎች	5	
9	የስራ ስራዎች ስራዎች	5	
10	የስራ ስራዎች ስራዎች	10	

ገምገማውን የሚሰጡ ሰነድ ስራተኛው ስም .....

የስራ ተክል ስም .....

ሰነድ ስራተኛው ስም .....

ፊርማ .....

ቀን .....

በገምገማው የሚገኝ ያዘጋጀ ስም .....

ሰነድ ስራተኛው ስም .....

ፊርማ .....

ቀን .....

Figure 2. 5 Administrative Employee Evaluations form (By minute)

## 2.5. Drawbacks of the Existing System

The major drawback of the existing manual systems is

- ❖ Lacks flexibility and is resistant to change.
- ❖ Physical copies are kept for personal records.
- ❖ The system does not span across all the colleges within the university.
- ❖ Stops abruptly in the middle of the organizational structure
- ❖ Tailored for academic employees, the system does not take into account the performance evaluation needs of administrative staff.

## **2.6. Business rules of the existing system**

A business rule is an organization running principle or guide for practitioners of a system. These are always used for a specific system to be controlled over the policies of the whole organization. University has rules and regulations which the students, and officers should follow to properly clear customers.

**The main business rules or principles of the existing system are: -**

**BR1:** Individuals undergoing a performance appraisal must be current employees of the organization.

**BR2:** Employees should obtain performance appraisal forms from their respective department heads or officers.

**BR3:** Department heads and officers should access performance appraisal forms from the Human Resource (HR) department.

**BR4:** Employees seeking a performance appraisal must fulfill their job responsibilities and meet any outstanding requirements.

**BR5:** Officers are responsible for assessing employees, addressing any concerns, and verifying the completion of responsibilities.

**BR6:** Heads have the authority to approve performance appraisal submissions.

**BR7:** Head must submit the approved performance appraisal to the HR department. A copy is retained for their personal records.

**BR8:** Employees submit completed performance appraisal forms by the specified deadline.

**BR9:** All performance appraisal data should be treated as confidential and shared only with individuals directly involved in the appraisal process.

**BR10:** All performance appraisal data should be treated as confidential and shared only with individuals directly involved in the appraisal process.

**BR11:** Employees must complete peer evaluation before initiating their self-evaluation in the performance appraisal process.

## **CHAPTER THREE**

### **PROPOSED SYSTEM**

The proposed performance appraisal system is a significant expansion of the existing system, both in terms of functionality and scope. It includes a few new features and capabilities that are not available in the existing system, such as

- ❖ **Complaint feature:** Employees can now submit complaints about their performance reviews or other aspects of the performance appraisal process through the system. This feature helps to ensure that employees have a voice in the process and that their concerns are heard.
- ❖ **Inclusion of administrative employees:** The proposed system now includes administrative employees, who were previously excluded from the performance appraisal process. This ensures that all employees, regardless of their role, are evaluated and have opportunities for development.
- ❖ **Expansion from HR to employees:** The proposed system is now accessible to both HR professionals and employees. This allows employees to take greater ownership of their performance and development goals.

#### **3.1 Functional Requirements**

- ❖ Users must be able to evaluate their assigned staff members.
- ❖ The system must provide a streamlined and efficient process for department heads to evaluate instructors.
- ❖ The system must provide a streamlined and efficient process for students to review instructors.
- ❖ The system must enable administrative employees to evaluate their performance.
- ❖ The system shall automatically generate final employee appraisal results based on predefined rules.
- ❖ The system must maintain a historical record of each employee's appraisal results.

- ❖ This record shall be accessible to authorized users for reference and tracking purposes.
- ❖ Department heads must have the ability to manage instructors within their department.
- ❖ Team leaders must have the ability to manage administrative employees within their department.
- ❖ The system should be capable of generating comprehensive reports on employee appraisals.
- ❖ The system must allow employees to submit complaints regarding their appraisal results.
- ❖ The system must provide a streamlined and efficient process for college deans to review the Head of Department.
- ❖ The system must provide a streamlined and efficient process for the Director to review team leader performance.
- ❖ The system must allow the Director to approve final appraisal results of all team leaders within the department.
- ❖ The system must allow the Dean to approve final appraisal results of all department heads within the department.
- ❖ The department head and team leader must approve the review of their department employees.
- ❖ The system shall enable department heads to assign courses to instructors
- ❖ The system shall enable HR to easily set dynamic appraisal criteria and rules.
- ❖ The system must allow the system admin to assign roles to users that correspond to their designated positions.

## **3.4. Non-functional Requirements**

### **3.4.1 User Interface and Human Factors**

The system interface is designed to accommodate diverse user roles and capabilities within the performance appraisal system. Its goal is to offer a user-friendly experience that simplifies navigation and access to sensitive data, such as employee records and assessment outcomes.

Key steps Followed to improve the interface

- ❖ Simplify menus and navigation paths to ensure easy access to essential functions and features.
- ❖ Utilize clear icons, labels, and visual cues for intuitive guidance and ease of use.
- ❖ Maintain uniformity in colors, fonts, and layout throughout the interface for a cohesive user experience
- ❖ Ensure the interface adapts seamlessly to various devices and screen sizes for accessibility.
- ❖ Streamline task flows to reduce user effort and enhance the system's usability.

### **3.4.2 Performance Consideration**

The performance of the evaluation system is essential due to its extensive user base. Optimizing system performance is crucial to accommodate numerous users efficiently, ensuring rapid access to critical data and maintaining seamless interactions. This optimization is imperative to enhance productivity and meet the demands of a large user pool.

To optimize system performance, those measures will be taken

#### **Backend Optimization**

- ❖ Implement indexing in MongoDB to optimize queries.
- ❖ Employ database denormalization for faster data retrieval.
- ❖ Optimize Node.js and Express APIs for efficient data processing.
- ❖ Utilize connection pooling and timeouts for streamlined database access.

#### **Frontend Techniques:**

- ❖ Utilize lazy loading in React components for improved initial page load.
- ❖ Implement client-side rendering optimization to enhance frontend performance.
- ❖ Employ predictive prefetching and resource preloading for quicker data access.

### **Database and Query Optimization**

- ❖ Optimize database queries, including using indexes and efficient data retrieval strategies.

### **Media and Asset Optimization:**

- ❖ Optimize images and media assets for web consumption to reduce loading times.
- ❖ Utilize responsive image techniques and compression to minimize file sizes.

### **3.4.3 Security Issues**

The performance appraisal system is a complex ecosystem involving various stakeholders, each entrusted with distinct roles, capabilities, and access to sensitive data encompassing employee credentials and assessment results. Safeguarding this diverse and confidential information requires a multifaceted security approach, addressing both internal and external threats, as well as potential risks posed by authorized users.

To strengthen the system's security, several protective measures will be adopted:

- ❖ Implement robust encryption mechanisms, utilizing industry-standard algorithms, to safeguard sensitive data stored in MongoDB against unauthorized access, ensuring confidentiality and integrity.
- ❖ Utilize a secure token-based authentication system that generates and validates unique tokens to authenticate and authorize users, enhancing security by requiring these tokens in addition to traditional credentials like usernames and passwords for user access
- ❖ Implement role-based access control (RBAC) with granular permissions, carefully managing user privileges based on their roles and responsibilities to mitigate unauthorized data access or modification.
- ❖ Enforce strict validation and sanitization of user inputs across the system, preventing injection attacks and preserving data integrity against potential vulnerabilities

### **3.4.4 Error Handling and Validation**

Error handling in our system is crucial to safeguard the accuracy and integrity of assessment records, preventing any compromise due to incorrect or improperly formatted data entries during the appraisal process.

To handle exceptions several protective measures will be adopted:

- ❖ Validate user inputs to prevent incorrect data entry. Display clear messages prompting users to correct their inputs for accurate assessments.
- ❖ Implement secure authentication mechanisms to manage and notify users about failed login attempts without disclosing sensitive information
- ❖ Employ connection pooling and retry mechanisms to manage database connectivity problems, providing informative error messages when queries fail

### **3.4.5 Quality Issues**

In the appraisal system, reliability stands as a priority, ensuring the accurate presentation of exclusively requested data. Users can confidently trust the system's accuracy, fostering unwavering confidence in the information received.

Furthermore, robustness defines the system's core. It actively identifies and guides users through errors, promoting corrective actions. This proactive approach enhances system resilience, ensuring a smooth and fault-tolerant appraisal experience

Strategy used to achieve those quality attribute

#### **Reliability**

- ❖ Implement stringent data retrieval mechanisms tailored to user requests.
- ❖ Conduct regular data integrity checks and validation procedures.
- ❖ Utilize caching strategies to optimize data retrieval and minimize errors.

#### **Robustness**

- ❖ Develop comprehensive error detection and handling protocols.
- ❖ Provide intuitive and clear error messages for users.
- ❖ Incorporate fail-safes and recovery mechanisms to mitigate system errors.

### **3.4.6 Resource Issues**

The ReactJS library operates as a single-page application, inherently minimizing resource consumption. Its design allows for the initial loading of content and subsequent updates without the need for complete page refreshes. This efficient approach reduces the strain on resources, notably in terms of network requests and server processing, resulting in a more fluid user experience while optimizing resource

### **3.4.7. Physical Environment**

The system's backend resides on Render, providing robust infrastructure for secure data management.

Meanwhile, Vercel hosts the frontend, delivering a fast and reliable interface

### **3.4.8. Backup and Recovery**

Backups are crucial for an appraisal system to ensure the safety and integrity of sensitive data, including employee records and assessment results. They act as a safety net, safeguarding against data loss due to system failures, human errors, or cyber threats, ensuring the continuity and reliability of the system

Those measures will be taken

- ❖ Utilizing tools like 'mongo dump' for point-in-time backups of the MongoDB database
- ❖ Utilize MongoDB's replication features for the creation of redundant copies of the database across multiple nodes

### **3.4.9. Documentation**

The system demands a comprehensive documentation hierarchy. This includes user-friendly guides for system navigation, technical documentation to support maintainers, and detailed records outlining the development process.

# CHAPTER FOUR

## SYSTEM ANALYSIS

This chapter aims to present the system to stakeholders in a manner that ensures clear and accessible understanding. Requirements analysis delves deeper into fundamental requirements established during the elicitation process. It represents the initial technical depiction of the PAS System through a variety of requirements modeling approaches.

Utilizing a spectrum of techniques

- ❖ **System Model:** Use case diagrams construct vivid scenarios showcasing how PAS operates in real-life situations.
- ❖ **Dynamic Model:** Sequence diagrams, state chart diagrams, and activity diagrams paint a dynamic picture of PAS's functioning and behavior.
- ❖ **Object Model:** Class diagrams and data dictionaries create a structural framework defining key elements and their relationships within PAS.

Each modeling method contributes uniquely, allowing stakeholders to visualize and comprehend PAS from multiple angles. These approaches bridge the gap between intricate technicalities and a comprehensive understanding for all stakeholders involved in PAS's development and use. This marks the foundational step towards crafting a well-defined and user-friendly Performance Appraisal System.

### 4.1 System Model

This section of the Chapter elaborates on the modeling phase of the proposed system, employing object-oriented methodology and tools. It encompasses a comprehensive analysis of all activities executed by the system's actors and the various processes involved. This analysis primarily revolves around the utilization of use case modeling, which serves as a pivotal framework for depicting interactions, functionalities, and system behaviors.

### **4.1.1 Use Case Model**

In PAS the following use cases are available

- ❖ Logout
- ❖ Login
- ❖ Request evaluation
- ❖ View Result
- ❖ Review self
- ❖ Review Peers
- ❖ Review Instructors
- ❖ Review Subordinate
- ❖ Approve evaluation Result
- ❖ Manage Users
- ❖ Assign Role
- ❖ Define Criteria
- ❖ Access Employee detail
- ❖ Assign Course
- ❖ Manage Department

### **Actor Identification and Description**

In PAS there are the following actors

#### **HR**

This actor is responsible for managing performance reviews, whose responsibilities on the system include:

- ❖ Define Criteria
- ❖ Access Employee detail
- ❖ Request evaluation
- ❖ Define Rule

#### **Admin**

This actor is the administrator of the system who is in charge of the system and user accounts.

The responsibilities or use cases of System admin include:

- ❖ Creating user account
- ❖ Modifying user account
- ❖ Deleting user account
- ❖ Displaying user accounts.
- ❖ Searching for user account
- ❖ Assign Role
- ❖ Manage Department

### **Employee**

This actor is the user of the system who is participating in the evaluation process. Employee can be academic employee or Administrative employee

The responsibilities or use cases of Employee include:

- ❖ View Result
- ❖ Review Peers
- ❖ Review self (Administrative)

### **Student**

This actor is the user of the system who oversees evaluating Instructors who currently teach the course.

The responsibilities or use cases of Student include:

- ❖ Review Instructors

### **Team Leader**

Team Leader serves as a system user responsible for assessing and overseeing the performance of administrative Employees within the Department. Their responsibilities encompass

- ❖ Reviewing subordinates' performance
- ❖ Approving evaluation results to ensure effective management

## **Head**

Department Head serves as a system user responsible for assessing and overseeing the performance of Academic Employees within the Department. Their responsibilities encompass

- ❖ Reviewing Instructors and other employee evaluation
- ❖ Approving evaluation results to ensure effective management
- ❖ Assign Course

## **Dean**

Dean of College serves as a system user responsible for assessing and overseeing the performance of Heads within the College.

Their responsibilities encompass

- ❖ Reviewing Team Leader performance
- ❖ Approve evaluation Result of Team Leader

### 4.1.1.1 Use Case Diagrams

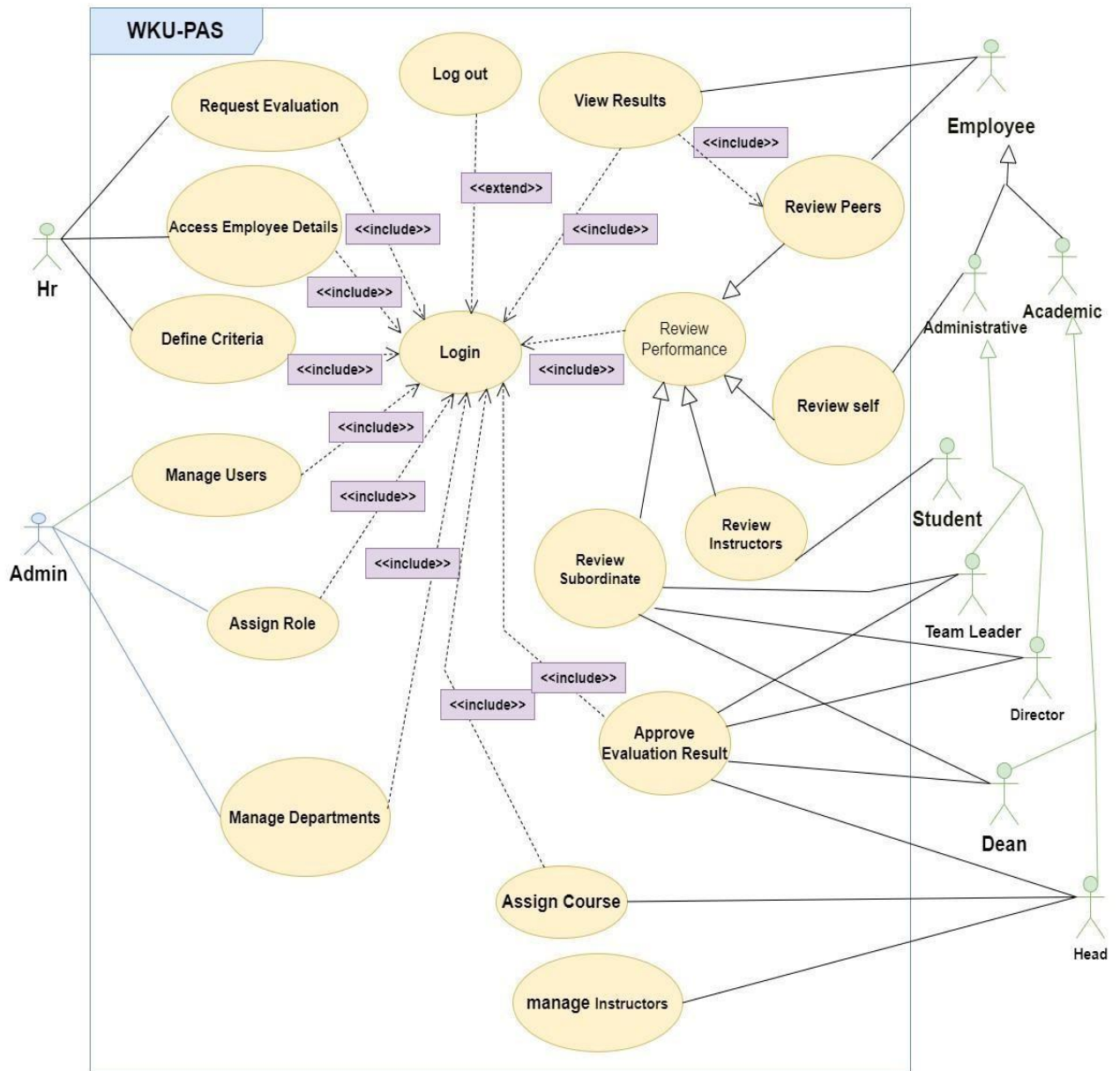


Figure 4. 1 Use Case Diagram

#### 4.1.1.2 Use Case Description

Table 4. 1 Define evaluation criteria

<b>Use case name</b>	<b>Define Evaluation Criteria</b>
Use Case Identifier	UC001
Participating actor	HR Personnel
Description	This use case describes the process by which HR personnel prepare evaluation rules and criteria for employee assessments.
Pre-conditions	The HR personnel have the appropriate permissions to define evaluation rules and criteria.  The HR personnel logs into the system.
Flow of event	The system displays a form or interface for defining evaluation criteria.  The HR personnel enters criteria details, such as the name, description, and weightage for each criterion.  The HR personnel confirms the addition of the new evaluation criteria.  The system updates the criteria list, and the HR personnel receives a confirmation message.
Post condition	New evaluation criteria and rules are defined and added to the system.  Existing criteria and rules are reviewed and adjusted as necessary.

Table 4. 2 Request Evaluation

<b>Use case name</b>	<b>Request Evaluation</b>
Use Case Identifier	UC002
Participating actor	HR Personnel

Pre-conditions	<p>HR personnel have the appropriate permissions to initiate the evaluation process.</p> <p>Departments and employees are pre-defined in the system.</p> <p>The HR personnel logs into the system.</p>
Flow of event	<p>The system presents a dashboard or menu with options for managing the evaluation process.</p> <p>The HR personnel selects the "Set Evaluation Period" option.</p> <p>The system displays a form or interface for setting the start and end dates for the evaluation period.</p> <p>The HR personnel enters the start and end dates for the evaluation period.</p> <p>The HR personnel confirms the evaluation period, and the system updates the information.</p> <p>The system generates notifications or messages to inform all department heads about the upcoming evaluation period.</p> <p>The notifications include details about the start and end dates of the evaluation period.</p> <p>Department heads receive notifications through email.</p>
Post condition	<p>The evaluation period is set in the system.</p> <p>All departments and employees are informed about the evaluation process.</p> <p>Employees submit their evaluations during the specified period.</p>

*Table 4. 3 Evaluate Self*

<b>Use case name</b>	<b>Self-evaluation</b>
Use Case Identifier	UC003

Participating actor	Administrative employee
Description	This use case help admin employee to give result about one's performance
Entry condition	The employee is logged into the employee appraisal system.
Flow of event	<p>The employee clicks on the "Self-evaluation" button.</p> <p>The employee is presented with a form to evaluate their performance.</p> <p>The employee completes the form and submits it.</p> <p>The employee's self-evaluation is stored in the system.</p>
Alternative flow of event	If the HR department does not initiate the evaluation process, the employee will need to wait until they do
Post condition	The employee's self-evaluation is available for review by their Team Leader.

*Table 4. 4 Evaluate Peers*

<b>Use case name</b>	<b>Evaluate Peers</b>
Use Case Identifier	UC004
Participating actor	Employee, peer
Description	This use case help employee to assess his staff members
Pre-conditions	<p>The employee is logged into the employee appraisal system.</p> <p>The employee has been assigned to evaluate their peer.</p>

Flow of event	<p>The employee clicks on the "Peer evaluation" button.</p> <p>The employee is presented with a list of their peers.</p> <p>The employee selects a peer to evaluate.</p> <p>The employee is presented with a form to evaluate their peer's performance.</p> <p>The employee completes the form and submits it.</p> <p>The employee's peer evaluation is stored in the system.</p>
Alternative flow of event	<p>If the HR department does not initiate the evaluation process, the employee will need to wait until they do</p>
Post condition	<p>The employee's peer evaluation is available for review by their Dean or Team Leader</p>

*Table 4. 5 Evaluate Subordinate Employee*

<b>Use case name</b>	<b>Evaluate Subordinate Employee</b>
Use Case Identifier	UC005
Participating actor	Team Leader, Head, Dean, Director
Description	This use case describes the process by which various organizational roles can evaluate their respective subordinates.
Pre-conditions	<p>The actors (team leader, head, dean, director) have appropriate permissions to access and evaluate their subordinates</p> <p>The system has a list of employees with their roles, and the hierarchy of the organizational structure is defined.</p> <p>The actors is logged into the system</p>
Alternative flow of event	<p>If the actor does not have the appropriate permissions, the system displays an error message, and the evaluation process is terminated.</p>

Post condition	The evaluations are stored in the system and can be accessed for performance reviews and decision-making processes.
----------------	---

*Table 4. 6 Evaluate Instructors*

Use case name	Evaluate Instructors
Use Case Identifier	UC006
Participating actor	Student
Description	This use case describes the process by which students can provide feedback and evaluations for their instructors.
Pre-conditions	The student has access to the system. The system has a list of instructors associated with courses the student is enrolled in. The student logs into the system.
Flow of event	The system presents a list of courses the student is currently enrolled in. The student selects a course for which they want to evaluate the instructor. The system displays an evaluation form or interface specific to the instructor of the selected course. The student provides ratings, comments, and feedback for the instructor's performance. The evaluation is submitted to the system.
Alternative flow of event	If the student has not enrolled in any courses, the system notifies the student that there are no courses available for evaluation.

Table 4. 7 Approve the Evaluation Result

<b>Use case name</b>	<b>Approve the Evaluation Result</b>
Use Case Identifier	UC007
Participating actor	Head ,Team Leader
Description	This use case describes the process by which heads of departments And Team Leader can review and approve the evaluation results submitted by Employees within department
Pre-conditions	Evaluation results are submitted by Employees The actors (Head of Department, Team Leader) have appropriate permissions to access and approve evaluation results. The Head of Department or Team Leader logs into the system.
Flow of event	The system presents a list of evaluation results submitted by Employee within departments The Head of Department selects an evaluation result to review. The system displays the details of the evaluation, including ratings and comments. The Head of Department decides whether to approve or reject the evaluation. If approved, the evaluation status is updated, and the system notifies the relevant parties. If rejected, the system prompts the Head of Department to provide comments explaining the rejection.
Alternative flow of event	Need to wait until the department staff members Finish the evaluation process

Post condition	Approved evaluation results are marked as finalized and can be used for further decision-making processes.
----------------	--

*Table 4. 8 Assign Course*

<b>Use case name</b>	<b>Assign Course</b>
Use Case Identifier	UC008
Participating actor	Head
Description	This use case describes the process by which a department head assigns a course to an instructor.
Pre-conditions	Instructors are available and eligible to be assigned courses. The system has a list of courses that need to be assigned. The Department Head logs into the system.
Flow of event	The system presents a list of courses that need to be assigned or are available for assignment.  The Department Head selects a course to be assigned.  The system displays a list of eligible instructors  The Department Head selects an instructor to assign to the selected course.  The system updates the course assignment, associating the selected instructor with the chosen course.
Post condition	The course assignment is updated in the system.

*Table 4. 9 View Appraisal Result*

<b>Use case name</b>	<b>View Appraisal Result</b>
Use Case Identifier	UC009

Participating actor	Employee
Description	This use case describes the process by which employees can view their appraisal results after the evaluation criteria have been completed and the evaluation is finalized.
Pre-conditions	The employee has completed evaluating peers. The employee's evaluation has been finalized by the relevant authorities.
	3. The employee logs into the system.
Flow of event	The employee navigates to the "Appraisal" or "Performance" section of the system. The system presents the finalized appraisal results The employee reviews the appraisal results.
Alternative flow of event	If the employee has not completed Evaluating others or if the evaluation has not been finalized, the system displays a message indicating that the appraisal results are not yet available.
Post condition	The employee has viewed the finalized appraisal results. Can complain about the result

*Table 4. 10 Access Employee Information*

<b>Use case name</b>	<b>Access Employee Information</b>
Use Case Identifier	UC0010
Participating actor	HR Personnel
Description	This use case describes the process by which HR personnel can view and edit employee information, including employment history and evaluation results.

Pre-conditions	<p>The HR personnel have the appropriate permissions to access and edit employee information.</p> <p>Employee information, including history and evaluation results, is stored in the system.</p> <p>The HR personnel logs into the system.</p>
Flow of event	<p>The system presents a dashboard or menu with options for HR-related tasks.</p> <p>The HR personnel selects the "Employee Information" or a similar option.</p> <p>The system displays a list of employees or a search interface.</p> <p>The HR personnel searches for and selects a specific employee.</p> <p>The system presents detailed information about the selected employee, including personal details, employment history, and any available evaluation results.</p>
Alternative flow of event	<p>If the HR personnel does not have the appropriate permissions, the system displays an error message, and the viewing/editing process is terminated.</p>
Post condition	<p>The employee information is updated in the system.</p> <p>The HR personnel has successfully viewed and edited the employee information.</p>

*Table 4. 11 Manage User*

Use case name	Manage User
Use Case Identifier	UC0011
Participating actor	Admin
Description	This use case describes the process by which an admin manages system users, including assigning roles, adding, editing, and deleting user accounts.
Pre-conditions	<p>The admin has the appropriate permissions to manage user accounts.</p> <p>User accounts and roles are defined in the system. ▪</p>

	The admin is logged into the system
Flow of event	<p>The system presents a dashboard or menu with options for user management.</p> <p>The admin selects the "Assign Role" option.</p> <p>The system displays a list of existing users.</p> <p>The admin selects a user to assign a role to.</p> <p>The system presents a list of available roles.</p> <p>The admin selects a role from the list and confirms the assignment.</p> <p>The system updates the user's role, and the admin receives a confirmation message.</p>
Post condition	User roles are assigned, new users are added, user information is edited, and users are deleted based on the admin's actions.

*Table 4. 12 Manage Instructor*

Use case name	Manage Instructor
Use Case Identifier	UC0012
Participating actor	Head
Description	This use case describes the process by which a department head manages instructors by adding new instructors and editing existing instructor information.
	<p>The Department Head has the appropriate permissions to manage user accounts.</p> <p>The Head is logged into the system</p>

Flow of event	<p>The system displays a form or interface for adding a new instructor.</p> <p>The department head enters the necessary details for the new instructor, such as name, contact information, and qualifications.</p> <p>The department head confirms the addition of the new instructor.</p> <p>The system updates the list of instructors</p>
Post condition	New instructor are added, instructor information is edited, and instructor are deleted

*Table 4. 13 Manage Departments*

<b>Use case name</b>	<b>Manage Departments</b>
Use Case Identifier	UC0013
Participating actor	Admin
Description	This use case help admin create, edit , delete and update college and department
Pre-conditions	The admin has the appropriate permissions to manage departments.

Flow of event	<p>The admin logs into the system.</p> <p>The system presents a dashboard or menu with options for managing departments.</p> <p>The admin selects the appropriate option for managing departments.</p> <p>The system provides a form or interface for modifying department details.</p> <p>The admin makes the necessary modifications</p> <p>The admin confirms the changes.</p> <p>The system updates department details, and the admin receives a confirmation message.</p>
Post condition	<p>New departments are added to the system.</p> <p>Heads are assigned departments, respectively.</p>

*Table 4. 14 Login use case description*

Use case name	Login
Use Case Identifier	UC0014
Participating actor	All system user
Description	Any user who wants to use the system's functionality must first log in and be authenticated and authorized.
Entry condition	The user must already be registered (the user must have username, password and account type).
Flow of event	<p>The user opens the system</p> <p>The user clicks the login link.</p> <p>The system displays the login page</p> <p>The user enters his or her personal information (Username and password)</p>

	<p>QThe user clicks the login button.</p> <p>The system verifies the username and password.</p> <p>The system takes the user to his/her page.</p>
Alternative flow of event	1. If the user inserted the wrong username and password 2. The system displays errors and tries again.
Post condition	Users log to the system.

*Table 4. 15 Logout*

<b>Use case name</b>	<b>Logout</b>
Use Case Identifier	UC0014
Participating actor	All system users
Description	Any user who wants to logout from the system clicks the logout button.
Entry condition	The user must be logged into the system.
Flow of event	<p>The system displays the logout button.</p> <p>The user clicks the logout button</p>
Post condition	The user leaves from the system

#### **4.1.1.3 Use case Scenario**

The system has many different scenarios that are divided into participating actors and user groups. In this specific case we identify the following scenarios

#### **Login scenario**

When Mr. Abraham wants to start the system, they enter their username, password and then enter into the system.

**Scenario Name: - Login Participating**

**Actor: - Mr. Abebe**

#### **Flow of Event**

1. The user opens the system
2. The user clicks the login link.
3. The system displays the login page
4. The user enters his or her personal information (Username and password)
5. The user clicks the login button.
6. The system verifies the username and password.
7. The system takes the user to his/her page.

#### **Request Evaluation Scenario**

Participating Actor: Sisay Demes (HR Personnel)

#### **Flow of Event**

1. Sisay Demes accesses the system using their credentials.
2. Upon successful login, the system presents a dashboard with various options.
3. Sisay Demes selects the "Manage Evaluations" option from the dashboard.
4. The system displays an interface showcasing ongoing and upcoming evaluation processes.
5. Sisay Demes clicks on the "New Evaluation" button.
6. The system prompts Sisay Demes to set the start and end dates for the evaluation period. Sisay Demes enters the desired start and end dates in the provided fields.
7. After confirmation, the system updates the evaluation period information.

8. The system automatically generates detailed notifications to all department heads regarding the upcoming evaluation.
9. Notifications contain comprehensive information about the start and end dates of the evaluation period.
10. Department heads receive these notifications within their respective system accounts.

### **Define Evaluation Criteria Scenario**

Scenario Name: Define Evaluation Criteria - Participating

Actor: Sisay Demes (HR Personnel)

#### **Flow of Event:**

1. Sisay Demes logs into the system using their credentials.
2. Upon successful login, the main dashboard or menu options are presented.
3. Sisay Demes selects the "Define Evaluation Criteria" option from the menu.
4. The system displays a dedicated form/interface for defining evaluation criteria.
5. Sisay Demes begins entering details for the criteria, including names, descriptions, and associated weightages for each criterion.
6. After inputting the necessary details, Sisay Demes confirms the addition of the new evaluation criteria.
7. The system processes the information and updates the criteria list.
8. Sisay Demes receives a confirmation message indicating the successful addition of the new evaluation criteria.

## **View Appraisal Result Scenario**

Use Case Name: View Appraisal Result

Participating Actor: Mr. Kebede (Employee)

### **Flow of Event**

1. Upon successful login, Mr. Kebede navigates to the "Result" section of the system.
2. The system presents the finalized appraisal results to Mr. Kebede.
3. Mr. Kebede reviews the provided appraisal results.
4. If Mr. Kebede has not completed evaluating others or if the evaluation has not been finalized
5. The system displays a message indicating that the appraisal results are not yet available.

## **Perform Evaluations scenario**

Participating Actor: Sarah (User)

### **Flow of Event:**

1. Sarah logs into the system using her credentials.
2. Upon successful login, the system presents different evaluation options based on Sarah's access level.
3. Sarah chooses the type of evaluation she wishes to conduct.
4. The system displays a list of entities available for evaluation (e.g., individuals, courses, projects).
5. Sarah selects the specific entity (e.g., person, course, project) to evaluate.
6. The system presents an evaluation form or interface tailored to the chosen entity. D. Sarah provides ratings, comments, and feedback relevant to the evaluation.
7. After completion, Sarah submits the evaluation through the system.

## **Approve the Evaluation Result scenario**

Use Case Name: Approve the Evaluation Result Participating

Actor: Abdi (Head, Team Leader)

### **Flow of Event:**

1. Abdi, as the Head of Department or Team Leader, logs into the system using his credentials.
2. Upon successful login, the system presents a list of evaluation results submitted by employees within the departments under Abdi's supervision.
3. Abdi selects an evaluation result from the presented list to review.
4. The system displays detailed evaluation information including ratings, comments, and relevant data provided by employees.
5. Abdi evaluates the details presented and decides whether to approve or reject the evaluation.
6. If approved: - The system updates the evaluation status to "approved." - Notifications are sent to relevant parties about the approved evaluation.

## 4.2. Object Model

Object modeling serves as a crucial technique in software engineering, enabling developers to effectively conceptualize, design, and communicate the structure and behavior of complex systems. At its core, object modeling revolves around the representation of a software system through objects and classes.

### Object modeling involves several key steps

#### Identifying Objects and Classes

- ❖ Objects represent individual entities in the system that possess both data (attributes) and behaviors (methods).
- ❖ Classes serve as blueprints defining the structure and behavior shared among a group of objects.

#### Defining Relationships:

- ❖ Objects and classes have connections or relationships with each other, such as associations, aggregations, or inheritances. These relationships illustrate how various elements collaborate within the system.

#### Detailing Attributes and Behaviors:

- ❖ Attributes encapsulate the data or properties within classes, representing the characteristics of objects.
- ❖ Behaviors, in the form of methods or functions, describe the actions or functionalities that objects can perform.

**Purpose and Outputs:** This section focuses on two key outputs to aid in understanding and representing the object model:

#### Class Diagram (Visual Representation):

- ❖ The class diagram provides a graphical depiction of the high-level relationships among classes/objects.
- ❖ It showcases the structure of classes, their attributes, methods, and connections, offering an abstract yet comprehensive overview.

### **Data Dictionary (Tabular Breakdown):**

- ❖ The data dictionary presents a detailed breakdown of attributes and constraints within identified entities or classes.
- ❖ It includes specifics like attribute names, data types, sizes, key constraints (e.g., primary keys), and any other relevant constraints.

This primary aim is to offer both a visual representation, through the class diagram, showcasing relationships among classes, and a detailed, structured view via the data dictionary, providing in-depth insights into the attributes and constraints within the object model. By combining these representations, it seeks to enhance comprehension and facilitate effective communication about the system's structure and behavior.

#### **4.2.1. Class Diagram**

In this subsection, our aim is to visually represent the conceptual connections among objects and classes. This includes excluding details such as data and member visibility, associations, generalizations, cardinality, and other related aspects in order to provide a simplified yet informative diagrammatic illustration.

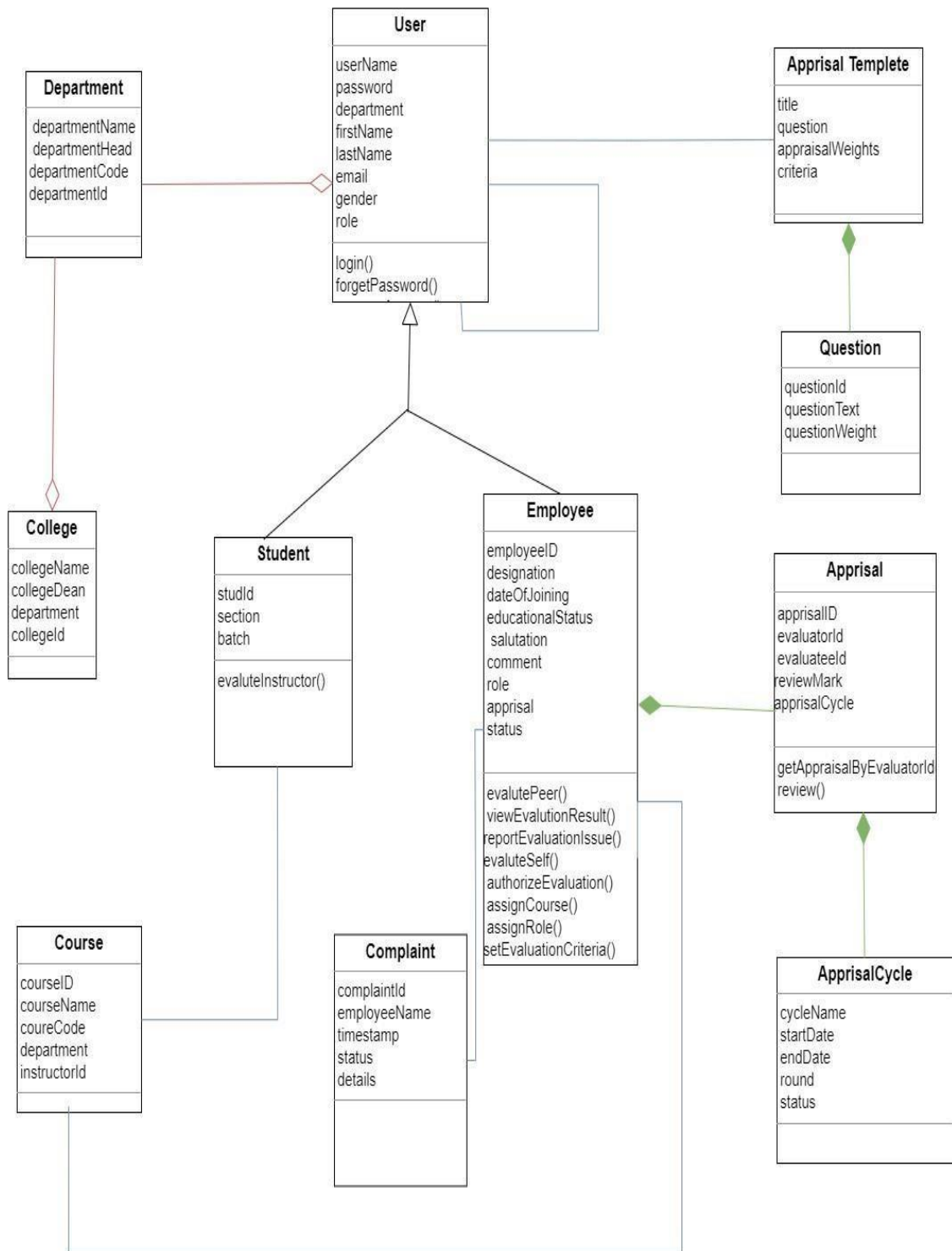


Figure 4. 2 high level class diagram

### 4.2.2. Data Dictionary

Here's the data dictionary for our Employee Performance Appraisal System, broken down into multiple tables for optimized storage and querying in MongoDB

*Table 4. 16 Users*

<b>Field</b>	<b>Data Type</b>	<b>Approx. Size</b>	<b>Key Constraint</b>	<b>Constraints</b>
user_id	ObjectId	12 bytes	Primary Key	Unique, Not Null
userName	String	Varies	-	Unique, Not Null
designation	String	Varies	-	-
status	String	Varies	-	-
departmentId	ObjectId	12 bytes	Foreign Key	References Department(departmentId)
email	String	Varies		Unique
salutation	String	Varies	-	-
dateOfJoining	Date	8 bytes	-	Not Null
photo	String	Varies	-	-
course_attended	String	Varies	-	-
password	String	Varies	-	Unique, Not Null
role	String	Varies		Not Null
active	Boolean	1 byte	-	Not Null

Table 4. 17 Department

Field	Data Type	Approx. Size	Key Constraint	Constraints
department_id	ObjectId	12 bytes	Primary Key	Unique, Not Null
departmentName	String	Varies	-	Unique, Not Null
departmentCode	String	Varies	-	Unique, Not Null
departmentHead	ObjectId	12 bytes	Foreign Key	References User(user_id)
collegeId	ObjectId	12 bytes	Foreign Key	References College(collegeId)

Table 4. 18 College

Field	Data Type	Approx. Size	Key Constraint	Constraints
college_id	ObjectId	12 bytes	Primary Key	Unique, Not Null
collegeName	String	Varies	-	Unique, Not Null
collegeDean	ObjectId	12 bytes	Foreign Key	References User(user_id)

Table 4. 19 Course

Field	Data Type	Approx. Size	Key Constraint	Constraints
course_id	ObjectId	12 bytes	Primary Key	Unique, Not Null
name	String	Varies	-	Not Null

Code	String	Varies	-	Not Null
instructorId	ObjectId	Varies	Foreign Key	References User(user_id)
Batch	Number	8 bytes	-	Not Null
Semester	Number	8 bytes	-	Not Null
department	ObjectId	12 bytes	Foreign Key	References Department(departmentId )
status	String	Varies	-	Not Null

*Table 4. 20 Appraisal Cycle*

<b>Field</b>	<b>Data Type</b>	<b>Approx. Size</b>	<b>Key Constraint</b>	<b>Constraints</b>
cycle_id	ObjectId	12 bytes	Primary Key	Unique, Not Null
description	String	Varies	-	-
startDate	Date	8 bytes	-	Not Null
endDate	Date	8 bytes	-	Not Null

*Table 4. 21 Evaluation Template*

<b>Field</b>	<b>Data Type</b>	<b>Approx. Size</b>	<b>Key Constraint</b>	<b>Constraints</b>
templete_Id	ObjectId	12 bytes	Primary Key	Unique, Not Null
evaluationType	String	Varies	-	Unique, Not Null
questions	Array	Varies		Not Null

Table 4. 22 Appraisal

<b>Field</b>	<b>Data Type</b>	<b>Approx. Size</b>	<b>Key Constraint</b>	<b>Constraints</b>
appraisal_id	ObjectId	12 bytes	Primary Key	Unique, Not Null
mark	Number	8 bytes	-	Not Null
cycle	ObjectId	12 bytes	Foreign Key	References AppraisalCycle (cycle_id)
evaluteeId	ObjectId	12 bytes	Foreign Key	References User(user_id)
evalutorId	ObjectId	12 bytes	Foreign Key	References User(user_id)

### 4.3. Dynamic Model

The dynamic model captures the system's time-sensitive elements, focusing on how the states of system objects evolve over time. This section aims to record the actions and interactions of the entity model using sequence, activity, and state chart diagrams

#### 4.3.1. Sequence Diagram

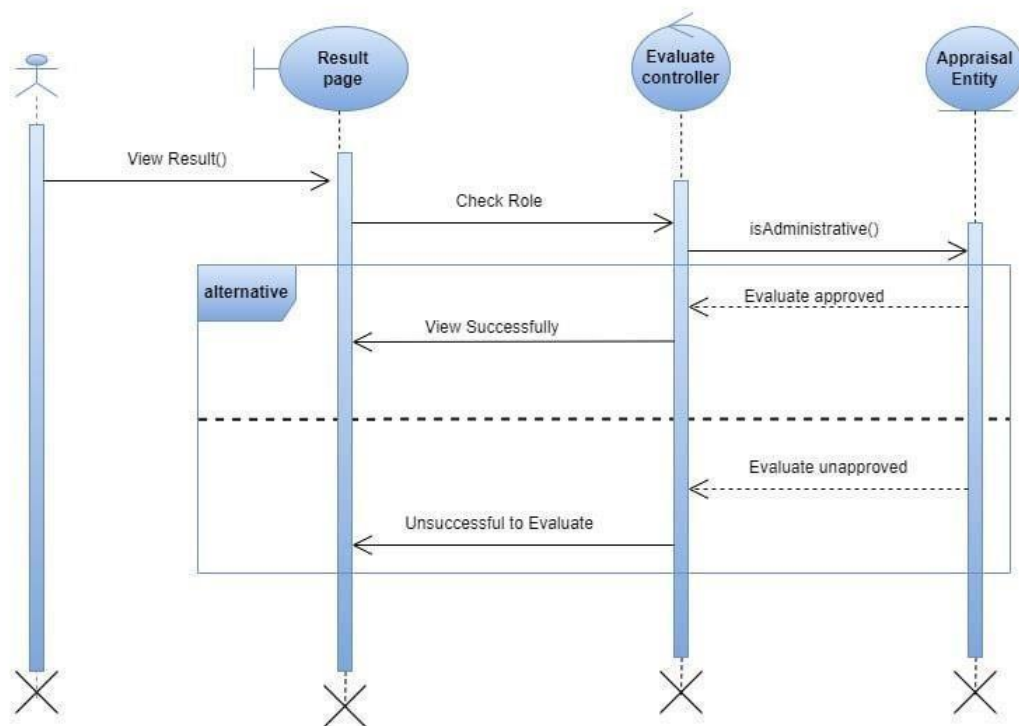


Figure 4. 3 Sequence Diagram (Self-Evaluation)

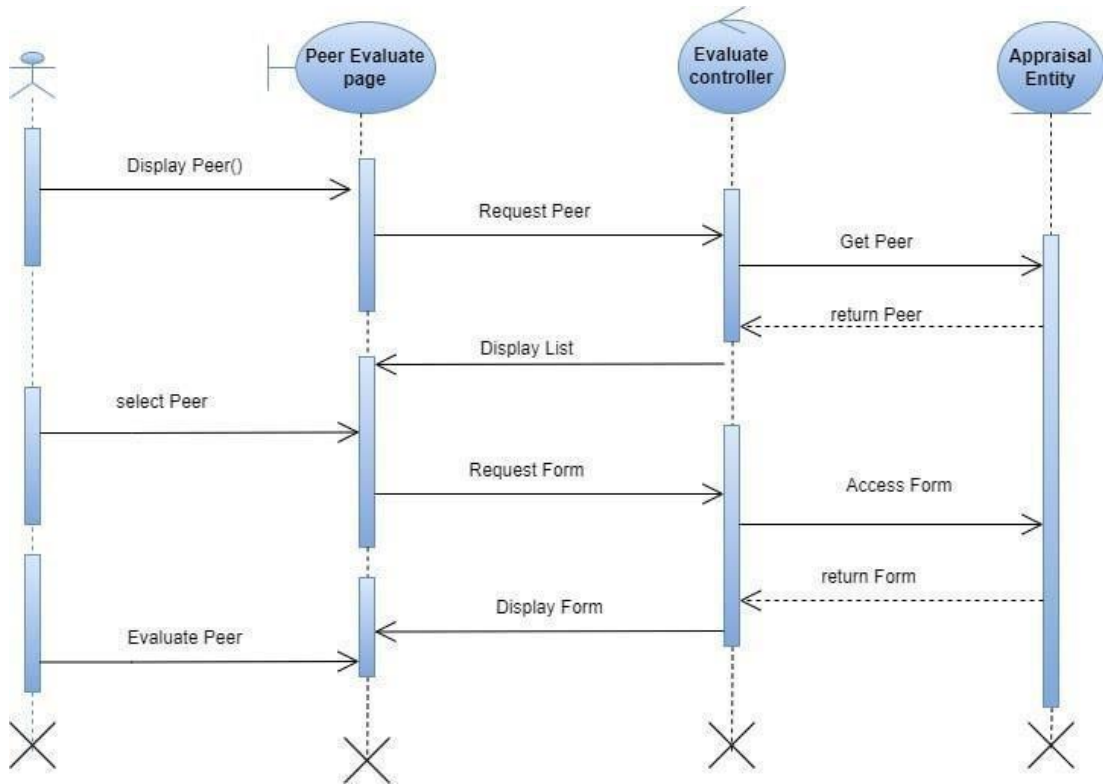
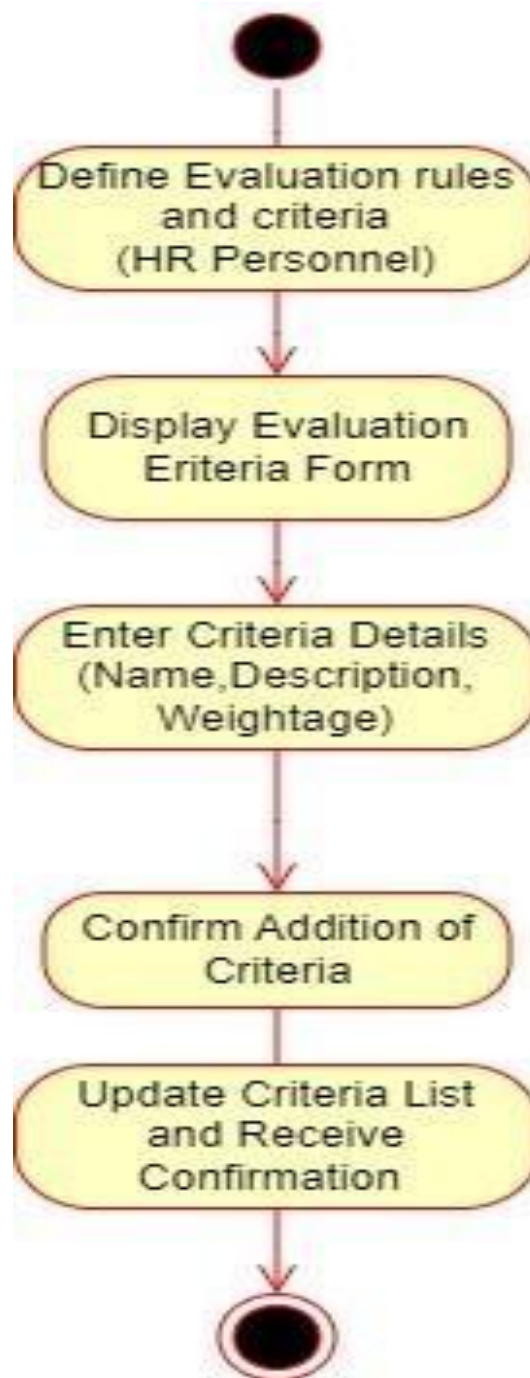


Figure 4. 4 Sequence Diagram (Peer Evaluation)

### 4.3.2 Activity Diagram



*Figure 4. 5 Activity Diagram (Define Evaluation rules*

# Evaluation

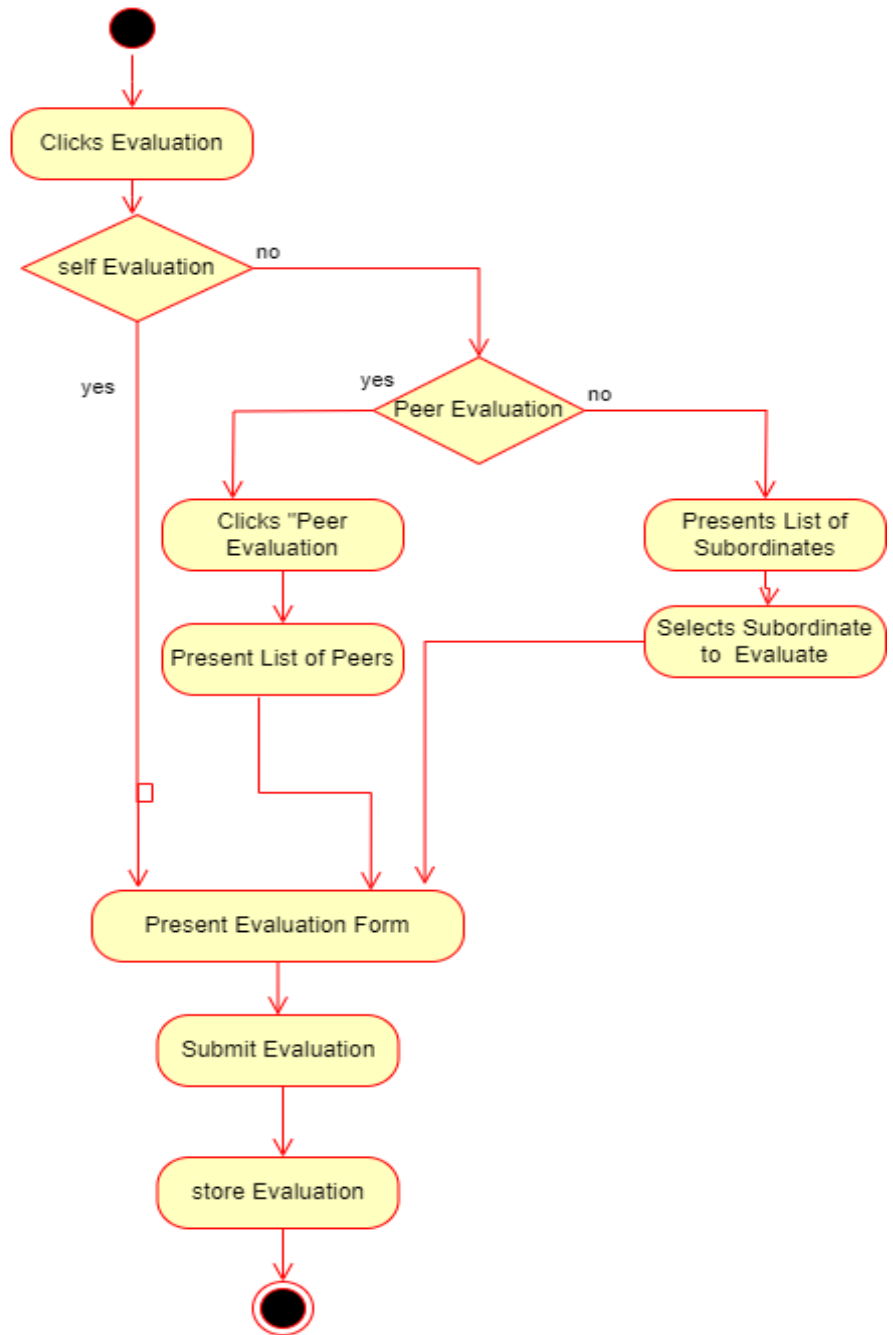


Figure 4. 6 Activity Diagram (Evaluation)

## View Appraisal Result

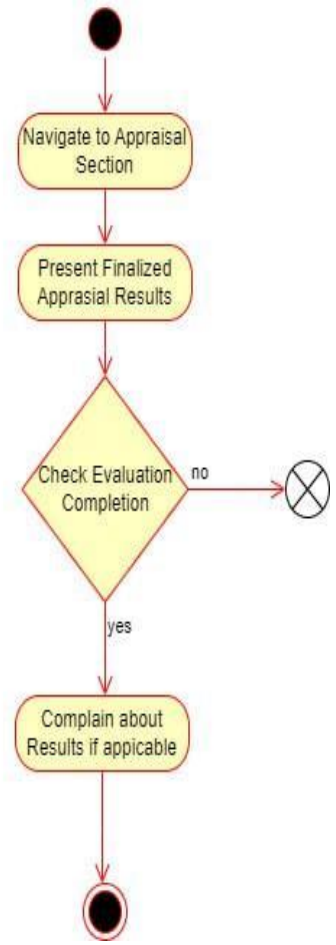
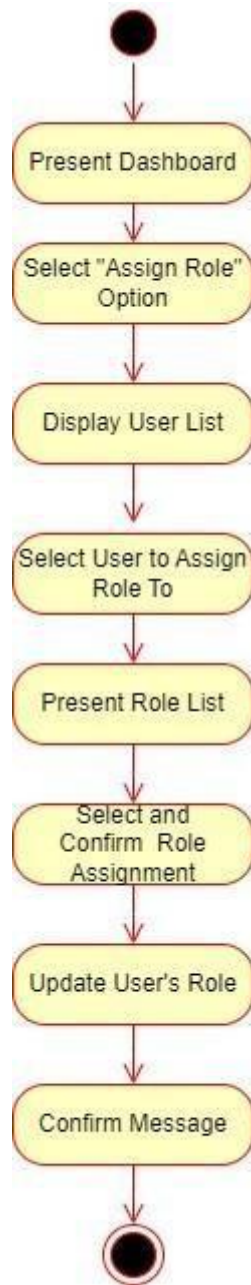


Figure 4. 7 Activity Diagram (View Appraisal Result)



*Figure 4. 8 Activity Diagram (Manage user)*

### 4.3.3 State Chart Diagram

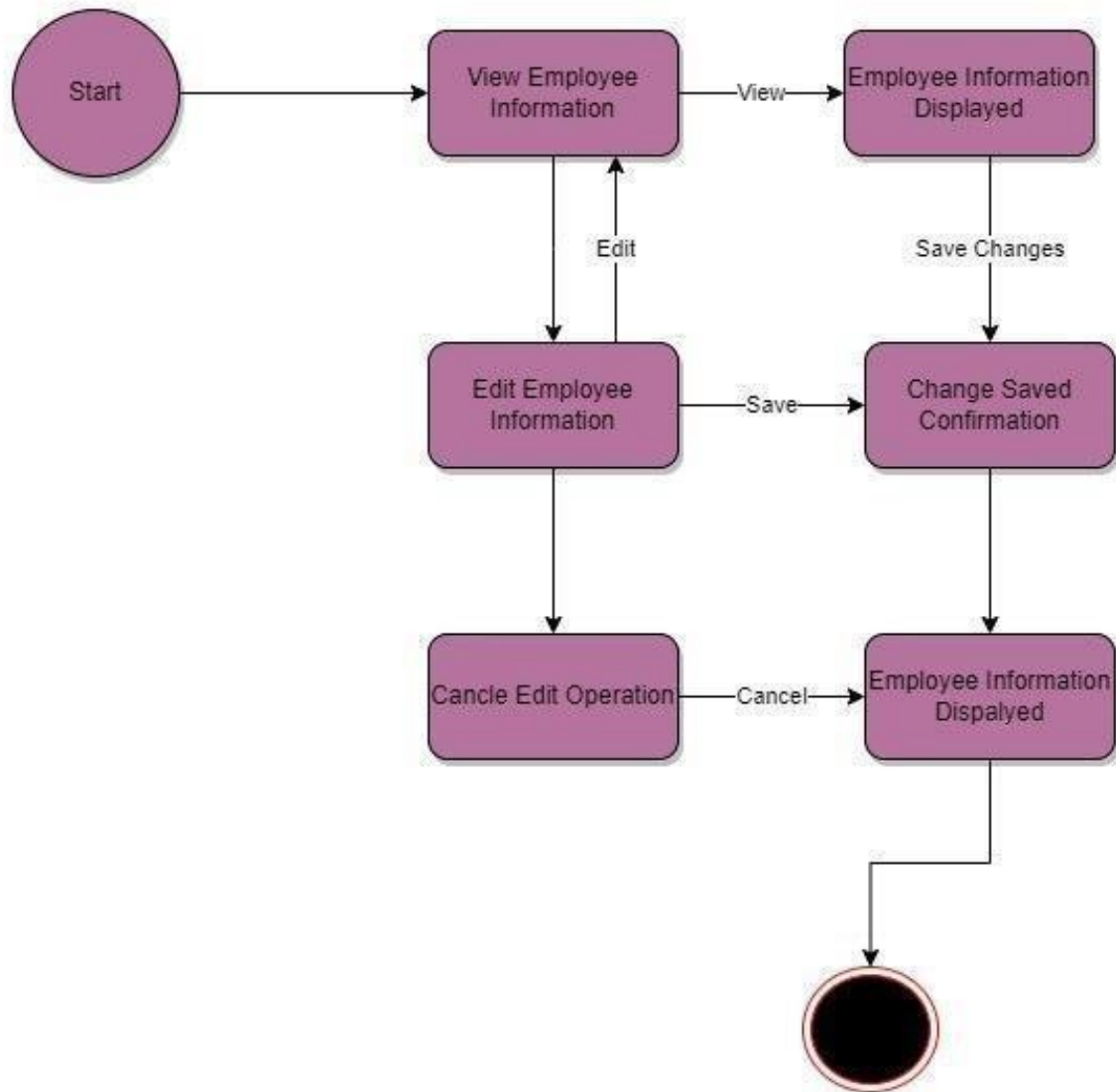
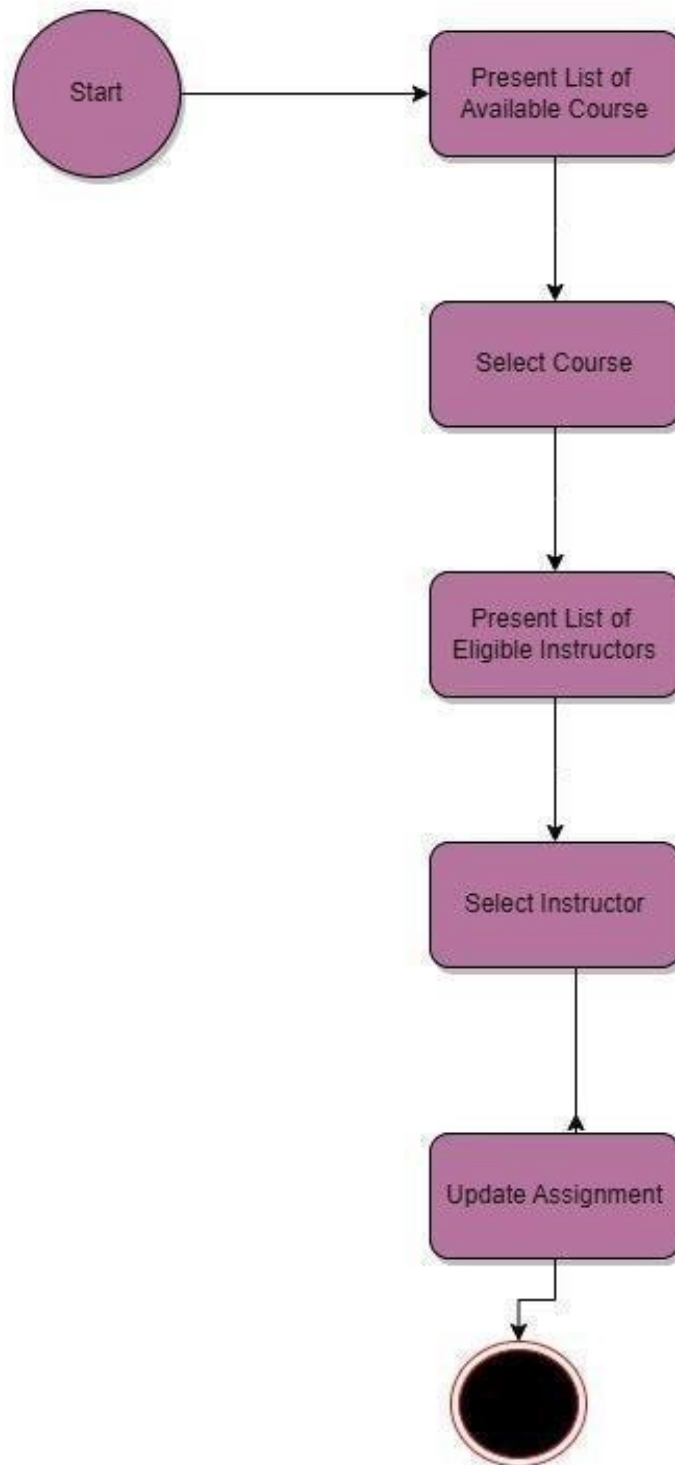


Figure 4. 9 State Diagram (Access Employee Information)



*Figure 4. 10 State Diagram (Assign Course)*

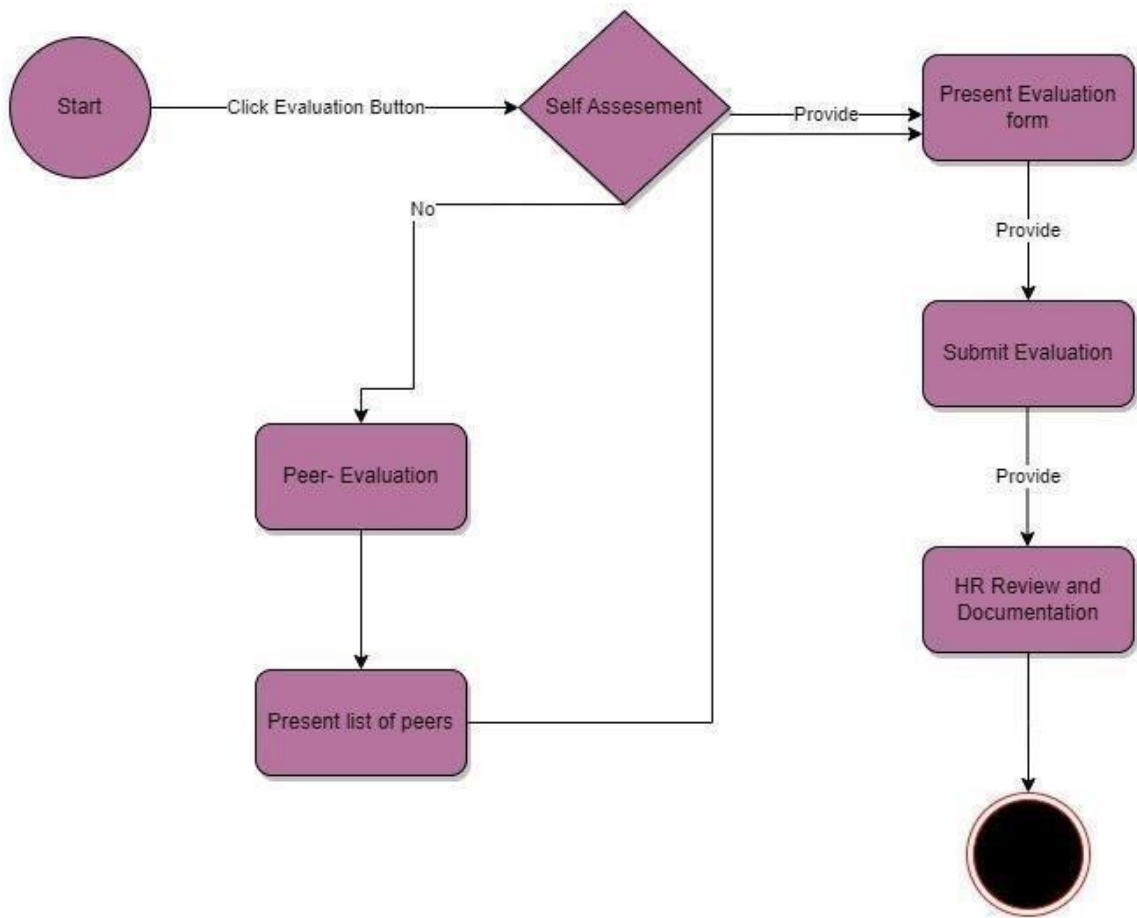


Figure 4. 11 State Diagram (evaluation Process)

# CHAPTER FIVE

## SYSTEM DESIGN

System Design involves outlining the framework for the architecture, modules, interfaces, and data structures necessary for a system to meet specified requirements. The process of Systems Design could be likened to applying system theory principles in the realm of product development. During this phase, a comprehensive collection of detailed information and insights regarding the system and its fundamental components is compiled. This wealth of information plays a vital role in ensuring that the subsequent implementation phase aligns seamlessly with the architectural entities delineated in various models and perspectives within the system architecture

### 5.1. Design Goals

Design goals describe the qualities of the system that developers should optimize. Such Goals are normally derived from the non-functional requirements of the system. In this design, the central focus revolves around three critical aspects

- ❖ Performance Consideration
- ❖ Security Issues
- ❖ Quality Considerations

#### 5.1.1. Performance Consideration

The appraisal system's performance aspect focuses on optimizing response times, throughput, and resource utilization for seamless operations. Key considerations encompass:

- ❖ **Response Time Optimization:** Prioritize quick response times within subsystems like "Result Generation" and "Reviewing," ensuring timely feedback and actions.
- ❖ **Scalability:** Design the system to handle increasing user loads during peak appraisal periods without compromising performance.
- ❖ **Resource Utilization:** Efficiently manage memory and processing resources to prevent system overload, particularly during heavy usage times such as end-of year appraisals.

### 5.1.2 Security Issues

Security remains paramount in preserving data integrity and confidentiality within the appraisal system. Strategies include:

- ❖ **Data Encryption:** Implement robust encryption protocols to safeguard sensitive employee data and appraisal records.
- ❖ **Access Control:** Utilize role-based access control to limit data access based on user roles (HR professionals, managers, employees) within the system.

### 5.1.3 Quality Considerations

#### **Robustness**

System will tolerate misuse without catastrophic failure. The misuse will be bad data, bad use or bad programming.

Robustness achieved in the following ways:

- ❖ By using data abstraction and encapsulation
- ❖ By initializing variables • Qualifying all inputs

#### **Efficiency**

PAS aims to make greatest use of the processing, memory size and network speed.

Efficiency will be achieved by: Writing clever algorithms and data structures.

#### **Reusability**

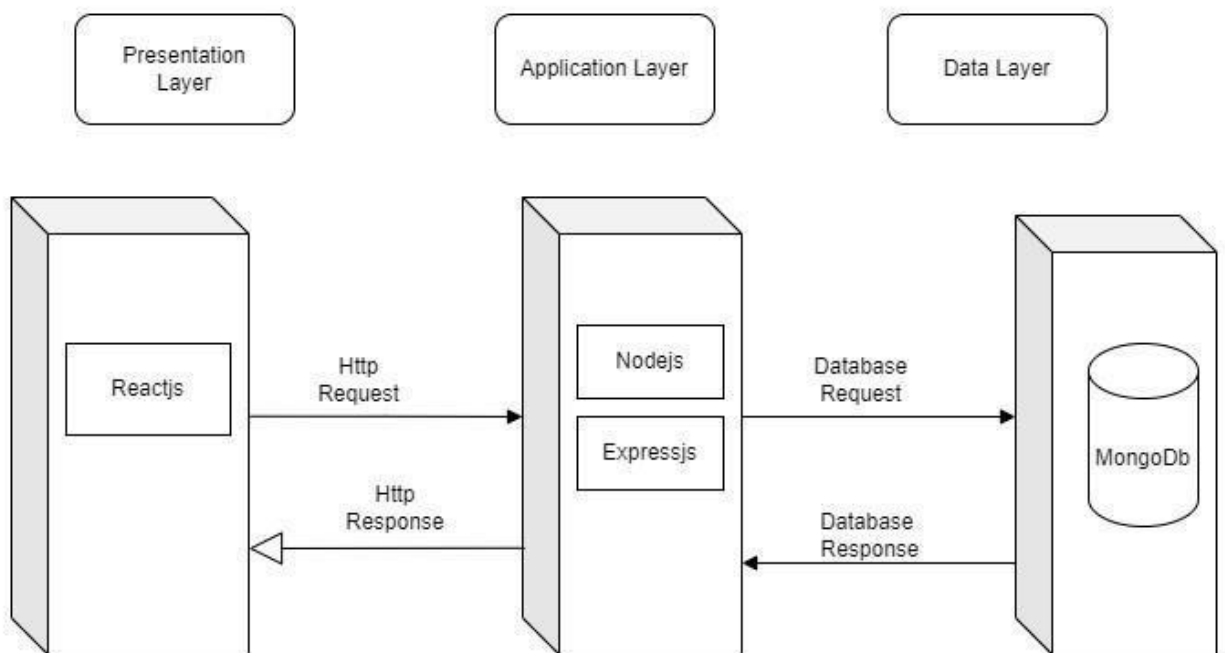
Systems aim is to cut the cost of code production over one or more projects.

The followings are the ways of reusability:

- ❖ Reuse source code,
- ❖ Reuse of Component,
- ❖ Reuse patterns of designs.

## 5.2 Proposed System Architecture

The 3-tier architecture is chosen for an employee appraisal system to provide a scalable, maintainable, and modular structure that aligns with the separation of concerns, making the system more robust and easier to manage over time. The decision is also influenced by the adaptability of this architecture to changing business requirements and advancements in technology. As the employee appraisal process evolves, the 3-tier structure allows for seamless integration of new features or adjustments without major disruptions to the existing system. Additionally, the modular nature of the architecture enables easy replacement or upgrading of individual tiers as needed, ensuring that the employee appraisal system remains agile and responsive to the dynamic nature of organizational appraisal needs and technological innovations.



*Figure 5. 1 System architecture*

### **5.2.1. Subsystem Decomposition and Description**

The following subsystems are part of the web-based automated Performance appraisal system

#### **User Management Subsystem**

Description: This subsystem manages user roles and permissions. It includes functionalities such as creating and updating user accounts for employees, department heads, HR professionals, and system administrators.

#### **User Authentication subsystem**

Description: It includes a user login and logout system.

#### **Result Generation Subsystem:**

Description: This subsystem automates the process of generating final employee appraisal results based on predefined rules and criteria. It ensures consistency and objectivity in the evaluation process.

#### **Appraisal Criteria Subsystem**

Description: This integrated subsystem empowers HR professionals to dynamically define and manage appraisal criteria and rules while facilitating the creation and distribution of customizable evaluation forms for various user roles. It ensures flexibility in adapting to changing organizational needs and supports a comprehensive approach to performance assessment.

#### **Complaint Handling Subsystem**

Description: This subsystem facilitates the submission and resolution of employee complaints related to the appraisal process. It ensures transparency and accountability in addressing concerns.

#### **Generate Report Subsystem**

Description: This subsystem generates comprehensive reports on employee appraisals. It provides insights for decision-makers and supports data-driven improvement strategies.

#### **Security and Compliance Subsystem**

Description: This subsystem addresses security concerns and ensures compliance with data protection regulations. It implements measures to protect employee data and maintain confidentiality.

**System Performance Subsystem:**

Description: This subsystem addresses performance considerations, ensuring that the system is responsive, scalable, and can recover from errors efficiently.

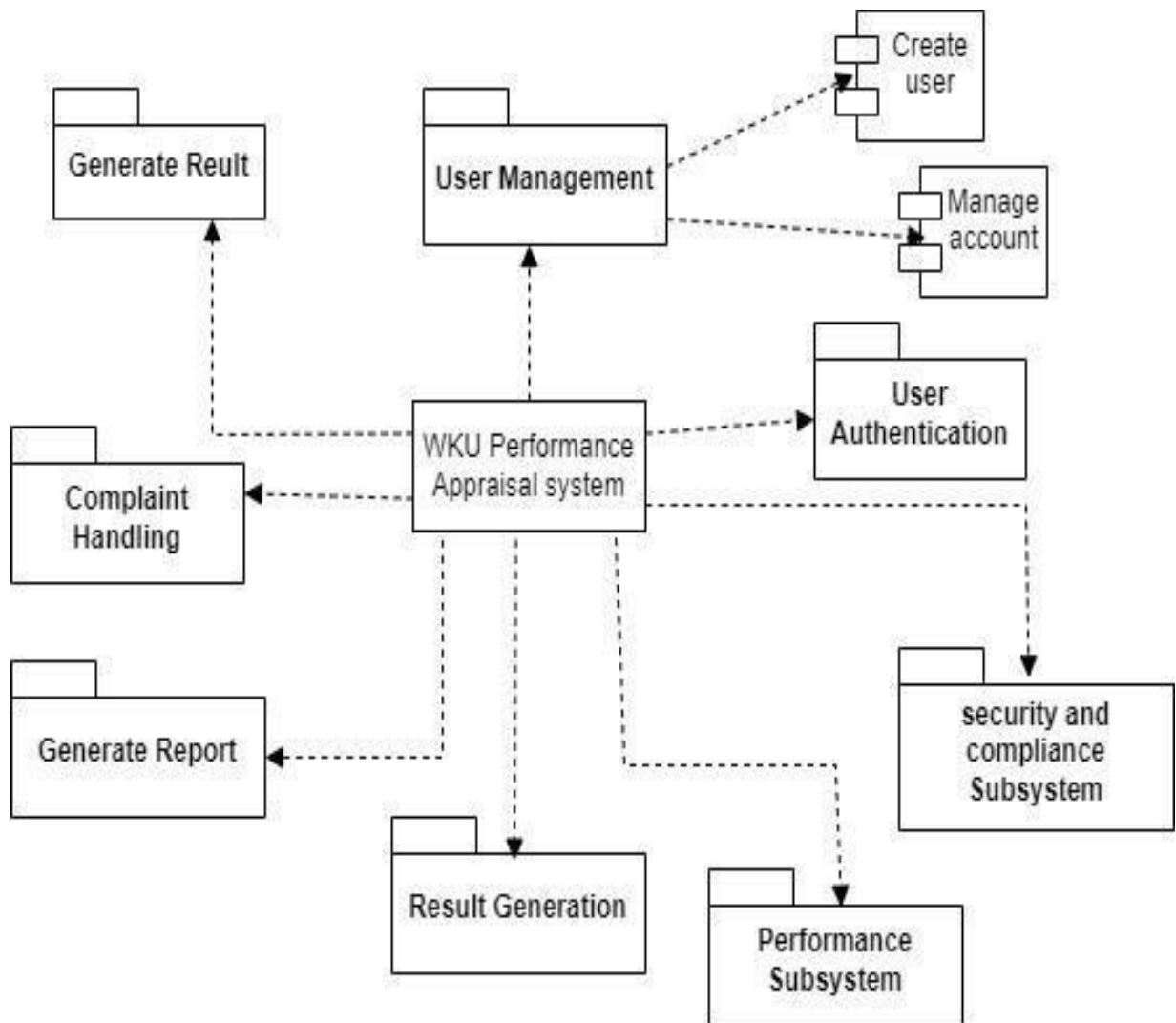
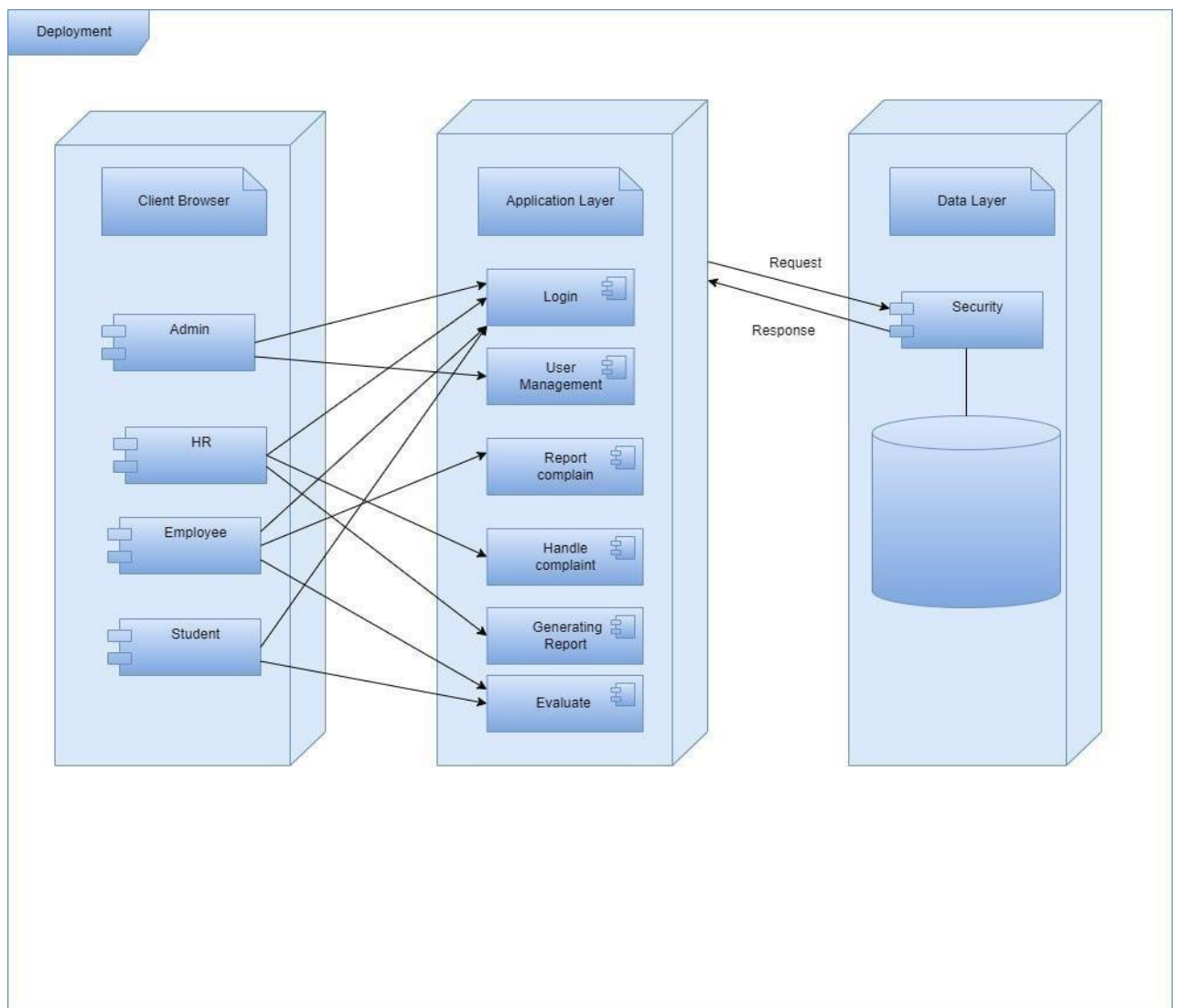


Figure 5. 2 Component Diagram

## 5.2.2. Hardware/Software Mapping



*Figure 5. 3 Hardware /Software mapping*

### 5.2.3. Detailed Class Diagram

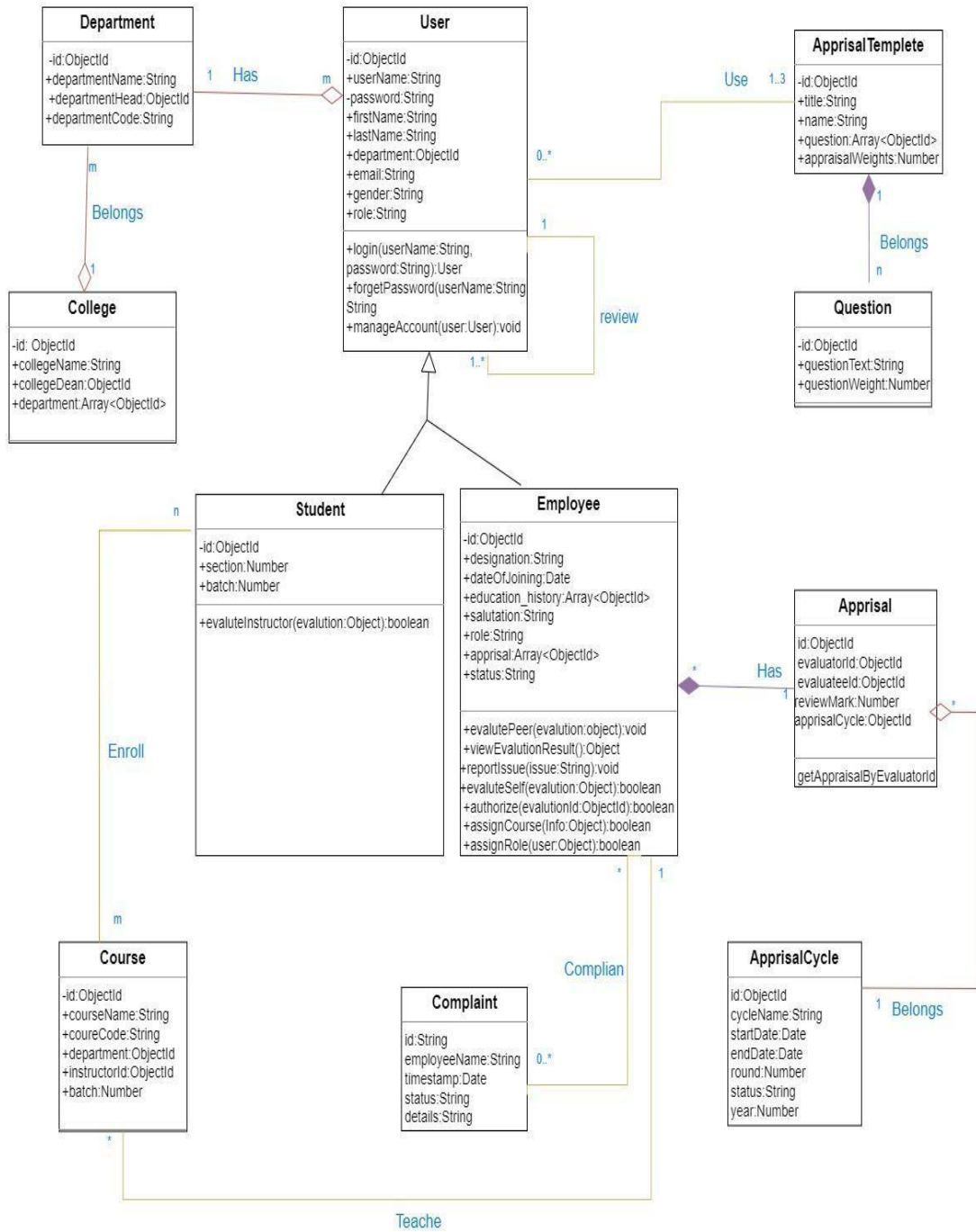


Figure 5. 4 Detailed Class Diagram

#### **5.2.4. Persistent Data Management**

An organized collection of structured information, commonly referred to as data, is typically electronically stored in a computer system, constituting a database. Usually managed by DBMS, the database, along with associated applications, is collectively termed a database system. Mongo DB was utilized in managing this particular system. Database design, involving the classification of data and identification of interrelationships, comprises the organization of data based on a database model. This activity encompasses the representation of classes, attributes, and relationships within a database. An Entity ER diagram, also recognized as an entity relationship model, is a graphical representation of an information system illustrating the relationships among people, objects, places, concepts, or events within that system.

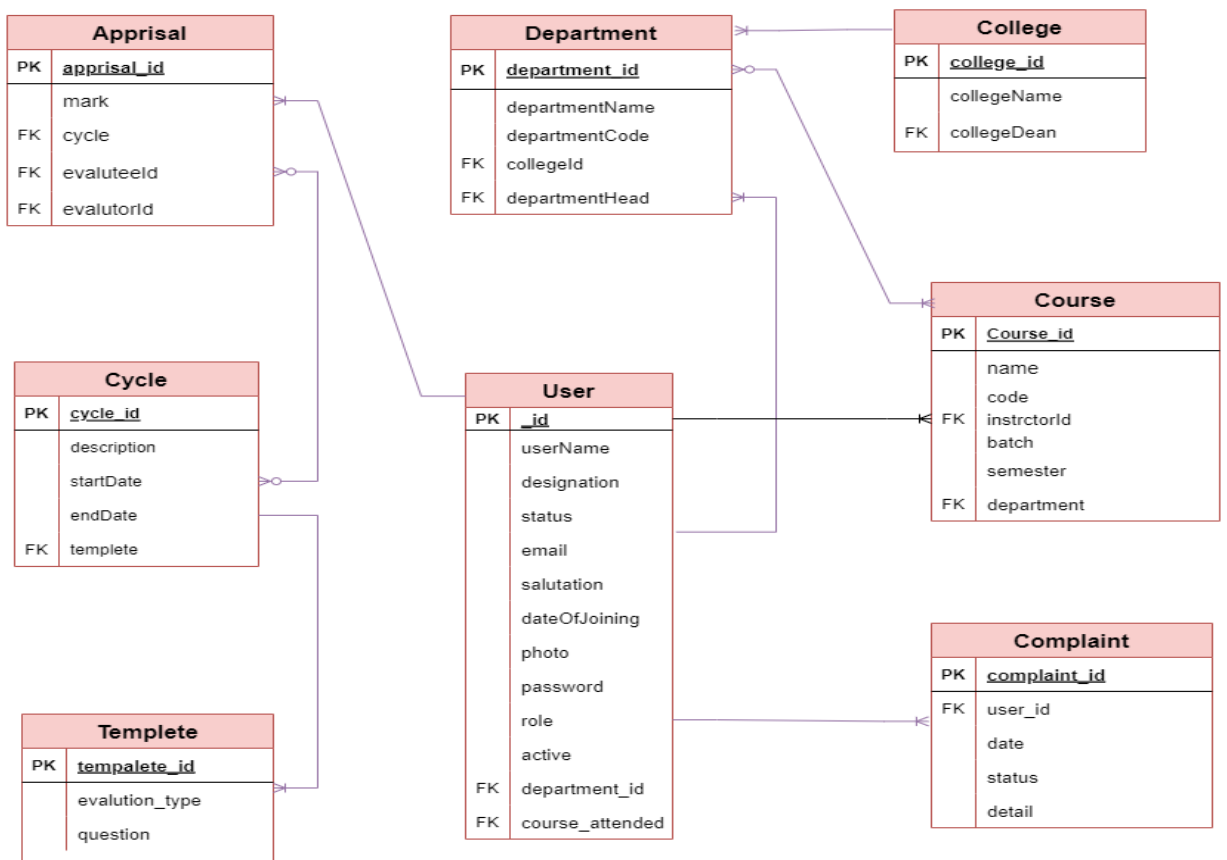


Figure 5. 5 Persistence Data Model

### 5.2.5. Access control and Security

Different users inside this system are granted varying levels of access to the system's features. By linking various use cases to various actors during analysis, we modeled these distinctions. When designing a system, we model access by looking at the object model, by deciding which items are shared by actors, and specifying how actors can manage access.

We also provide the method through which actors are authenticated based on the security requirements of the system.

Based on the user's username and password, the system will recognize them and display the window that is suitable for that specific user.

In general, we need to specify which operations each actor can access. We simulate access class control using a table.

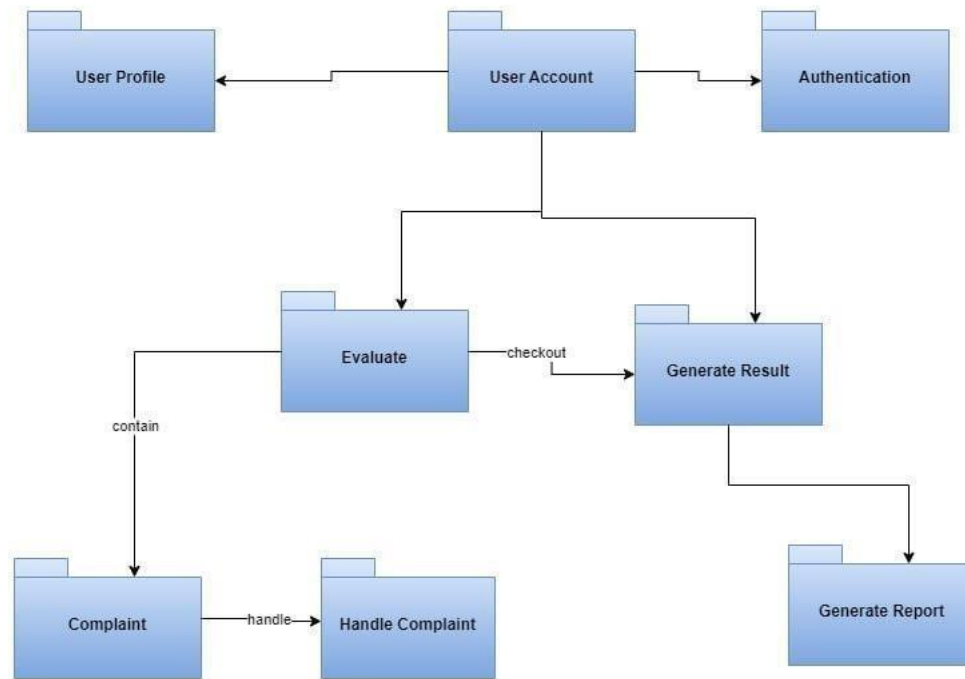
*Table 5. 1 Access control*

Actors	Verify Employee Account	Initiate Appraisal Cycle	View Employee Details	Record Appraisal	Log in	Submit report	manage system attribute
Admin	<b>yes</b>		<b>yes</b>		<b>yes</b>		<b>yes</b>
HR Personnel		<b>yes</b>	<b>yes</b>	<b>yes</b>	<b>yes</b>		
Student					<b>yes</b>	<b>yes</b>	
Head					<b>yes</b>	<b>yes</b>	
Employee					<b>yes</b>	<b>yes</b>	



### 5.3. Packages

Package diagram shows the decomposition of subsystems into packages and overview of each package dependencies with other packages.



*Figure 5.6 Package diagram*

## 5.4. Algorithm Design

```
1 FUNCTION ViewAppraisalResult(employee):
2     userLogin = AuthenticateEmployeeLogin(employee)
3
4     IF userLogin == Successful THEN
5         evaluationCompleted = CheckEvaluationCompletion(employee)
6         evaluationFinalized = CheckEvaluationFinalization(employee)
7
8         IF evaluationCompleted AND evaluationFinalized THEN
9             NavigateToAppraisalSection(employee)
10            finalizedResults = RetrieveFinalizedAppraisalResults(employee)
11
12            IF finalizedResults is Available THEN
13                PresentAppraisalResults(finalizedResults)
14                EmployeeReviewsAppraisalResults()
15                RETURN "Appraisal Results Reviewed Successfully"
16            ELSE
17                RETURN "No Appraisal Results Available"
18            END IF
19
20        ELSE
21            DisplayMessage("Appraisal results not available yet")
22            RETURN "Pre-conditions not met"
23        END IF
24
25    ELSE
26        RETURN "Login Failed"
27    END IF
28 END FUNCTION
29
```

*Figure 5. 7 View Appraisal Result*

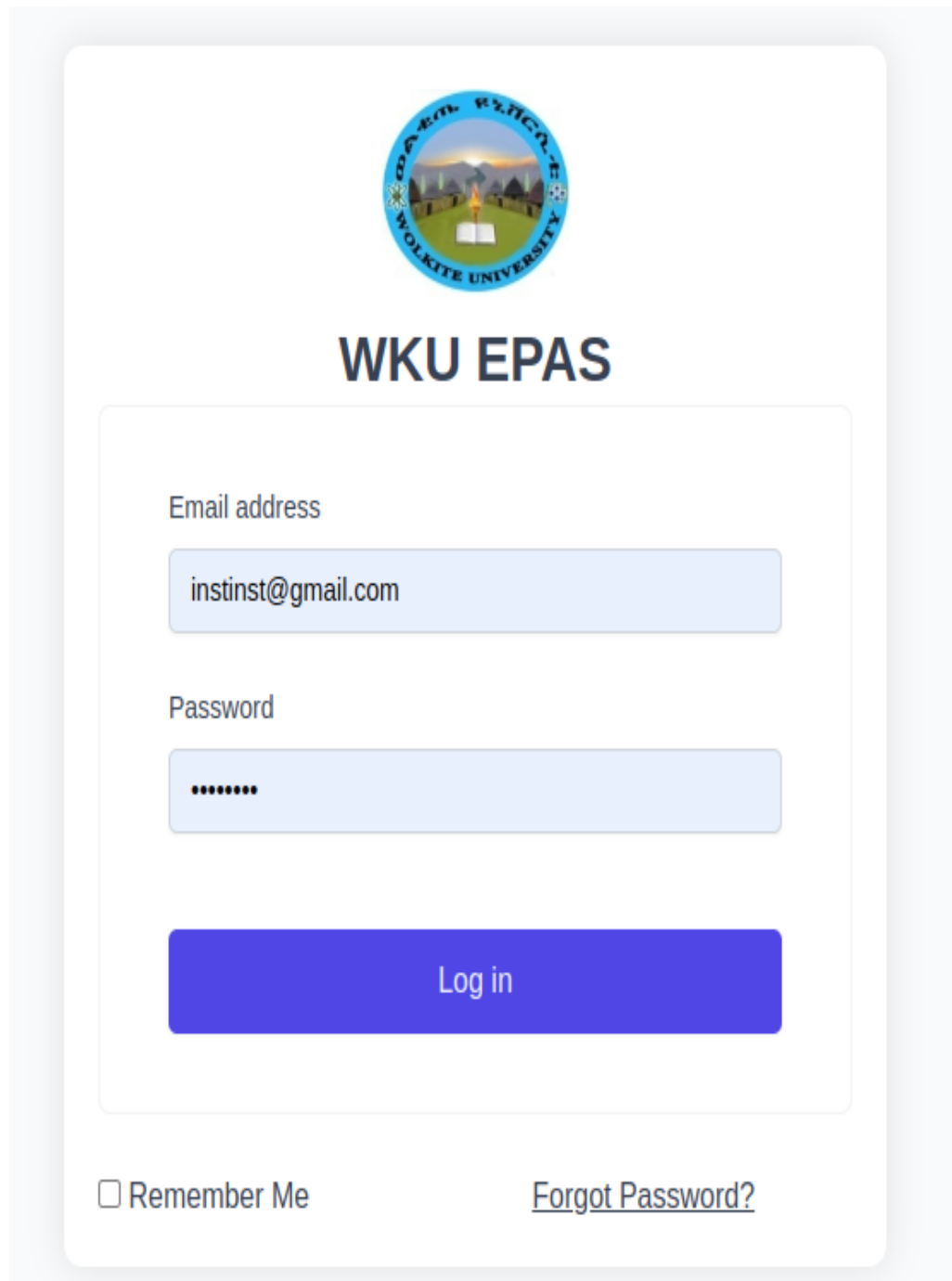
```

1  FUNCTION ViewAppraisalResult(employee):
2      userLogin = AuthenticateEmployeeLogin(employee)
3
4      IF userLogin == Successful THEN
5          evaluationCompleted = CheckEvaluationCompletion(employee)
6          evaluationFinalized = CheckEvaluationFinalization(employee)
7
8          IF evaluationCompleted AND evaluationFinalized THEN
9              NavigateToAppraisalSection(employee)
10             finalizedResults = RetrieveFinalizedAppraisalResults(employee)
11
12             IF finalizedResults is Available THEN
13                 PresentAppraisalResults(finalizedResults)
14                 EmployeeReviewsAppraisalResults()
15                 RETURN "Appraisal Results Reviewed Successfully"
16             ELSE
17                 RETURN "No Appraisal Results Available"
18             END IF
19
20         ELSE
21             DisplayMessage("Appraisal results not available yet")
22             RETURN "Pre-conditions not met"
23         END IF
24
25     ELSE
26         RETURN "Login Failed"
27     END IF
28 END FUNCTION
29

```

*Figure 5. 8 Perform Evaluation*

## 5.5. User Interface Design



The image shows a login page for WKU EPAS. At the top center is the WKU logo, a circular emblem with a blue border containing the text 'WOLKITE UNIVERSITY' and 'WOLKITE UNIVERSITY' at the bottom. The center of the logo features a landscape with a sun, a path, and a book. Below the logo, the text 'WKU EPAS' is displayed in a large, bold, dark blue font. The main login area is enclosed in a white rounded rectangle with a light blue border. It contains two input fields: 'Email address' with the text 'instinst@gmail.com' and 'Password' with a masked password '.....'. Below these fields is a prominent blue 'Log in' button. At the bottom left of the login area, there is a checkbox labeled 'Remember Me'. At the bottom right, there is a link labeled 'Forgot Password?'.

Figure 5. 9 Login page design

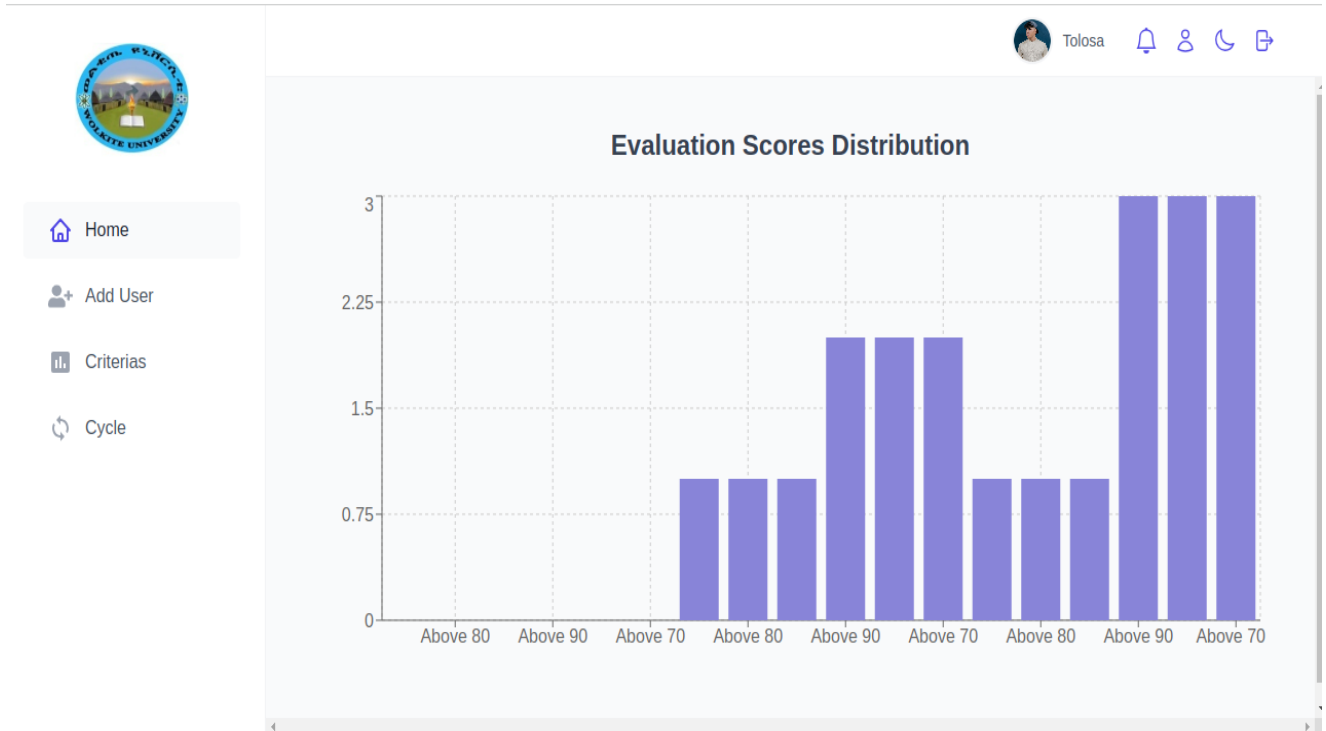


Figure 5. 10 Dashboard design

# CHAPTER SIX

## IMPLEMENTATION AND TESTING

The implementation phase of the project transforms conceptualization and planning into functional software components. Node.js acts as the JavaScript runtime environment for server-side development, enabling execution outside web browsers. MongoDB, chosen as the NoSQL database solution, offers scalability and flexibility in data storage. On the frontend, React is employed as the primary JavaScript library for building user interfaces, facilitating reusable UI elements and efficient state management. React seamlessly interacts with the backend, comprising Node.js and MongoDB, to retrieve and present data to users. Together, these technologies create a robust software product, managing data storage, business logic, and frontend presentation effectively for a seamless user experience.

### 6.1. Implementation of the Database

- ❖ Choice: MongoDB
- ❖ Justification: MongoDB has been selected as the database management system for the performance appraisal application due to its scalability, flexibility, and ability to handle semi-structured data, which is beneficial for managing various types of appraisal-related information.

#### 6.1.1 Table Creation and Constraints

**AppraisalCycle:** Stores information about appraisal cycles.

- ❖ Primary Key: `_id` field.
- ❖ Unique Constraints: None.
- ❖ Indexes: None.
- ❖ Foreign Keys: None.

**FinalResult:** Captures final appraisal results.

- ❖ Primary Key: `_id` field.
- ❖ Unique Constraints: None.

- ❖ Indexes: Created on cycle, evaluatedUserName, evaluator, department, and ApprovedBy for efficient querying.

- ❖ Foreign Keys

  - A. References AppraisalCycle collection through the cycle field.

  - B. References User collection through evaluatedUserName, evaluator, and ApprovedBy fields.

  - C. References Department collection through the department field.

**Complaint:** Stores information about user complaints.

- ❖ Primary Key: \_id field.

- ❖ Unique Constraints: None.

- ❖ Indexes: Created on the userId field to optimize queries.

- ❖ Foreign Keys: References the User collection through the userId field

**Course:** Represents courses offered.

- ❖ Primary Key: \_id field.

- ❖ Unique Constraints: None.

- ❖ Indexes: Compound index on code, batch, section, and department for uniqueness and query performance.

- ❖ Foreign Keys: None.

**Notification:** Tracks user notifications.

- ❖ Primary Key: \_id field.

- ❖ Unique Constraints: None.

- ❖ Indexes: Created on the user field for faster queries.

- ❖ Foreign Keys: References the User collection through the user field.

**College:** Represents colleges within the organization.

- ❖ Primary Key: \_id field.

- ❖ Unique Constraints: None.
- ❖ Indexes: None.
- ❖ Foreign Keys: None.

**Department:** Represents departments within the organization.

- ❖ Primary Key: `_id` field.
- ❖ Unique Constraints: Enforced on the `departmentCode` field for uniqueness.
- ❖ Indexes: Unique index on the `departmentCode` field to guarantee uniqueness.
- ❖ Foreign Keys: References the `College` collection through the `collegeId` field.

**User:** Stores user information.

- ❖ Primary Key: `_id` field.
- ❖ Unique Constraints: Enforced on the `email` field to ensure unique email addresses.
- ❖ Indexes: Created on the `email` field for faster lookups.
- ❖ Foreign Keys: None.
- ❖ Foreign Keys
  - A. Reference the `Department` collection with the `department` field.
  - B. Reference the `College` collection with the `college` field.

**Appraisal Template:** Stores information about appraisal templates.

- ❖ Primary Key: The `_id` field serves as the primary key for each template.
- ❖ Unique Constraints: No unique constraints are enforced.
- ❖ Indexes: Indexes are created on the `evaluationType` and `language` fields for faster lookups.
- ❖ Foreign Keys: None.

### **6.1.2. Normalization**

The database strives to achieve at least the Third Normal Form (3NF) whenever possible. Complex documents are decomposed into smaller, related documents to minimize redundancy and maintain data consistency. For instance, the FinalResult collection stores only foreign key references to related entities (AppraisalCycle, User, Department) instead of duplicating their entire data.

### **6.1.3. Database Level Security**

- ❖ Authentication: Email and password mechanisms control user access.
- ❖ Authorization: Roles and permissions are defined to restrict access to sensitive data and operations based on user roles.
- ❖ Encryption: Encryption at rest and in transit safeguards sensitive information from unauthorized access.

## 6.2. Implementation of the Class Diagram

```
1 import mongoose from "mongoose";
2 const appraisalCycleSchema = new mongoose.Schema({
3   status: {
4     type: String,
5     enum: ["planned", "active", "completed"],
6     required: true,
7   },
8   description: String,
9   startDate: {
10    type: Date,
11    required: true,
12  },
13  endDate: {
14    type: Date,
15    required: true,
16    validate: {
17      validator: function (value) {
18        return this.startDate ? value > this.startDate : true;
19      },
20      message: "End date must be after the start date",
21    },
22  },
23 });
24 appraisalCycleSchema.pre("save", async function (next) {
25   const currentDate = new Date();
26
27   if (currentDate >= this.startDate && currentDate <= this.endDate) {
28     this.status = "active";
29   }
30   if (this.status === "active") {
31     await this.constructor.updateMany(
32       { status: "active", _id: { $ne: this._id } },
33       { $set: { status: "completed" } }
34     );
35   }
36   next();
37 });
38
39 const AppraisalCycle = mongoose.model("AppraisalCycle", appraisalCycleSchema);
40
41 export default AppraisalCycle;
42
```

Figure 6 1 AppraisalCycle Model

```

1  import mongoose from "mongoose";
2  const resultDetailSchema = new mongoose.Schema({
3    cycle: {
4      type: mongoose.Schema.Types.ObjectId,
5      ref: "AppraisalCycle",
6      required: true,
7    },
8    evaluatedUserName: {
9      type: mongoose.Types.ObjectId,
10     ref: "User",
11   },
12   evaluator: {
13     type: mongoose.Schema.Types.ObjectId,
14     ref: "User",
15     required: true,
16   },
17
18   department: {
19     type: mongoose.Types.ObjectId,
20     ref: "Department",
21   },
22
23   byStudent: {
24     total: { type: Number, default: 0 },
25     countOfReviewer: { type: Number, default: 0 },
26   },
27   byPeer: {
28     total: { type: Number, default: 0 },
29     countOfReviewer: { type: Number, default: 0 },
30   },
31   byHead: Number,
32   byTeamLeader: Number,
33   byDean: Number,
34   byDirector: Number,
35   bySelf: Number,
36   Average: Number,
37   status: {
38     type: String,
39     enum: ["In Progress", "Completed", "Approved"],
40     default: "In Progress",
41   },
42   byPeerRank: { type: Number },
43   byStdRank: { type: Number },
44   byHeadRank: { type: Number },
45   byDeanRank: Number,
46   bySelfRank: Number,
47   byTeamLeaderRank: { type: Number },
48   byDirectorRank: { type: Number },
49
50   ApprovedBy: {
51     type: mongoose.Types.ObjectId,
52     ref: "User",
53   },
54
55   ApprovedDate: Date,
56 });
57
58 export default FinalResult;
59

```

Figure 6 2 Appraisal Model

```

1 import crypto from "crypto";
2 import mongoose from "mongoose";
3 import validator from "validator";
4 import bcrypt from "bcryptjs";
5
6 const userSchema = new mongoose.Schema({
7   firstName: {
8     type: String,
9     required: [true, "firstName should be given!"],
10  },
11  lastName: {
12    type: String,
13    required: [true, "last name must be given!"],
14  },
15  salutation: {
16    type: String,
17  },
18  dateOfJoining: {
19    type: Date,
20    default: Date.now,
21  },
22  experience: {
23    type: Number,
24    default: 0,
25  },
26  designation: {
27    type: String,
28  },
29  branch: String,
30  address: String,
31  email: {
32    type: String,
33    unique: true,
34    lowercase: true,
35    required: [true, "Email must be given"],
36    validate: [validator.isEmail, "Please provide a valid email"],
37  },
38  gender: {
39    type: String,
40  },
41  },
42  phone: Number,
43  age: Number,
44  section: Number,
45  companyEmail: String,
46  branch: {
47    type: String,
48  },
49  },
50  role: {
51    type: String,
52    enum: [
53      "student",
54      "administrative",
55      "academic",
56      "instructor",
57      "assistance",
58      "admin",
59      "director",
60      "teamLeader",
61      "head",
62      "dean",
63      "hr",
64    ],
65  },
66  batch: {
67    type: Number,
68    min: 1,
69  },
70  },
71  department: {
72    type: mongoose.Schema.Types.ObjectId,
73    ref: "Department",
74  },
75  college: {
76    type: mongoose.Schema.Types.ObjectId,
77    ref: "College",
78  },
79  },
80  password: {
81    type: String,
82    minlength: 4,
83    select: false,
84  },
85  passwordChangedAt: Date,
86  passwordResetToken: String,
87  passwordResetExpires: Date,
88  avatar: String,
89  avatarPublicId: String,
90  active: {
91    type: Boolean,
92    default: true,
93    select: false,
94  },
95  });
96
97 userSchema.index({ email: 1 });
98
99 const User = mongoose.model("User", userSchema);
100 export default User;
101

```

Figure 6 3 User Model

```

1 import mongoose from "mongoose";
2 import validator from "validator";
3 const complaintSchema = new mongoose.Schema({
4   issueDate: {
5     type: Date,
6     required: true,
7     default: Date.now,
8   },
9   resolvedDate: {
10    type: Date,
11    validate: {
12      validator: function (value) {
13        return !this.issueDate || value >= this.issueDate;
14      },
15      message: "Resolved date must be equal to or after the issue date",
16    },
17  },
18  status: {
19    type: String,
20    enum: ["open", "resolved", "in-progress", "pending", "closed"],
21    default: "open",
22    required: true,
23  },
24  resolvedText: String,
25  userId: {
26    type: mongoose.Schema.Types.ObjectId,
27    ref: "User",
28    required: true,
29    index: true,
30  },
31  detail: {
32    type: String,
33    required: true,
34    minlength: 10,
35    maxlength: 1000,
36  },
37 });
38
39 const Complaint = mongoose.model("Complaint", complaintSchema);
40
41 export default Complaint;
42

```

*Figure 6 4 Complaint Model*

```

1  import mongoose, { mongo } from "mongoose";
2  import validator from "validator";
3  const courseSchema = new mongoose.Schema(
4    {
5      name: {
6        type: String,
7        required: true,
8        trim: true,
9      },
10     code: {
11       type: String,
12       required: true,
13       trim: true,
14     },
15     semester: {
16       type: Number,
17       required: true,
18     },
19     batch: {
20       type: Number,
21       required: true,
22       min: 1,
23     },
24     department: {
25       type: mongoose.Schema.Types.ObjectId,
26       ref: "Department",
27     },
28
29     instructor: {
30       type: mongoose.Schema.Types.ObjectId,
31       ref: "User",
32     },
33     section: {
34       type: Number,
35       default: 1,
36     },
37     startDate: {
38       type: Date,
39       required: true,
40       validate: {
41         validator: function (value) {
42           return !this.endDate || value < this.endDate;
43         },
44         message: "Start date must be before end date",
45       },
46     },
47     endDate: {
48       type: Date,
49       required: true,
50       validate: {
51         validator: function (value) {
52           return !this.startDate || value > this.startDate;
53         },
54         message: "End date must be after start date",
55       },
56     },
57   },
58   { toJSON: { virtuals: true }, toObject: { virtuals: true } }
59 );
60
61 courseSchema.index(
62   { code: 1, batch: 1, section: 1, department: 1 },
63   { unique: true }
64 );
65
66 courseSchema.virtual("isActive").get(function () {
67   const currentDate = new Date();
68   return this.startDate <= currentDate && this.endDate >= currentDate;
69 });
70 const Course = mongoose.model("Course", courseSchema);
71 export default Course;
72
73

```

Figure 6 5 Course Model

```
1 import mongoose from "mongoose";
2 const collegeSchema = new mongoose.Schema({
3   collegeName: {
4     type: String,
5     required: true,
6     trim: true,
7   },
8   numberOfDepartment: Number,
9   dean: String,
10  collegeCode: {
11    type: String,
12    trim: true,
13    unique: true,
14  },
15 });
16
17 const College = mongoose.model("College", collegeSchema);
18
19 export default College;
20
```

*Figure 6 6 College Model*

```

1  import mongoose from "mongoose";
2  const appraisalTemplateSchema = new mongoose.Schema({
3    evaluationType: {
4      type: String,
5      enum: [
6        "student-to-instructor",
7        "head-to-instructor",
8        "head-to-other-employee",
9        "team-leader-to-employee",
10       "dean-to-head",
11       "director-to-team-leader",
12       "self",
13       "peer-instructor-to-instructor",
14       "peer-academic-to-academic",
15       "peer-administrative-to-administrative",
16     ],
17     required: true,
18     default: "self",
19   },
20   language: {
21     type: String,
22     enum: ["Amhric", "English"],
23     default: "English",
24   },
25   questions: [
26     {
27       criteria: {
28         type: String,
29         required: true,
30       },
31       category: {
32         type: String,
33       },
34       weight: {
35         type: Number,
36         required: true,
37         min: 1,
38         max: 10,
39       },
40     },
41   ],
42 });
43
44 appraisalTemplateSchema.pre("save", function (next) {
45   const template = this;
46   if (!template.isNew) {
47     return next();
48   }
49   template.questions.forEach((question) => {
50     question._id = new mongoose.Types.ObjectId();
51   });
52   next();
53 });
54
55 const AppraisalTemplate = mongoose.model(
56   "AppraisalTemplate",
57   appraisalTemplateSchema
58 );
59
60 export default AppraisalTemplate;
61

```

Figure 6 7 Appraisal Template Model

```
1 import mongoose from "mongoose";
2 const departmentSchema = new mongoose.Schema({
3   departmentName: {
4     type: String,
5     required: true,
6     trim: true,
7   },
8   departmentCode: {
9     type: String,
10    trim: true,
11    unique: true,
12  },
13  collegeId: {
14    type: mongoose.Schema.Types.ObjectId,
15    ref: "College",
16    required: true,
17  },
18 });
19
20 departmentSchema.index({ departmentCode: 1 }, { unique: true });
21
22 const Department = mongoose.model("Department", departmentSchema);
23
24 export default Department;
25
```

*Figure 6 8 Department Model*

### **6.3. Configuration of the Application Server**

Node.js, a runtime environment built on Chrome's V8 JavaScript engine, and Express, a minimalist web framework for Node.js, are chosen as the application server for several reasons

- A. Flexibility: Node.js allows developers to write server-side code in JavaScript, providing a unified language for both client-side and server-side development. This simplifies development workflows and promotes code reuse.
- B. Performance: Node.js's non-blocking, event-driven architecture enables handling concurrent connections efficiently. This makes Node.js well-suited for I/O-heavy applications, such as web servers, streaming services, and real-time communication platforms.
- C. Non-blocking I/O Model: Node.js's asynchronous, non-blocking I/O model allows handling multiple concurrent operations without blocking the execution thread. This results in improved scalability and responsiveness, particularly for applications with high concurrency requirements.
- D. Middleware Ecosystem: Express's middleware architecture facilitates modular and composable application design. Developers can easily integrate third-party middleware or write custom middleware to handle tasks such as request parsing, authentication, logging, and error handling.

### **6.4 Configuration of Application Security**

#### 6.4.1. Input Validations

- ❖ Implemented comprehensive input validation mechanisms using appropriate libraries and techniques.
- ❖ All user inputs are sanitized and validated to prevent injection attacks and other vulnerabilities.

#### 6.4.2. Encryption/Decryption

- ❖ Passwords are hashed using bcryptjs before storage in the database, ensuring protection against unauthorized access to user credentials.

### **6.4.3. Role-Based Permissions**

- ❖ Clearly defined distinct roles within the application "admin," "hr," "director," "teamLeader," "instructor", "academic," "administrative," "student," "dean ," "head."
- ❖ Documented responsibilities and permissions associated with each role to ensure clarity and consistency.
- ❖ Access control checks are enforced for actions requiring specific privileges, ensuring that users only access functionalities appropriate to their roles.

### **6.4.4. User Access Privileges**

- ❖ Assigned necessary access privileges to user accounts based on their roles.
- ❖ Implemented verification of user credentials during login to retrieve assigned roles and permissions from the database.
- ❖ Access control mechanisms are enforced to restrict unauthorized access to sensitive data and system functionalities.

### **6.4.5. Session Management**

- ❖ Implemented sessions using jsonwebtoken (JWT) for authentication and session tracking.
- ❖ Issued JWT tokens containing user information and relevant permissions upon successful authentication.
- ❖ Stored JWT securely on the client-side using HTTP-only cookies to maintain session state between requests.
- ❖ Validated JWT on subsequent requests to ensure authenticated and authorized access to resources.

#### 6.4.6. Compliance with Non-functional Requirements

- ❖ Reviewed non-functional requirements, including performance, scalability, reliability, and maintainability, to ensure compatibility with security measures.
- ❖ Conducted regular security audits and assessments to identify and mitigate potential vulnerabilities or weaknesses in the application's security architecture.

### 6.7 Testing

This section outlines the comprehensive testing strategy designed to ensure the functionality, reliability, and performance of the Employee Performance Appraisal System for WKU. Through a systematic approach encompassing various testing levels, the goal is to deliver a seamless and effective platform for evaluating and managing employee performance

The project has undergone various testing phases

- ❖ Code Review: The system's source code has been carefully reviewed to identify and resolve architectural and implementation problems.
- ❖ Unit Testing: Each individual module has been thoroughly tested.
- ❖ Integration Testing: Following the testing of individual modules, they were combined and tested to ensure their ability to work together.
- ❖ System Testing: This phase involves checking whether the entire system functions according to requirements.

#### 6.7.1. Test Cases

*Table 6 1 Login Test case (Valid credentials)*

Test ID	Test1
Test Name	Login
Description	This Login functionality of the system tested by providing it with login credentials
Input	Correct Email and Password

Expected outcome	navigate the user to the Dashboard
Actual result	As expected
Pass / Fail	pass

*Table 6 2 Login Test case (Invalid credentials)*

<b>Test ID</b>	<b>Test2</b>
Test Name	Login
Description	This login functionality of the system tested by providing it with login credentials
Input	<ol style="list-style-type: none"> <li>1. Incorrect Email and Incorrect password</li> <li>2. Incorrect Email and correct password</li> <li>3. Correct Email and incorrect password</li> </ol>
Expected outcome	The user is notified that the email or password entered is incorrect and is prompted to re-enter their credentials
Actual result	As expected
Pass / Fail	pass

*Table 6 3 Add User Test Case (Correct Input)*

<b>Test ID</b>	<b>Test3</b>
Test Name	Add User

Description	The process of adding a user to the system is tested by inputting the required user information, including username, email, and department. The system is then evaluated to ensure that the user is successfully created and registered within the system
Input	<ol style="list-style-type: none"> <li>1. Incorrect email format</li> <li>2. Duplicate email address</li> <li>3. Missing email</li> </ol>
Expected outcome	The HR user encounter failure message indicating that the new user was not successfully added to the system
Actual result	As expected
Pass / Fail	pass

*Table 6 Evaluation Test Case (Correct Input)*

<b>Test ID</b>	<b>Test4</b>
Test Name	Review Submission
Description	Evaluator submits a review for an Employee with all required criteria filled
Input	all required criteria filled
Expected outcome	<ol style="list-style-type: none"> <li>1. Review successfully submitted</li> <li>2. The review status is updated to reflect its completion</li> </ol>
Actual result	As expected
Pass / Fail	pass

*Table 6 4 6 Evaluation Test Case (Rating Left)*

<b>Test ID</b>	<b>Test5</b>
Test Name	Review Submission
Description	This test case checks the system's behavior when required fields in the review form are left blank.
Input	Ratings are left blank
Expected outcome	The system prevents submission and displays a clear error message explaining the issue
Actual result	As expected
Pass / Fail	pass

*Table 6 5 Evaluation Test Case (No Active Cycle)*

<b>Test ID</b>	<b>Test6</b>
Test Name	Review Submission
Description	This test case determines the system's behavior when a review is submitted after the designated deadline.
Input	Submit evolution after the end date
Expected outcome	The system promote no active cycle
Actual result	As expected
Pass / Fail	pass

*Table 6 6 Evaluation Test Case (deadline passed)*

<b>Test ID</b>	<b>Test7</b>
Test Name	Review Submission
Description	This test case determines the system's behavior when a review is submitted after the designated deadline.
Input	Submit evolution after the end date
Expected outcome	The system promote no active cycle
Actual result	As expected
Pass / Fail	pass

*Table 6 7 View Result Test Case (without Evaluating All His Peer)*

<b>Test ID</b>	<b>Test8</b>
Test Name	View Result
Description	This test case determines the system's behavior when an employee try to see his result
Input	Without evaluating all his peer
Expected outcome	The system promote Please first review you're peer first
Actual result	As expected
Pass / Fail	pass

*Table 6 8 View Result Test Case*

<b>Test ID</b>	<b>Test9</b>
Test Name	View Result
Description	This test case determines the system's behavior when an employee try to see his result
Input	With evaluating all his peer
Expected outcome	The system allow the user to see his final result
Actual result	As expected
Pass / Fail	pass

# CHAPTER SEVEN

## CONCLUSION AND RECOMMENDATION

### 7.1 Introduction

This chapter summarizes the key findings and accomplishments of the employee performance appraisal system project. It highlights how the project fulfills the objectives outlined earlier and explores potential areas for further development.

### 7.2 Conclusion

The employee performance appraisal system effectively addresses the limitations identified in the current system. These limitations hindered inclusiveness, efficiency, and resource management. The new system offers a standardized, data-driven approach that overcomes these challenges by providing

- A. **Improved Inclusiveness:** The system can now effectively evaluate all roles and employees, including administrative staff, previously excluded by manual processes.
- B. **Streamlined Distribution and Collection:** The system automates the distribution and collection of performance criteria and evaluation results, saving significant time and resources for HR and employees.
- C. **Automated Calculations:** Manual calculations by HR are eliminated. The system automatically calculates evaluations based on predefined criteria, minimizing errors and resource demands. **Reduced Costs:** By transitioning from paper-based methods to a digital system, the new approach minimizes resource consumption and operational costs.
- D. **Elimination of Manual Data Entry:** Manual data entry is no longer required. The system automates the process of recording evaluation results in employee profiles, improving efficiency and accuracy. This comprehensive solution promotes a more inclusive, efficient, and resource-efficient approach to employee performance evaluation.

## 7.3 Recommendations

Building upon the established foundation, several recommendations are offered to further enhance the performance appraisal system:

- ❖ **Recommendation 1:** Develop a mobile application for on-the-go performance reviews and feedback exchange
- ❖ **Recommendation 2:** Integrate artificial intelligence for personalized development suggestions based on performance data
- ❖ **Recommendation 3** Conduct user surveys to gather feedback on system improvement and address identified shortcomings

## References

- [1] <https://medium.com/the-andela-way/system-design-in-software-developmentf360ce6fcbb9> , Last accessed on December 10, 2024.
- [2] <https://www.oracle.com/database/what-is-database.html>
- [3] D. Linda and S. m. Paul, The Use of Object-Oriented Models in Requirements Engineering, 1998.
- [4] S. Jeffery and B. D, Object Oriented System Analysis and Design Method, 5th ed., 2001.
- [5] Rumbaugh, I. Jacobson and G. Booch, the Unified Modeling Language Reference Manual, Singapore: Pearson Education Inc., 2003.
- [6] Unknown Writer, “prutt05\_lec7.pdf”, retrieved from <http://www.nada.kth.s> , last accessed on December 7, 2024.
- [7] Roger S. Pressman, Ph.D., Software Engineering a Practitioner Approach, Fifth Edition 2001.
- [8] Sommerville, I. (2011). Software Engineering (9th ed.). Addison-Wesley.

## APPENDICES

### Appendix I: sample source code

This code snippet defines styled components for a login page in a React application, enhancing the visual presentation and interactivity of the login form, including structured containers, input fields, buttons, and a link for user registration.

```
1 import styled from "styled-components";
2 import LoginForm from "../features/authentication/LoginForm";
3 import Logo from "../ui/Logo";
4 import Heading from "../ui/Heading";
5 import { useState } from "react";
6
7 import FormRowVertical from "../ui/FormRowVertical";
8 import { Link } from "react-router-dom";
9 import { Button } from "react-bootstrap";
10 const LoginLayout = styled.main`
11   min-height: 100vh;
12   display: grid;
13   grid-template-columns: 48rem;
14   align-content: center;
15   justify-content: center;
16   gap: 3.2rem;
17   background-color: var(--color-grey-50);
18 `;
19 const StyledLoginSection = styled.section`
20   background-color: white;
21   padding: 2rem;
22   border-radius: 10px;
23   box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.1);
24 `;
25 const ForgotPasswordLink = styled.a`
26   color: var(--color-primary);
27   text-decoration: underline;
28   margin-top: 1rem;
29   margin-left: 13rem;
30   cursor: pointer;
31 `;
32 const StudentRegister = styled.div`
33   text-align: center;
34   display: block;
35   color: var(--color-brand);
36   text-decoration: none;
37   margin-top: 2rem;
38   font-size: 1.4rem;
39 `;
40 const RememberMeCheckbox = styled.input.attrs({ type: "checkbox" })`
41   margin-top: 1rem;
42 `;
```

This code snippet defines the React component responsible for rendering a login form, providing a user interface for users to input their credentials and log into the application.

```
1  function Login() {
2    const [rememberMe, setRememberMe] = useState(false);
3
4    const handleRememberMeChange = () => {
5      setRememberMe(!rememberMe);
6    };
7    return (
8      <LoginLayout>
9        <StyledLoginSection>
10         <Logo />
11         <Heading as="h4">WKU EPAS</Heading>
12         <LoginForm />
13         <RememberMeCheckbox
14           id="rememberMe"
15           checked={rememberMe}
16           onChange={handleRememberMeChange}
17         />
18         <label htmlFor="rememberMe"> Remember Me</label>
19         <ForgotPasswordLink href="/forgot-password">
20           Forgot Password?
21         </ForgotPasswordLink>
22       </StyledLoginSection>
23     </LoginLayout>
24   );
25 }
26
27 export default Login;
28
```

This code snippet defines a React component responsible for rendering an evaluation form within the application.

```
1  const Tmeval = () => {
2    const [language, setLanguage] = useState("English");
3
4    const handleLanguageChange = (e) => {
5      setLanguage(e.target.value);
6    };
7
8    return (
9      <>
10     <LanguageSelector>
11       <Label htmlFor="languageSelect">Select Language:</Label>
12       <Select
13         id="languageSelect"
14         value={language}
15         onChange={handleLanguageChange}
16       >
17         <option value="English">English</option>
18         <option value="Amhric">Amharic</option>
19       </Select>
20     </LanguageSelector>
21
22     <RatingScale>
23       <h2>RATING SCALE</h2>
24       <div className="rating-bar">
25         <div className="rating-segment veryGood">Very Good (5)</div>
26         <div className="rating-segment good">Good (4)</div>
27         <div className="rating-segment neutral">Neutral (3)</div>
28         <div className="rating-segment bad">Bad (2)</div>
29         <div className="rating-segment very-bad">Very Bad(1)</div>
30       </div>
31     </RatingScale>
32
33     <EvaluateAll language={language} />
34   </>
35 );
```



This code snippet defines the utilization of json web token for session management

```
1 import crypto from "crypto";
2 import { StatusCodes } from "http-status-codes";
3 import jwt from "jsonwebtoken";
4 const createJwt = (payload) => {
5   const token = jwt.sign({ payload }, process.env.JWT_SECRET, {
6     expiresIn: process.env.JWT_EXPIRES_IN * 24 * 60 * 60 * 1000,
7   });
8
9   return token;
10 };
11 const verifyJwt = async (token) => {
12   const decoded = await jwt.verify(token, process.env.JWT_SECRET);
13   return decoded;
14 };
15 const sendToken = (user, req, res) => {
16   const token = createJwt(user._id);
17   const tokenExpire =
18     Date.now() + process.env.JWT_EXPIRES_IN * 24 * 60 * 60 * 1000;
19
20   user.password = undefined;
21   return res.status(StatusCodes.OK).json({
22     status: "sucess",
23     token,
24     data: user,
25   });
26 };
27
28 export { verifyJwt, sendToken, createJwt };
29
```

## Appendix II: sample Test Script

This is unit test case written in Mocha to make sure evaluation process work as expected

```
1 import { assert } from "chai";
2 import sinon from "sinon";
3 import { getFinalResult } from "../src/controllers/resultController.js";
4
5 import FinalResult from "../src/models/resultDetail.js";
6 import getActiveCycle from "../src/utils/review/getActiveCycle.js";
7 import Course from "../src/models/courseModel.js";
8
9 describe("getFinalResult", function () {
10   afterEach(function () {
11     sinon.restore();
12   });
13
14   it("should return final result with weights for academic role", async function () {
15     const req = { user: { _id: "662bfa22149fe3ae70d4331b", role: "head" } };
16     const res = {
17       status: sinon.stub().returnsThis(),
18       json: sinon.stub(),
19     };
20     const next = sinon.stub();
21     const testCycle = { _id: "66221bc933be47f7293a7e85" };
22     const testResult = {
23       status: "completed",
24       byPeer: { total: 5 },
25       byDean: 3,
26       byHead: 4,
27       byStudent: { total: 6 },
28       byStdRank: 2,
29       byPeerRank: 1,
30       byDeanRank: 3,
31       byHeadRank: 2,
32     };

```

```

1  sinon.stub(getActiveCycle.prototype, "getActiveCycle").resolves("");
2  sinon.stub(FinalResult, "findOne").resolves(testResult);
3  sinon.stub(FinalResult.prototype, "calculateRanks").resolves();
4  sinon.stub(Course, "find").resolves([
5    [
6      {
7        _id: "662bfa22149fe3ae70d4331b",
8        name: "Introduction to Programming",
9        code: "CS101",
10       semester: 2,
11       batch: 2019,
12       department: "660959dc4482a6bd00331685",
13       section: 1,
14       startDate: "1970-01-01T00:00:45.412Z",
15       endDate: "2024-05-08T00:00:00.000Z",
16       instructor: {
17         _id: "66164c3f242ab909809173d3",
18         firstName: "Hirut",
19         lastName: "Jenbere",
20         salutation: "Mrs.",
21         dateOfJoining: "2024-04-18T00:00:00.000Z",
22         experience: 2,
23         designation: "plant",
24         branch: "Main",
25         email: "hirut@gmail.com",
26         gender: "Female",
27         phone: 995562404,
28         age: 21,
29         companyEmail: "wku@gmail.com",
30         department: "660959dc4482a6bd00331685",
31         college: "66093fb4dc86fc4380d73abe",
32         __v: 0,
33         role: "instructor",
34       },
35     ],
36   ],
37   ]);
38

```

## **Appendix III: Interview question prepared**

### **1. ICT Director**

#### **Challenges in the Current System**

- ❖ What challenges have been encountered in the current appraisal system from an ICT perspective?
- ❖ Can you highlight any specific issues or difficulties faced in the existing system?

#### **Desired Features in the New System**

- ❖ In envisioning a new and improved appraisal system, what features or functionalities would you like to see incorporated?
- ❖ Are there specific enhancements or capabilities that you believe would address the current system's shortcomings?

#### **Positive Aspects of the Current System**

- ❖ Despite any challenges, what are the positive or commendable aspects of the current appraisal system?
- ❖ Are there elements that are particularly effective or well-received by users?

### **2. HR**

#### **Overview of the Current Appraisal Process**

- ❖ Can you provide a detailed overview of the current appraisal process followed at the university?
- ❖ What key steps and components are involved in this process?

#### **HR's Role in the Appraisal Process**

- ❖ What role does the HR department play in facilitating and managing the appraisal process?
- ❖ How does HR contribute to ensuring a fair and effective evaluation?

### **Organizational Structure Inquiry**

- ❖ Could you describe the organizational structure of the university, especially as it relates to the appraisal and evaluation process?
- ❖ How are different departments and roles involved in the overall structure?

### **Frequency and Initiation of Evaluation**

- ❖ When does the evaluation process typically commence, and how many times is it conducted in a year?
- ❖ Are there specific triggers or events that initiate the appraisal cycle?

### **Appraisal Result Storage and Access**

- ❖ How are the results of the appraisal stored, and who has access to these results?
- ❖ Have there been any challenges or difficulties encountered in storing or retrieving appraisal data?

### **Business Rule Clarification**

- ❖ Can you provide insights into the business rules governing the appraisal process?
- ❖ How are these rules defined and implemented to ensure consistency and fairness?

## **3. Quality Coordinator**

### **Participation in the Evaluation Process**

- ❖ Who actively participates in the employee evaluation process, particularly from the quality coordination perspective?
- ❖ Are there specific roles or individuals responsible for ensuring the quality of the evaluation?

### **Forms Utilized for Assessment**

- ❖ What forms are used to assess the employee evaluation process, and how are they designed?
- ❖ Are there specific criteria or metrics included in these assessment forms?

**Calculation of Final Results:**

- ❖ How is the final result of the evaluation calculated, and what factors contribute to this calculation?

**Frequency of Criteria and Form Changes:**

- ❖ How often do the criteria and assessment forms undergo changes or updates?
- ❖ Who holds the responsibility for initiating and preparing these changes?

**4. Head of Software Department**

**Challenges in the Evaluation Process:**

- ❖ What challenges have you faced in performing the evaluation process within the software department?
- ❖ Are there specific obstacles or difficulties unique to your department's context?

**Tasks Related to the Evaluation Process:**

- ❖ Could you outline the key tasks and responsibilities that the head of the software department is involved in concerning the evaluation process?
- ❖ How do these responsibilities align with the overall goals of the department?



የገቢዎች ሪፖርት  
 2015-16  
 የወ/ክ/የ

የገቢዎች ሪፖርት ለገቢዎች ሪፖርት 2015 ዓ.ም. የገቢዎች ሪፖርት ለገቢዎች ሪፖርት

የገቢዎች ሪፖርት ለገቢዎች ሪፖርት

የገቢዎች ሪፖርት ለገቢዎች ሪፖርት

የገቢዎች ሪፖርት ለገቢዎች ሪፖርት

የገቢዎች ሪፖርት ለገቢዎች ሪፖርት 4.05 እሴት 4.05

ተ.ቁ	የገቢዎች ሪፖርት	የገቢዎች ሪፖርት ለገቢዎች ሪፖርት	ገቢዎች ሪፖርት (የገቢዎች ሪፖርት)
1	በገቢዎች ሪፖርት ለገቢዎች ሪፖርት የገቢዎች ሪፖርት ለገቢዎች ሪፖርት የገቢዎች ሪፖርት ለገቢዎች ሪፖርት	50	
2	በገቢዎች ሪፖርት ለገቢዎች ሪፖርት ገቢዎች ሪፖርት	5	
3	በገቢዎች ሪፖርት ለገቢዎች ሪፖርት በገቢዎች ሪፖርት ለገቢዎች ሪፖርት	5	
4	በገቢዎች ሪፖርት ለገቢዎች ሪፖርት በገቢዎች ሪፖርት ለገቢዎች ሪፖርት	5	
5	በገቢዎች ሪፖርት ለገቢዎች ሪፖርት በገቢዎች ሪፖርት ለገቢዎች ሪፖርት	5	
6	በገቢዎች ሪፖርት ለገቢዎች ሪፖርት በገቢዎች ሪፖርት ለገቢዎች ሪፖርት	5	

የገቢዎች ሪፖርት ለገቢዎች ሪፖርት

ገቢዎች ሪፖርት

ገቢዎች ሪፖርት

# WOLKITE UNIVERSITY

## INSTRUCTORS PERFORMANCE EVALUATION QUESTIONNAIRE

No	A. Evaluation Criteria	Scale					
		V L	L	A	H	V H	N A
<b>Core Competency</b>							
1	Efforts of self development in his/her specialization	1	2	3	4	5	N A
2	Adequacy of subject matter knowledge	1	2	3	4	5	N A
3	Willingness to accept additional teaching assignments when compelling situation arises in the department	1	2	3	4	5	N A
4	Willingness to accept related assignments other than regular teaching in the department (mentorship...)	1	2	3	4	5	N A
5	Effectiveness as a mentor in educational development army, internship etc..	1	2	3	4	5	N A
6	Active participation in improvement of teaching-learning process	1	2	3	4	5	N A
7	Participation in community service affairs	1	2	3	4	5	N A
8	Participating actively in seminars/workshop/symposia	1	2	3	4	5	N A
9	Identifying priority areas in one's discipline and pursuing research in that area	1	2	3	4	5	N A
10	Participation in research project and project proposal development	1	2	3	4	5	N A
11	Performance as an academic advisor	1	2	3	4	5	N A



1 2	Participation in problem identification solving and at department/college/---institution	1	2	3	4	5	N A
1 3	Continuous assessment implementation	1	2	3	4	5	N A
1 4	Participating actively in departmental/Faculty/institution affairs	1	2	3	4	5	N A
1 5	Providing and reporting tutorial activities designed for the students	1	2	3	4	5	N A
1 6	Participation in preparation and reviewing of teaching materials	1	2	3	4	5	N A
7	Updating teaching materials	1	2	3	4	5	N A
8	Willingness to share University resources with other colleagues						

### Time Management

1 9	Executing assigned classes/invigilation on time.	1	2	3	4	5	N A
2 0	Notifying and implementing consultation timely	1	2	3	4	5	N A
2 1	Giving timely feedback to students	1	2	3	4	5	N A
2 2	Meeting deadlines (in reporting, SIMS result feeding, submission of grade/ documents..etc...)	1	2	3	4	5	N A

### Ethical Competence

2 2 4 3	Showing concern for the use of resources of the department and the University	1	2	3	4	5	N
	Having positive attitude to work with others	1	2	3	4	5	N A A
	Willingness and participation in committee works at department /University level	1	2	3	4	5	N A
2 6	Willingness to take assignments outside the University in his area of specialization	1	2	3	4	5	N A
2 7	His/her professional ethics (dressing, hair style, personality...)	1	2	3	4	5	N A

**General comments about the**

**instructor** Strength of the instructor? -

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Suggested points/aspects the instructor should improve.

No	Evaluation Criteria	Scale					
		VL	L	A	H	VH	NA
<b>Core Competency: Subject matter</b>							
1	Contribution in preparing and searching for teaching materials	1	2	3	4	5	NA
2	Continuous update of the subject matter	1	2	3	4	5	NA
3	Attending seminars organized by the department/faculty/ institution	1	2	3	4	5	NA

4	Level of his/her subject matter knowledge and skill	1	2	3	4	5	NA
<b>Core Competency: Research and Community Services</b>							
5	Willingness and level of engagement in community service activities	1	2	3	4	5	NA