



DAFFODIL INTERNATIONAL UNIVERSITY

PROJECT

ON

Appraisal Service System

BY

AEID SAROWAR

ID: 201-16-519

SUPERVISED BY

Dr. Md. Nadir Bin Ali

Associate Professor

Department of Computing and Information System (CIS)

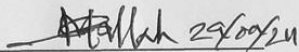
Daffodil International University

29th September 2024

APPROVAL

This Project titled “**Appraisal Service System**”, Submitted by Aeid Sarowar, ID No: **201-16-519** to the Department of Computing & Information System, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information System and approved as to its style and contents. The presentation has been held on **29-09-2024**.

BOARD OF EXAMINERS



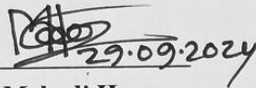
Md Sarwar Hossain Mollah
Associate Professor and Head
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Chairman



Md. Nasimul Kader
Assistant Professor
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Md. Mehedi Hassan
Lecturer
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Md. Miftah Uddin
Head of SBU, Brain Station 23

External Examiner

Declaration

I hereby declare that; this project has been done by me under supervision of **Dr. Md. Nadir Bin Ali, Associate Professor**, of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

Supervised By

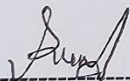


Dr. Md. Nadir Bin Ali

Associate Professor

Daffodil International University

Submitted By



Name: Acid Sarowar

ID: 201-16-519

Department of CIS

Daffodil International University

Acknowledgment

I want to begin by expressing my profound gratitude to Almighty Allah for this incredible opportunity. Without His guidance, completing this project would have been impossible. I am also thankful to Daffodil Computers Limited (DCL) for selecting me as an intern trainee, as this program has imparted invaluable lessons.

Teachers are often seen as second parents due to their pivotal role in guiding students toward success. During my internship, I was fortunate to have the support of two remarkable mentors. Md. Mazharul Islam, my intern trainer and Technical Leader, and Dr. Md. Nadir Bin Ali, Associate Professor, who supervised my academic work, consistently provided me with wise advice and assistance in overcoming challenging tasks. Their unwavering support alleviated my stress and fortified my resolve.

I am deeply appreciative of their kindness, readiness to help, and belief in my potential. Their mentorship and instruction have equipped me with the skills to navigate future challenges and manage critical projects effectively. I am grateful for their guidance and the valuable lessons they have imparted.

Dedication

This project stands as a tribute to Almighty Allah and to the unwavering support of my beloved Mom, Dad, and elder sister, whose guidance and encouragement were instrumental in my successful completion of the B.S.c program. I attribute all glory and honor to Almighty God for His steadfast support, protection, and guidance throughout this journey. His blessings have sustained me, giving me the strength to overcome every challenge.

I am deeply indebted to my family for their selfless sacrifices, unconditional love, and unwavering belief in my potential. My mom's boundless encouragement, my dad's invaluable wisdom, and my sister's unwavering support have been my pillars of strength. They have celebrated my successes joyfully and supported me through my toughest moments. This achievement is a testament to their unwavering dedication and belief in my abilities, making it as much their triumph as mine.

Abstract

The **Appraisal Service** is an integral component of our HR Management System, designed to streamline and enhance the performance appraisal process within our organization. This web-based application automates appraisal workflows, ensuring efficient and comprehensive evaluation of employee performance. Implementing the Scrum framework within the Agile Model allows continuous improvement and adaptation based on iterative feedback. The service offers significant benefits, including reduced administrative burdens, cost savings, enhanced data security, and optimized time management for both managers and employees. Its key features encompass automated appraisal processing, robust data management, secure operations, an intuitive user interface, and advanced reporting and analytics tools. These elements collectively ensure a consistent, fair, and productive appraisal process. As a strategic asset, the Appraisal Service enables efficient performance evaluations, fosters fairness, and enhances organizational productivity. Its web-based nature ensures accessibility and ease of use across various devices and locations, promoting organizational agility. By supporting a culture of continuous improvement and aligning with our commitment to technological advancement, the Appraisal Service exemplifies our dedication to innovation and excellence in human resource management.

Table of Contents

List of Tables:	xi
Chapter 1.....	1
Introduction.....	1
1.1 The System Developed	1
1.2 Feasibility Analysis.....	1
1.3 Technical Feasibility	1
1.4 Operational Feasibility.....	2
1.5 Economic Feasibility	2
1.6 Environmental Feasibility	2
1.7 Contribution	2
Chapter 2.....	3
Initial Study.....	3
2.1 Project Proposal	3
2.2 Background of the Appraisal Service System.....	3
2.3 Problem Area	4
2.4 Possible Solution.....	4
2.5 Objectives	4
Chapter 3	5
Literature Review	5
Chapter 4	8
Methodology	8
Dynamic systems development model (DSDM).....	12
Utilizing the DSDM Approach for the Appraisal Service	13
4.3 Sections of methodology.....	13
4.4 Implementation Plans	14
Chapter 5.....	15
Planning	15
5.1 Project Plan	15
5.1.1 Management Plan / Work Breakdown Structure (WBS):	16
5.1.2 Resource Allocation (H/W, S/W. Models, Documentation):.....	16
5.1.3 Time Duration / Time Boxing:.....	18
5.1.4 Gantt chart:.....	19

Figure 5.4: Gantt chart	20
Chapter 6.....	20
Feasibility.....	20
6.1 All Possible Types of Feasibility	20
6.1.1 Operational Feasibility	20
6.1.2 Technical feasibility	21
Figure 6.1: Technical Requirements	21
Figure 6.1: Technical Feasibility	22
6.2 Cost Benefit Analysis	22
6.3 DSDM – good or not for this project -PAQ.....	23
Chapter 7.....	24
Foundation	24
7.1 The Problem Area Identification.....	24
7.1.1 Interviews.....	24
7.1.2 Observation	24
7.1.3 Questionnaires.....	25
7.2 Rich Picture.....	25
7.3 Simple Data Flow	26
7.4 Specific Problem Area Identification and Description	26
7.5 Possible Solution.....	27
7.6 Overall Requirement List.....	27
7.6.3 Technology Implementation	28
7.6.4 Recommendation and Justifications:	28
Chapter 8.....	30
Feasibility.....	30
8.1 Old System Use Case:.....	31
8.2 Activity Diagram:	32
8.3 Full System Use Case Diagram:	33
8.4 Full System Activity Diagram:	34
8.5 Requirements Catalogue:	35
8.6 Requirement Prioritized:	37
Chapter 9.....	39
Feasibility.....	39

9.2 Use Case of Appraisal Service System:	41
9.3 Class Diagram of Appraisal Service System:	42
9.4 ERD Diagram of IUMS Admission Service System:	43
9.5 Sequence Diagram of Appraisal Service System:	44
9.6 Component Diagram of Appraisal Service System:	45
9.7 Deployment Diagram of IUMS Admission Service System:	45
Chapter 10.....	46
Deployment / Development	46
10.1 Core Module Coding Samples and Outputs:.....	46
Outputs:.....	50
10.2 Possible problem breakdown:	52
10.3 Prioritization while Developing:	52
Chapter 11.....	53
Testing	53
11.1 Old System Use Case:.....	54
11.2 Primary Testing Overview: Appraisal Service System	55
11.2 Test Case:.....	57
11.3 Unit Testing:	57
11.4 Module Testing:	58
11.5 Integration Testing:	59
11.6 Acceptance Testing:.....	60
11.7 Security Testing:	60
Summary of Testing Efforts	61
Chapter 12.....	63
Implementation	63
12.1 Training.....	63
12.2 Big Bang	64
Chapter 13.....	65
Critical Appraisal and Evaluation	65
Chapter 14.....	67
Lessons Learned.....	67
14.1 Pre project – review – closing.....	67
14.2 What have I learned	67

14.3 What problem I have faced	68
14.4 What solutions occurred.....	68
Chapter 15.....	69
Conclusion	69
Appendix:.....	71
Reference	72
Plagiarism report.....	73

List of figure:

Figure 1: Waterfall Methodology	10
Figure 2: V Methodology.....	11
Figure 3: DSDM Methodology	12
Figure 4: Gantt chart	20
Figure 5: Rich picture	25
Figure 6: Simple Data Flow	26
Figure 7: Old System Use Case	31
Figure 8: Activity Diagram.....	32
Figure 9: Full System Use Case Diagram.....	33
Figure 10: Full System Activity Diagram.....	34

Figure 11: Use Case	41
Figure 12: Class Diagram	42
Figure 13: ERD Diagram	43
Figure 14: Sequence Diagram.....	44
Figure 15: Component Diagram	45
Figure 16: Deployment Diagram	45
Figure 17: Admission Center Service	47
Figure 18: Admit Card Signature Service.....	48
Figure 19: Applicant Assigned to center Service.....	49
Figure 20: Payment	49
Figure 21: Appraisal Service System.....	54

List of Tables:

Table 1: Breakdown Structure of DSMS	16
Table 2: Resource Allocation List	17
Table 3: List of the Time Boxes.....	18
Table 4: Technical Feasibility	Error! Bookmark not defined.
Table 5: Cost Benefits.....	22
Table 6: Requirements Catalogue	37
Table 7: Requirement Prioritized.....	38
Table 8: Prioritization Requirement.....	52
Table 9: Test Case.....	57
Table 10: Unit Test Case 1.....	57
Table 11: Unit Test Case 2.....	58
Table 12: Module Test Case 1	58
Table 13: Module Test Case 2	Error! Bookmark not defined.
Table 14: Integration Test Case 1	59
Table 15: Integration Test Case 2	59
Table 16: Acceptance Test.....	60
Table 17: Security Test 1	60
Table 18: Security Test 2	Error! Bookmark not defined.
Table 19: Final Test Report	62
Table 20: Training Based.....	64

Chapter 1

Introduction

The **Appraisal Service** stands at the core of our HR Management System, dedicated to revolutionizing how we conduct performance evaluations. This cutting-edge web-based application automates and streamlines the appraisal process, ensuring thorough assessments of employee performance with precision and efficiency. Embracing the Scrum framework within the Agile Model, the system evolves continuously through iterative enhancements and user feedback. Designed to address the challenges of traditional appraisal methods, the Appraisal Service offers a comprehensive suite of features. These include automated processing, robust data management capabilities, secure operations, an intuitive user interface, and advanced analytics tools. By significantly reducing administrative overhead and optimizing time management for managers and employees alike, our system fosters a fair and consistent appraisal experience. Beyond efficiency, the Appraisal Service enhances organizational agility by providing seamless accessibility across various devices and locations. This accessibility not only improves user experience but also underscores our commitment to leveraging technology for superior human resource management. As a strategic asset, the Appraisal Service empowers our organization to uphold fairness, drive productivity, and cultivate a culture of continuous improvement in performance evaluation practices.

1.1 The System Developed

In this project, I have developed the "Appraisal Service" as an integral component of our HR Management System. This web-based application is meticulously designed to streamline the performance appraisal process within our organization. It automates and enriches our appraisal workflows, ensuring a thorough and efficient evaluation of employee performance. By embracing agile principles, particularly within the Scrum framework, we continuously enhance and adapt the system based on ongoing feedback and iterative improvements.

1.2 Feasibility Analysis

Conducting a feasibility analysis is critical to assessing the practicality and benefits of implementing the "Appraisal Service" within our organization.

1.3 Technical Feasibility

Hardware and Software Requirements: Evaluate the infrastructure to support the application, ensuring compatibility with existing systems.

Technology Compatibility: Ensure seamless integration with current HR and management tools.

1.4 Operational Feasibility

User Training: Assess the system's ease of use and complexity, determining training needs for administrators and users.

Integration with Existing Processes: Evaluate how well the "Appraisal Service" integrates with current appraisal procedures and organizational workflows.

1.5 Economic Feasibility

Cost-Benefit Analysis: Conduct a detailed assessment of costs versus benefits to justify the investment in the system.

Return on Investment (ROI): Project expected returns and benefits over time, considering improvements in productivity and efficiency.

1.6 Environmental Feasibility

Sustainability: Consider environmental impacts, such as energy consumption and resource utilization.

Scalability: Assess whether the system can scale to accommodate future growth and changes in organizational needs.

1.7 Contribution

The "Appraisal Service" makes a significant contribution to our organization across various dimensions:

Operational Efficiency: Automates manual processes, reduces administrative burdens, and accelerates the appraisal cycle, leading to faster decision-making and improved employee management.

User-Centric Approach: Enhances user experience with an intuitive interface and personalized access to appraisal-related information.

Data Security and Compliance: Implements robust security so that it can secure data of employee. It also ensures that all required regulations are followed.

Strategic Asset: Acts as a strategic tool to uphold fairness, drive productivity, and align employee performance with organizational goals.

Chapter 2

Initial Study

Introduction

The initial study of the **Appraisal Service** marks a pivotal phase in our journey towards transforming performance appraisal practices within our organization. This chapter provides a comprehensive overview of the project proposal, background context, identified problem areas, proposed solutions, and strategic objectives that define the foundation of our Appraisal Service system. Our commitment to enhancing performance evaluations stems from the recognition of inherent challenges in our current appraisal framework. Manual processes, disparate systems, and limited technological integration have posed obstacles to efficiency, transparency, and effective decision-making. The Appraisal Service emerges as a strategic initiative to address these challenges by centralizing appraisal processes, enhancing communication, and providing robust tools for seamless management and evaluation. This chapter delves into the objectives guiding the development of the Appraisal Service, emphasizing automation, user experience enhancement, data security, comprehensive reporting, and scalability. By aligning these objectives with organizational goals, we aim to foster a culture of continuous improvement and ensure the Appraisal Service becomes a cornerstone of operational excellence and employee development.

2.1 Project Proposal

The Appraisal Service is designed to revolutionize our performance appraisal processes, providing a centralized and efficient platform for evaluating employee performance. This system will streamline the appraisal process, enhance communication, and offer robust tools for managing appraisals seamlessly. The proposed Appraisal Service is crucial for ushering in a new era of efficiency and transparency in our organizational appraisal procedures.

2.2 Background of the Appraisal Service System

Our organization is dedicated to optimizing employee performance through structured appraisal practices. With diverse roles and responsibilities across various departments, managing performance evaluations manually has led to inefficiencies, delays, and challenges in maintaining transparency. The Appraisal Service system caters to two primary user roles:

HR Administrators: Responsible for system administration, configuration, user management, and overall control of the Appraisal Service.

Managers: Involved in conducting appraisals, reviewing performance metrics, and making informed decisions related to employee development and rewards.

2.3 Problem Area

The Appraisal Service addresses inherent challenges in our current performance appraisal framework. Manual processes, disparate systems, and limited technological integration have hindered effective performance evaluation. These issues necessitated the development of the Appraisal Service to streamline and enhance our appraisal workflows.

2.4 Possible Solution

Transformation of the Appraisal Service through an Integrated and Automated Module. Automated Appraisal Processes, Centralized Communication Hub, Enhanced Data Security, Comprehensive User Access Control, Improved Reporting and Analytics, Integrated Email Communication, and Continuous Monitoring and Updates. By addressing these aspects, the Appraisal Service aims to transform our performance evaluation practices into a modern, cohesive, and employee-centric platform that enhances organizational effectiveness and employee satisfaction.

2.5 Objectives

The Appraisal Service System is integral to our broader HR Management System project, designed to optimize organizational efficiency and enhance employee development. By automating and standardizing performance evaluations, our system aims to streamline processes, ensure fairness, and foster continuous improvement. Through structured feedback and data-driven insights, we empower managers to make informed decisions that align talent management strategies with organizational goals, promoting a culture of transparency and growth within our workforce. The objectives of the Appraisal Service align with our organizational goals for improving performance evaluation processes:

- **To Streamline Appraisal Processes:** Automate and integrate appraisal procedures to reduce manual effort and streamline workflows for employees, managers, and HR/admin personnel.
- **To Ensure Comprehensive Performance Evaluation:** Implement a standardized framework for performance assessment that includes clear metrics and criteria, ensuring fair and consistent evaluations across all departments.

- **To Facilitate Continuous Feedback and Development:** Enable regular feedback loops and development discussions between employees and managers through the appraisal system, promoting continuous learning and growth.
- **To Enhance Data-Driven Decision Making:** Utilize performance data collected from appraisals to support strategic decision making related to talent management, succession planning, and workforce development initiatives.

Chapter 3

Literature Review

In Bangladesh, the landscape of human resource management is evolving, yet the utilization of modern software solutions for performance appraisal services remains limited. This literature review aims to explore the current state of appraisal service coordination in organizations in Bangladesh, emphasizing the prevalence of traditional management systems and the need for the adoption of software solutions.

Historically, organizations in Bangladesh have relied on manual and paper-based appraisal management systems. Studies highlight the challenges posed by these traditional systems, including inefficiencies, data inaccuracies, and prolonged processing times.

Despite the global trend toward digital transformation in management practices, a significant portion of organizations in Bangladesh continues to lag in adopting software solutions for appraisal services "Islam, M. (2015). The landscape of human resource management and appraisal services in Bangladesh: A literature review."). Research reveals that only a small fraction of institutions have embraced modern technologies, leaving the majority to grapple with outdated processes.

This literature review emphasizes the urgent need for organizations in Bangladesh to transition from traditional, paper-based appraisal management systems to modern, software-driven solutions. The findings underscore the potential benefits in terms of efficiency, accuracy, and overall improvement in the quality of appraisal services.

(Rahman, S. A., Taghizadeh, S. K., Ramayah, T., & Alam, M. M. D. (2017). Technology acceptance among micro-entrepreneurs in marginalized social strata: The case of social innovation in Bangladesh. *Journal Name*, 118, 236-245)

3.1 Discussion on Problem Domain Based on Published Articles

Complexity and Usability Challenges: Published articles consistently emphasize the inherent complexity and poor usability of existing appraisal service systems. The interfaces are often convoluted and non-intuitive, leading to difficulties for both administrators and employees. Addressing the usability challenges requires a user-centric design approach. Insights from research recommend redesigning interfaces to simplify navigation, reduce cognitive load, and improve the overall user experience.

Document Management and Data Handling Issues: Several articles highlight the inadequacies in document tracking, slow retrieval mechanisms, and data redundancy within appraisal service systems. These issues result in delays, inaccuracies, and inconsistencies in the appraisal process. Research underscores the critical role of efficient document management in the appraisal process. Inefficient systems hinder quick access to relevant documents, leading to bottlenecks in decision-making and potential errors.

The insights derived from published articles collectively underscore the multifaceted challenges within the problem domain of inefficient appraisal service systems. Usability issues, document management challenges, training deficiencies, and integration hurdles all contribute to a suboptimal appraisal process. Implementing solutions aligned with the recommendations from research can pave the way for more streamlined, efficient, and user-friendly appraisal service systems in organizations.

3.2 Discussion on Problem Solutions Based on Published Articles

Usability Improvement through User-Centric Design: Research consistently emphasizes the need for a user-centric design approach to improve the usability of appraisal service systems. Simplifying interfaces, enhancing navigation, and incorporating user feedback are critical elements of this approach.

Solutions:

- Test the system to identify and improve the problems.
- Iteratively redesign interfaces based on user feedback.
- Implement intuitive navigation and provide contextual help features.
- Prioritize accessibility to cater to diverse user needs.

Advanced Document Management Systems: Efficient document management is crucial for the smooth operation of appraisal service systems. Published articles recommend the implementation of advanced document management systems with features tailored for the appraisal process.

Solutions:

- Integrate document tracking mechanisms for real-time monitoring.
- Implement quick retrieval mechanisms for efficient access to documents.
- Ensure data accuracy and consistency through version control.
- Explore cloud-based solutions to facilitate seamless document sharing.

3.3 Comparison of Three/Four Leading Solutions

Usability:

- **Solution A:** User-friendly interface, intuitive navigation.
- **Solution B:** Streamlined user experience, clear workflows.
- **Solution C:** Intuitive design, accessible features.

Document Management:

- **Solution A:** Advanced document tracking, quick retrieval mechanisms.
- **Solution B:** Robust document management, version control.
- **Solution C:** Efficient document sharing, cloud-based solutions.

Training and Support:

- **Solution A:** Comprehensive training programs, user manuals.
- **Solution B:** Hands-on training, continuous learning initiatives.
- **Solution C:** Ongoing user support, refresher courses.

3.3.1 Best Features

- User-Friendly Interface
- Application Management
- Reporting and Analytics
- Customization Options
- Automation of Processes
- Security Measures
- Financial Management

3.3.2 Limitations

- Technical Challenges
- Data Security and Privacy

- Dependency on Internet Connectivity
- Maintenance Challenges
- Incomplete Data

Chapter 4

Methodology

Introduction

Effective implementation of the "Appraisal Service" project within the HR Management System necessitates a systematic approach to software development. For this project, the Dynamic Systems Development Model (DSDM) has been selected as the methodology of choice. DSDM, a variant of the Agile framework, is known for its flexibility and iterative approach, making it suitable for adapting to evolving stakeholder needs and project dynamics. This chapter discusses the rationale behind selecting DSDM over other methodologies, outlines its phases and benefits, and details the structured approach taken to conceptualize, design, implement, and deploy the Appraisal Service system.

4.1 What to Use

This chapter describes the approach taken to accomplish the objectives of the "Appraisal Service" project within the HR Management System. SDLC is a process for building and maintaining software. It provides an overview of the steps, stages, objectives. Different frameworks are employed in this software development, including agile, waterfall, V-, spiral, and dynamic systems development model (DSDM) software development models. For this project, we have chosen to implement the Agile Model, specifically the Dynamic Systems Development Model (DSDM), to ensure a flexible and adaptive approach to development.

4.2 Why to Use

DSDM is more focused than most other agile approaches since it addresses projects rather than just the creation and delivery of products, which are usually software. The project context must take into account the overall business demand as well as every element of the solution that emerges to address that need. With a long history of implementing Agile projects successfully in a variety of business contexts, DSDM has proven to be entirely scalable, operating effectively in large, complex organizations, small, simple businesses, and highly regulated environments.

Additionally, it has been shown to work just as well for non-IT projects like business change initiatives as for IT ones.

I'll outline a few situations in which this project will benefit from this procedure:

- **Focus on Business Needs:** DSDM makes sure the project fits the overall business goals and covers all parts of the solution needed to fulfill these goals.
- **Iterative Development:** Continuous feedback and incremental improvements help refine the "Appraisal Service" to better meet user requirements.
- **Scalability:** DSDM's scalable nature makes it suitable for both large, complex environments and smaller, simpler projects.
- **Flexibility:** The ability to adapt to changes in requirements or project scope ensures that the final product remains relevant and effective.

Now, let's discuss the Waterfall Model and its applicability to the project, highlighting its benefits and drawbacks.

Waterfall Model:

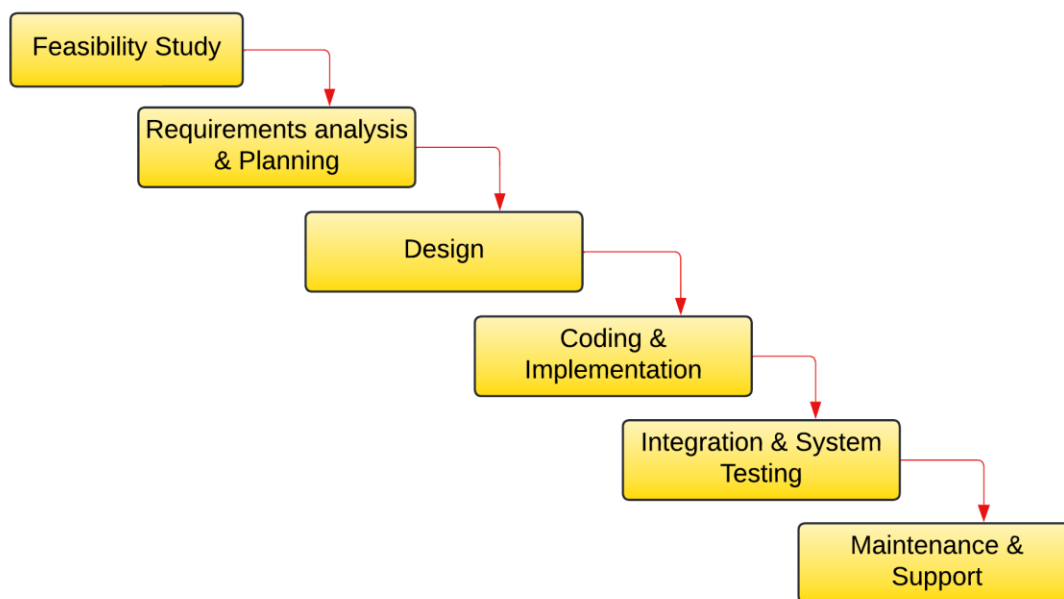


Figure 1: Waterfall Methodology

The Waterfall Model is a step-by-step or linear software development model. A waterfall model requires well-defined exit conditions, such as getting approval from every project member, and

each stage needs to be completed in the correct order. A set of tasks, accompanying documentation, and exit requirements for every step of each phase are all part of the Waterfall Model.

When utilizing this model:

- **Requirements are not frequently changing:** If the project has stable requirements that are unlikely to change, the Waterfall Model might be suitable.
- **The application is not intricate or drawn out:** For simpler projects with straightforward objectives, this model can be effective.
- **The undertaking is short:** Short-term projects with clear, predefined goals can benefit from the Waterfall Model.
- **The prerequisite is unchanging:** When requirements are well-understood and fixed, this model can provide a structured approach.
- **The surroundings remain stable:** Projects in stable environments where external factors are not likely to affect development can utilize this model.

Advantages of the Waterfall Model

- **Ideal for smaller projects with well-defined objectives:** The model works best when project requirements are clear and stable.
- **Sequential development:** Each step of development must be completed before going on to the next.
- **Quality assurance testing:** Verification and validation are done prior to concluding each step.
- **Comprehensive documentation:** Required at every phase of the software development cycle.
- **Low client engagement:** The project team is mostly responsible for its success; client involvement is minimal.
- **Controlled changes:** Any changes made to the software are done during the development phase.

Drawbacks of the Waterfall Model

- **Error fixing constraints:** Errors may only be fixed within the current phase.
- **Inflexibility for complex projects:** It is not suitable for projects where needs frequently change.
- **Late testing stage:** Testing occurs at a late stage in the development process.
- **Limited customer feedback:** It is impossible to incorporate customer feedback continuously; the focus is on documentation for most of the development process.

V Model

The V model represents the verification and validation phases of the SDLC. A V-shaped model's life cycle is carried out in sequential order. Before moving on to the next phase in the V model for software testing, each step must be completed. It is a methodical approach to software development and problem-solving compared to the waterfall technique.

When utilizing this model:

- Efficient when the project's needs are definite and consistent.
- Advantageous when documentation is given first priority.
- Preventing early-stage problems.

V Model Diagram:

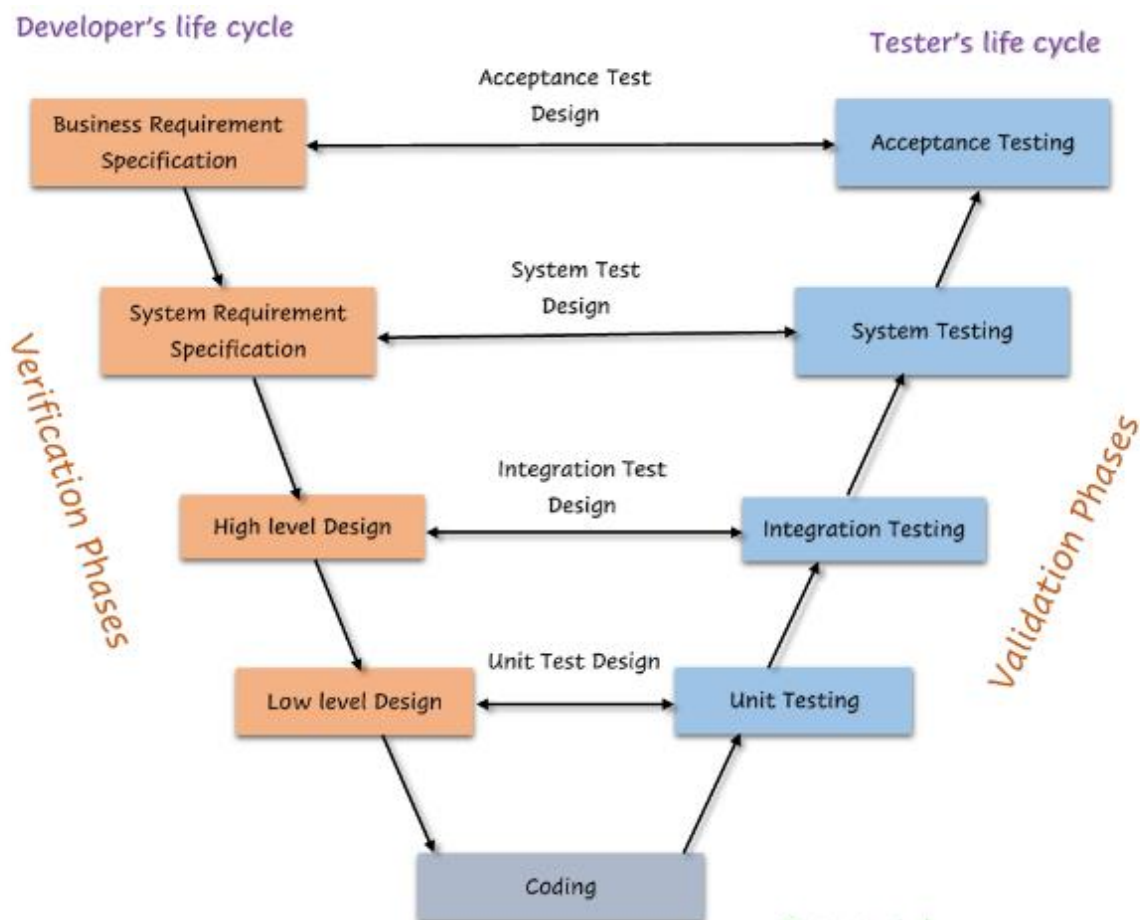


Figure 2: V Methodology (Adapted from <https://selenium.blog/sdlc-v-model/>)

Advantages of the V Model

- **Definite and consistent requirements:** Efficient when project needs are stable.
- **Prioritizes documentation:** Advantageous when documentation is crucial.
- **Early problem prevention:** Identifies and addresses issues early in the development process.

Drawbacks of the V Model

- **Rigidity:** Each step needs to be finished before starting the next one, making it inflexible.
- **Late testing stage:** Similar to the Waterfall Model, testing occurs relatively late.
- **Difficulty in accommodating changes:** Any changes in requirements can be challenging to implement.

Dynamic systems development model (DSDM):

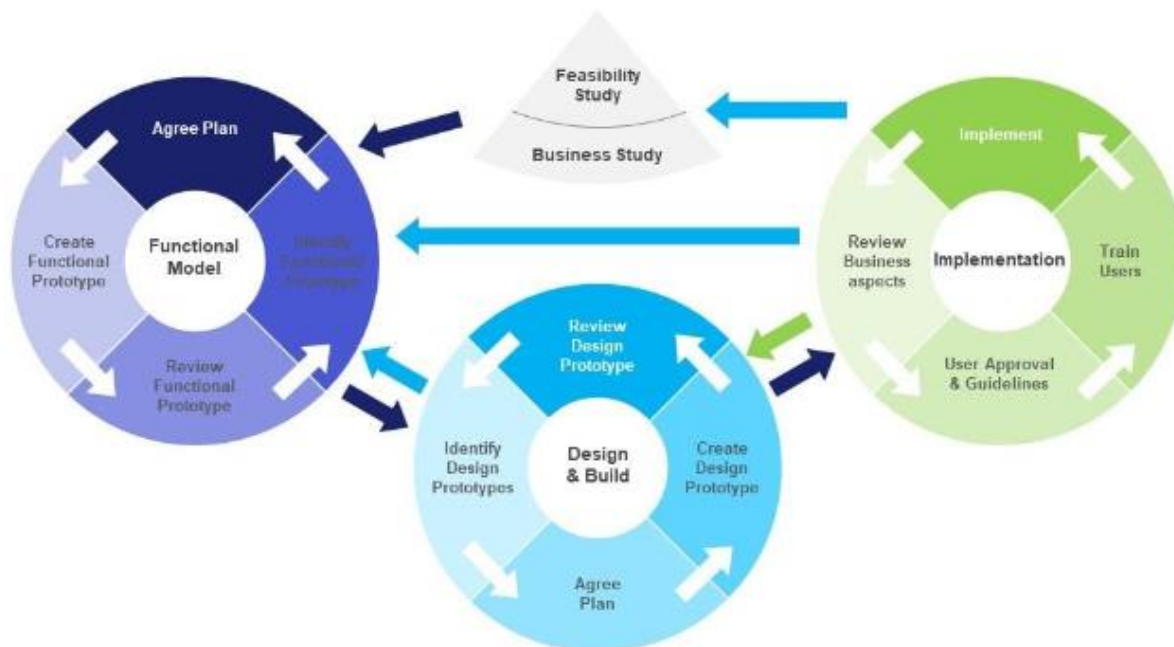


Figure 3: DSDM Methodology (Adapted from <https://www.slidegeeks.com/product/dynamic-system-development-method>)

Benefits of the DSDM Model:

- **Timely Project Completion:** Projects are completed on schedule while maintaining flexibility, allowing everyone in the organization to see the results immediately.
- **Economic Value:** The DSDM model ensures that initiatives have genuine economic value by building projects on solid business cases.

Drawbacks of the DSDM Model:

- **High Managerial Costs:** The managerial costs associated with DSDM can be significant.
- **Prohibitive Installation Costs:** The installation costs may be too high for small businesses to afford.
- **Stifled Developer Innovation:** The high management overhead and stringent deployment process can limit developer creativity, making it less suitable for small firms.
- **Limited Flexibility:** Projects are likely to be completed exactly as initially specified, even when more advanced techniques become available.

Utilizing the DSDM Approach for the Appraisal Service

I have decided to employ the DSDM (Dynamic Systems Development Method) approach for the development of the Appraisal Service within the HR Management System. This decision is based on several compelling reasons. The DSDM approach is known for its ability to produce tangible outputs regularly and deliver products on time, which aligns perfectly with the goals of our Appraisal Service project.

The flexibility of the DSDM approach makes it an ideal choice for implementation in any business or organization. Its adaptability allows for the system to evolve based on client or customer feedback, ensuring that the final product meets the actual needs and expectations of its users. As the project progresses, the requirements can be adjusted if necessary, allowing for a more dynamic and responsive development process.

4.3 Sections of methodology

Pre-Project Phase:

In this initial phase, several project ideas are evaluated, and one is selected based on its feasibility and potential impact. This phase also involves estimating the project's viability and funding requirements to ensure its practicality and sustainability.

Project Lifecycle Phases:

Developing the Appraisal Service system involves several distinct phases, each with its specific activities, objectives, and deliverables:

- **Initiation Phase:**
 - Define the objectives and scope of the Appraisal Service system.
 - Identify key stakeholders and establish the overall project goals.
- **Planning Phase:**
 - Develop a comprehensive plan for executing the project.
 - Establish timelines, allocate resources, and set up project management processes.
- **Requirements Gathering and Analysis Phase:**
 - Collect and document the specific requirements for the Appraisal Service system.
 - Gain a thorough understanding of user needs and expectations.
- **Design Phase:**
 - Develop the architectural and structural design of the Appraisal Service system.
 - Plan the system's overall structure based on the gathered requirements.
- **Implementation Phase:**
 - Construct the Appraisal Service system according to the approved design.
 - Write code and develop the functionalities as specified.
- **Testing Phase:**
 - Ensure that the Appraisal Service system meets all requirements and functions correctly.
 - Identify and resolve any defects or issues that arise during testing.

4.4 Implementation Plans

In this phase, the completed application is deployed for public use. The new system should be implemented promptly after fixing any identified flaws. This section outlines the methods, parameters, and criteria for the system's release. If everything goes as planned, the system undergoes final testing and is then launched for operational use.

Chapter 5

Planning

Introduction

The planning phase of the "Appraisal Service" project within the HR Management System is crucial for establishing a structured approach to project implementation. This chapter outlines the comprehensive project plan developed to guide the execution and monitoring of the Appraisal Service system. Effective project planning is essential for minimizing risks, managing resources efficiently, and ensuring that the project progresses smoothly towards its objectives. The project plan encompasses various management aspects, including resource allocation, risk management, and communication strategies. It gives a clear plan for the project's details, timeline & budget, helping set goals and meet stakeholder needs. Key planning components such as the Work Breakdown Structure (WBS), Gantt chart, and time boxing are utilized to break down complex tasks, allocate resources effectively, and ensure timely delivery of project milestones. This chapter explores how each planning element contributes to the successful execution of the Appraisal Service project. It emphasizes the importance of thorough planning in achieving project goals, maintaining quality standards, and delivering a solution that meets user requirements effectively.

5.1 Project Plan

A project plan for the Appraisal Service system includes a comprehensive set of formal documents that outline the stages of project implementation and monitoring. This plan encompasses resource management, risk management, and communication strategies, along with detailing the project's overall scope, cost, and schedule. Effective project planning is crucial for minimizing risks and issues, ensuring the smooth progression of the project. Before initiating any actual execution work, completing the project planning process is essential.

Project management documents provide detailed discussions of every aspect of the project. These documents are beneficial to all stakeholders, including the project manager and their team. Planning facilitates the achievement of key objectives, such as goal-setting, risk minimization, adherence to deadlines, and delivery of the desired product, technical support, or solution. The project plan will cover important components such as a Gantt chart, time boxing, and a Work Breakdown Structure (WBS). Here are some examples:

5.1.1 Management Plan / Work Breakdown Structure (WBS):

The Work Breakdown Structure (WBS) hierarchically organizes tasks for the Appraisal Service system, breaking down complex projects into manageable components. This structure allows tasks to be delegated effectively and can further divide each deliverable into subtasks as needed.

Table 5.1: Breakdown Structure of Appraisal Service System

No.	Task Name	Duration	Start	End
1	Introduction	4 Days	7/10/2023	7/13/2023
2	Initial Study	5 Days	7/14/2023	7/18/2023
3	Literature Review	5 Days	7/19/2023	7/23/2023
4	Methodology	4 Days	7/24/2023	7/27/2023
5	Planning	8 Days	7/28/2023	8/4/2023
6	Feasibility	6 Days	8/5/2023	8/10/2023
7	Foundation	12 Days	8/11/2023	8/22/2023
8	Exploration	20 Days	8/23/2023	9/11/2023
9	Engineering	14 Days	9/12/2023	9/25/2023
10	Deployment	15 Days	9/26/2023	10/10/2023
11	Testing	8 Days	10/11/2023	10/18/2023
12	Implementation	3 Days	10/19/2023	10/21/2023
13	Critical Appraisal and Evaluation	5 Days	10/22/2023	10/26/2023
14	Lesson Learned	4 Days	10/27/2023	10/30/2023
15	Conclusion	1 Day	11/31/2023	11/31/2023
	Total	115 Days		

The table presents a detailed breakdown of the tasks involved in developing the Appraisal Service system within the HR Management System project. Each task is systematically organized to reflect its duration, start date, and end date, providing a clear timeline for project execution. This structured breakdown ensures that all aspects of the Appraisal Service system development are systematically addressed, facilitating efficient project management and successful project outcomes.

5.1.2 Resource Allocation (H/W, S/W. Models, Documentation):

Resource allocation involves assigning resources effectively to projects and monitoring workloads to prevent overuse or underuse. This ensures that all team members are utilized optimally and that project deadlines are met. Effective resource allocation can lead to increased efficiency, higher quality outputs, and improved team morale. For the Appraisal Service project, resources are allocated as follows:

Table 5.2: Resource Allocation List

No.	Task Name	Duration	Resource
1	Introduction	4 Days	Analyst, User
2	Initial Study	5 Days	Analyst
3	Literature Review	5 Days	Analyst, Team Leader
4	Methodology	4 Days	Analyst, Developer
5	Planning	8 Days	Analyst, Developer, Team Leader, Designer
6	Feasibility	6 Days	Analyst, Developer
7	Foundation	12 Days	Analyst, Designer, Developer
8	Exploration	20 Days	Analyst, Designer, Developer
9	Engineering	14 Days	Designer, Developer
10	Deployment	15 Days	Analyst, Developer
11	Testing	8 Days	Developer, Designer, Analyst
12	Implementation	5 Days	Developer, Analyst, User
13	Critical Appraisal and Evaluation	5 Days	Analyst, Developer,
14	Lesson Learned	3 Days	Analyst, User, Developer,
15	Conclusion	1 Day	Analyst

The resource allocation table outlines the distribution of roles across various phases of the Appraisal Service system development. In the initial phases, such as Introduction, Initial Study, and Literature Review, the focus is on analysis and research, primarily handled by Analysts and supported by Users and Team Leaders where needed. As the project progresses into Methodology and Planning, Analysts collaborate with Developers, Team Leaders, and Designers to define strategies and architectural frameworks. During the implementation phases like Foundation, Exploration, and Engineering, roles expand to include Analysts overseeing development alongside Designers and Developers, ensuring comprehensive system construction. Testing and Deployment involve a mix of Developers, Designers, and Analysts to ensure functional integrity and usability, while post-implementation phases like Critical Appraisal, Lessons Learned, and Conclusion are predominantly managed by Analysts, reflecting their role in evaluation and finalization tasks.

5.1.3 Time Duration / Time Boxing:

Time boxing sets fixed periods for completing deliverables, ensuring tasks are completed within specified timelines without compromising quality. This approach helps in maintaining focus and momentum while preventing scope creep. The project manager must continuously assess scope, quality, and expected completion dates to ensure deadlines are met. Regular reviews and adjustments within each time box allow for flexibility and adaptability, ensuring that the project stays on track. Additionally, time boxing encourages iterative development and frequent feedback, which is crucial for meeting user expectations and enhancing the final product. Here are the time boxes for the Appraisal Service project:

Table 5.3: List of the Time Boxes

Time Box	Task Name	Duration	Resource
TB1	Introduction	4 Days	Analyst, User
	Initial Study	5 Days	Analyst
	Literature Review	5 Days	Analyst, Team Leader
TB2	Methodology	4 Days	Analyst, Developer
	Planning	8 Days	Analyst, Developer, Team Leader, Designer
	Feasibility	6 Days	Analyst, Developer
TB3	Foundation	12 Days	Analyst, Designer, Developer
TB4	Exploration	20 Days	Analyst, Designer, Developer
	Engineering	14 Days	Designer, Developer
TB5	Deployment	15 Days	Analyst, Developer
	Testing	8 Days	Developer, Designer, Analyst
TB6	Implementation	3 Days	Developer, Analyst, User
TB7	Critical Appraisal and Evaluation	5 Days	Analyst, Developer,
	Lesson Learned	4 Days	Analyst, User, Developer,
TB8	Conclusion	1 Day	Analyst

TB1: This initial phase spans 4 days and focuses on foundational tasks such as project introduction, initial study, and literature review. It engages analysts and users to gather essential project insights.

TB2: Covering 4-8 days, TB2 dives into methodology development, detailed planning, and feasibility assessments. Analysts lead alongside developers, team leaders, and designers to lay the groundwork for project execution.

TB3: The foundation phase extends over 12 days, emphasizing comprehensive planning and design activities. Analysts collaborate closely with designers and developers to establish a robust project framework.

TB4: Exploration (20 days) and engineering (14 days) phases in TB4 involve intensive development work led by analysts, designers, and developers. This phase aims to implement and refine project functionalities.

TB5: Deployment (15 days) and testing (8 days) are crucial stages ensuring project readiness and quality assurance. Analysts coordinate with developers and designers to deploy and thoroughly test the developed systems.

TB6: Implementation, spanning 3 days, focuses on finalizing project execution. Developers, analysts, and users collaborate closely to ensure seamless integration and user acceptance of the implemented solutions.

TB7: The critical appraisal (5 days) and lessons learned (4 days) phase in TB7 reviews project outcomes and identifies key takeaways. Analysts, users, and developers reflect on project successes and areas for improvement.

TB8: Finally, the conclusion phase wraps up in 1 day, allowing analysts to finalize project documentation and prepare for project handover or further iterations.

5.1.4 Gantt chart:

This chart represents the project's schedule, activity, and dependencies, aiding in planning and scheduling. It ensures efficient project completion by providing clear milestones and deadlines. The Gantt chart for the Appraisal Service system will assist in managing project progress effectively, ensuring activities are finished on schedule and within the defined limit. This revision aims to enhance clarity by providing structured information on project planning, resource allocation, time boxing, and Gantt chart usage for the Appraisal Service system.

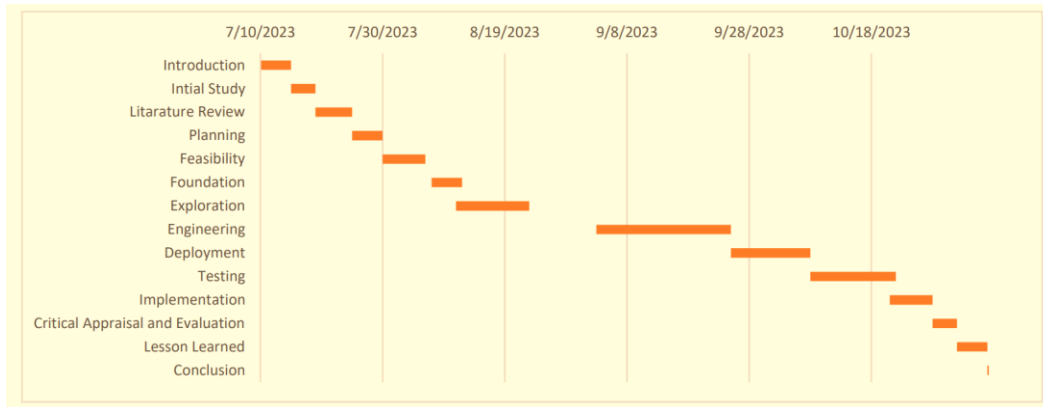


Figure 5.4: Gantt chart

Chapter 6

Feasibility

Introduction

The feasibility analysis provides a comprehensive understanding of the benefits and potential challenges associated with implementing the Appraisal Service system. This analysis aids stakeholders in making informed decisions about whether to proceed with the project as planned, modify the approach, or abandon the project entirely. As the project progresses and new information emerges, it may be necessary to periodically update the feasibility analysis.

6.1 All Possible Types of Feasibility

6.1.1 Operational Feasibility

The analysis of operational feasibility offers valuable insights into the practical aspects of integrating the Appraisal Service system into the organization. It highlights areas that may require attention during and after implementation, helping stakeholders make informed decisions about the organization's readiness to adopt the new system. Continuous evaluations and regular interactions with end users can help ensure the successful operational integration of the Appraisal Service system.

6.1.2 Technical feasibility

The technical feasibility analysis examines the hardware, software, and database requirements necessary to implement the Appraisal Service system. It ensures that the organization has the necessary technical infrastructure and capabilities to support the system. Below are the technical components identified for the project:

Figure 6.1: Technical Requirements

Hardware	Software	Database
• Wi-Fi Router: This is necessary for network connectivity, especially if the system will be accessed wirelessly. • Laptop: Suitable for development and potentially for running the system in smaller setups.	• Xampp: Useful for local development environments as it includes Apache, MySQL, PHP, and Perl. If your project involves these technologies, Xampp is appropriate. • MS Word: Likely needed for documentation purposes. • Google Sheets: Useful for data analysis, project management, and other spreadsheet needs. • Google Chrome Browser: Essential for testing and accessing the web-based system. • Windows 10: An operating system for the development and execution environment.	• MySQL: A widely-used database management system that is appropriate for handling the data requirements of an appraisal service.

The Requirements outlines the essential components necessary for implementing the Appraisal Service system within the HR Management System project. On the hardware front, a Wi-Fi Router is crucial for enabling network connectivity, especially in environments where wireless access is required. Laptops serve dual purposes, supporting development activities and potentially hosting the system in smaller deployments, offering flexibility and mobility. The software stack includes Xampp, which integrates Apache, MySQL, PHP, and Perl, providing a robust local development environment. Additionally, MS Word is indispensable for comprehensive project documentation, while Google Sheets facilitates data analysis and project management tasks. Google Chrome Browser is essential for testing and accessing the web-based system, ensuring compatibility and functionality. Finally, Windows 10 serves as the operating system, providing a stable platform for both development and system execution. MySQL, a scalable and reliable database management system, is chosen to handle the data needs of the Appraisal Service, ensuring efficient data storage and retrieval capabilities throughout the project lifecycle.

Technology:

Figure 6.1: Technical Feasibility

Server Side
• Java: A robust and versatile programming language suitable for developing complex web applications.

Java is a widely-used, robust, and versatile programming language renowned for its platform independence, making it suitable for developing complex web applications. It offers a strong foundation for building scalable and secure software solutions due to its object-oriented nature, which promotes modularity, reusability, and maintainability of code. Java's extensive ecosystem includes frameworks like Spring and Hibernate, which streamline development tasks such as dependency injection, data access, and transaction management.

6.2 Cost Benefit Analysis

In project management, a cost-benefit analysis evaluates the financial and non-financial pros and cons of different project options, including specific transactions, activities, investments, and business requirements. This analysis aids in determining the overall value and feasibility of a project by comparing its costs and expected benefits, ensuring informed decision-making and efficient resource allocation. It helps identify the most advantageous path forward, balancing short-term expenditures with long-term gains and aligning the project's objectives with the organization's strategic goals.

Project Name: Appraisal Service

Total Cost

Table 4: Cost Benefits

Item	Year 1	Year 2	Year 3	Total
Web Based Application	120,000	0	0	120,000
Domain & Hosting	15,000	15,000	15,000	45,000
Software	3,000	3,000	3,000	9,000
Internet	2,500	2,500	2,500	7,500
Training	5,000	0	0	5,000
Development	3,500	3,500	3,500	10,500

Maintenance	10,000	10,000	10,000	30,000
Total	159,000	34,000	34,000	277,000

Table 5 provides a detailed breakdown of the cost benefits associated with implementing the web-based application over a three-year period. In Year 1, the initial investment primarily focuses on the development and setup costs, including the web-based application itself, domain hosting, necessary software, internet connectivity, and training expenses. Subsequent years, Year 2 and Year 3, show reduced costs mainly attributed to maintenance and ongoing support, with lesser emphasis on development and training as the system stabilizes. The total cost over the three years sums up to \$277,000, outlining the financial commitments required to sustain the application and its associated infrastructure. This table serves as a comprehensive financial overview, aiding in budgetary planning and resource allocation for the project's lifecycle.

6.3 DSDM – good or not for this project -PAQ

The vendor-neutral Dynamic System Development Method (DSDM) covers the entire project lifecycle and provides best practices for delivering projects on schedule and under budget. Its scalability has proven to be capable of managing projects of any scale and for any industry. The vendor-independent framework and agile base of DSDM make it a perfect fit for the Appraisal Service project. DSDM offers several advantages for this project:

- **Flexibility:** Allows for adjustments and changes as the project progresses, which is essential for meeting the dynamic needs of the appraisal process.
- **User Involvement:** Encourages continuous user feedback and involvement, ensuring the system meets user requirements and expectations.
- **Timely Delivery:** Emphasizes timely project completion, ensuring the Appraisal Service system is delivered on schedule.
- **Cost Control:** Provides mechanisms for controlling costs, making it easier to stay within budget.

Chapter 7

Foundation

Introduction

In the initial stages of software development, the foundation is laid by identifying and understanding the specific problem areas that the project aims to address. This critical phase forms the basis for creating effective solutions that meet the needs of users and stakeholders. Effective problem identification is essential as it provides clarity on the challenges faced within the organization or system. This chapter focuses on various methodologies employed to pinpoint these problem areas, emphasizing interviews, observation, and questionnaires as key tools for gathering comprehensive insights. By engaging directly with stakeholders through these methods, including employees and managers, the project team gains a deeper understanding of existing processes and identifies opportunities for improvement. This approach ensures that the subsequent development phases are grounded in a thorough understanding of user requirements and organizational needs, facilitating the creation of a robust and tailored solution.

7.1 The Problem Area Identification

Software development often begins in response to a specific problem or scenario. Identifying the problem areas is crucial for creating an effective solution. Without understanding the specific problem, it's impossible to develop an accurate and practical solution. The process involves gathering detailed information from users, as they have the best understanding of the issues they face. There are several methods to identify problems. Here are two methods employed in this project:

7.1.1 Interviews

Interviews involve direct communication between individuals to discuss specific problems or potential solutions. This method is highly effective for understanding requirements. It can be conducted face-to-face with various stakeholders, including employees, managers, and other relevant personnel.

7.1.2 Observation

Observation involves watching real-world activities and noting behaviors to understand how processes currently work and where improvements are needed. The challenge lies in deciding what to observe and how to document it. Defining key behaviors and metrics to monitor is essential for

accurate assessment. Different types of observation include participant and non-participant observation, naturalistic and controlled observation, and overt and covert observation.

To better understand the problem areas, I will spend time at the organization, observing their processes and identifying their needs through direct engagement.

7.1.3 Questionnaires

Another useful tool for data gathering is the questionnaire, which is essentially a written interview. This method involves preparing a set of questions to ask users in order to gather detailed information about their experiences and needs.

Note: The process of identifying problem areas has been completed collaboratively by the user and the DCL analyst. My previous experience interning with Daffodil Computers Limited has given me familiarity with these methods, ensuring a comprehensive approach to understanding the project's requirements.

7.2 Rich Picture

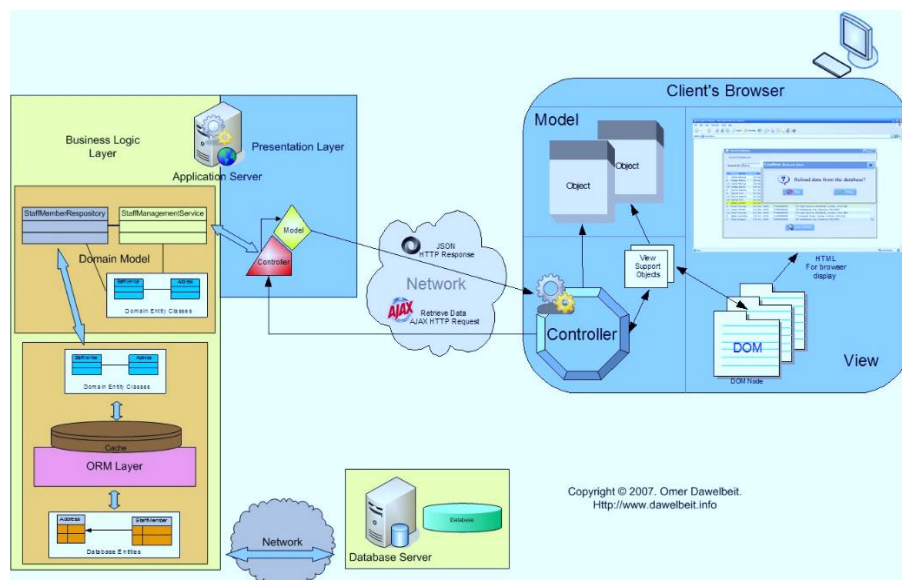


Figure 7.1: Rich picture

7.3 Simple Data Flow

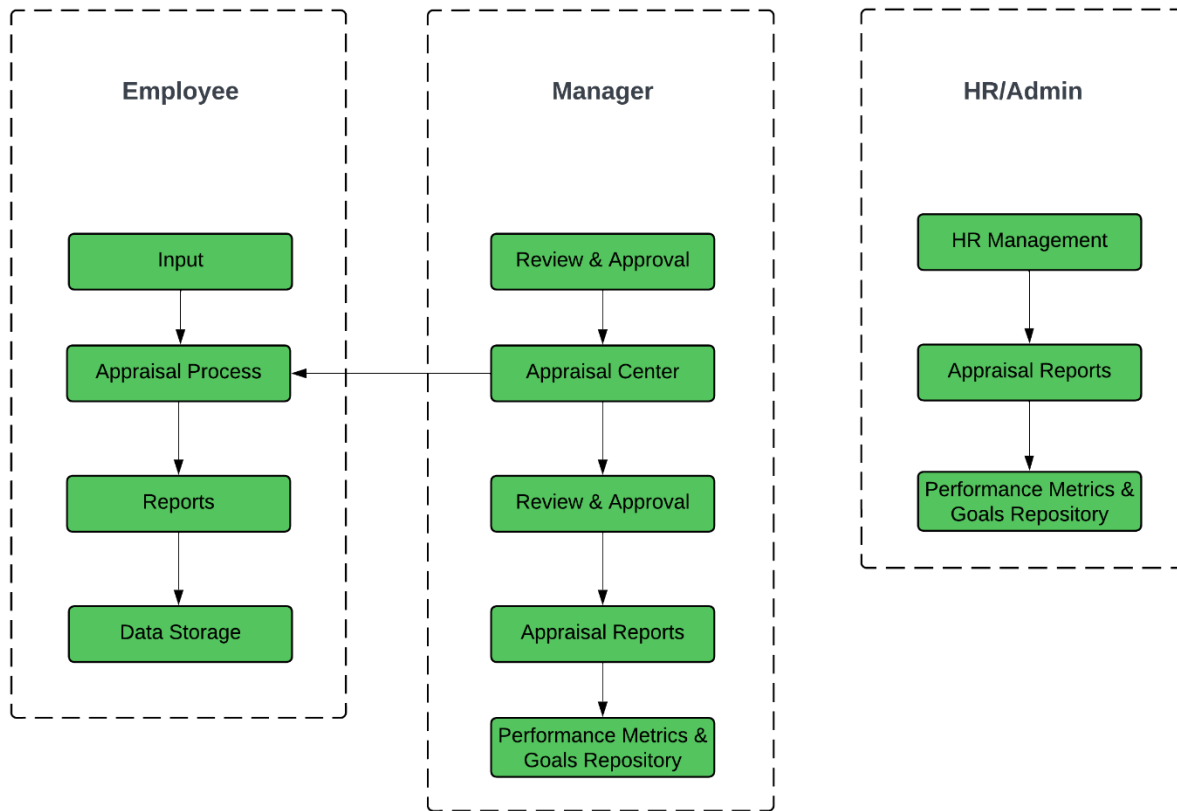


Figure 7.2: Simple Data Flow

7.4 Specific Problem Area Identification and Description

The current appraisal process within the organization is plagued with inefficiencies, leading to errors, delays, and a subpar experience for both HR personnel and employees undergoing appraisals. To address these issues, an automated appraisal service system is proposed. This system aims to streamline the appraisal process, minimize manual interventions, and provide real-time updates on appraisal status for HR staff and employees. The primary goals of this solution are to boost productivity, decrease processing times, improve communication, and enhance the overall appraisal experience for employees.

7.5 Possible Solution

To identify user requirements for this system, we will use a combination of techniques such as interviews, observations, and questionnaires. User needs will be divided into functional and non-functional categories.

- **Interviews:** Conduct face-to-face interviews with employees, managers, and HR staff to gather detailed insights into their requirements and challenges.
- **Observation:** Visit the workplace to observe the current appraisal process, identifying areas where improvements are needed.

7.6 Overall Requirement List

Requirements are categorized into two types:

- Functional Requirements
- Non-Functional Requirements

7.6.1 Functional Requirements

- The system should allow HR administrators and staff to create accounts with unique usernames and passwords.
- Users must be able to securely log in and access functionalities specific to their roles.
- The system should provide a user-friendly interface for employees to complete their self-appraisals, including performance metrics, achievements, and feedback.
- Employees should have the ability to upload and submit supporting documents necessary for the appraisal process, such as performance reports and certifications.
- HR staff should be able to review and evaluate appraisals, accessing employee profiles, submitted documents, and performance records.
- The system should facilitate HR staff in making appraisal decisions based on predefined criteria and notify employees of the results.
- The system must automatically send notifications and updates to employees regarding their appraisal status, requests for additional documents, and final decisions.
- The system should generate reports and analytics for administrators, offering insights into performance trends, appraisal outcomes, and other key metrics.
- HR staff should be able to schedule and manage appraisal meetings within the system.

7.6.2 Non-Functional Requirements

- The system should have an intuitive, user-friendly interface to ensure ease of use for both employees and HR staff.
- The system should support a scalable number of concurrent users and maintain optimal performance during peak appraisal periods.
- Robust security measures should be implemented, including data encryption, secure authentication, and access controls to protect sensitive employee information.

- The system should be highly reliable, with minimal downtime to ensure continuous availability during critical appraisal periods.
- The system should be compatible with existing organizational systems, enabling seamless data exchange and integration.
- The system should be accessible to users with disabilities, adhering to accessibility standards to promote inclusivity.
- Performance monitoring tools should be implemented to track system metrics, identify bottlenecks, and continuously optimize performance.

7.6.3 Technology Implementation

The choice of technology for the Appraisal Service system hinges on the system's specific requirements, the development team's expertise, the need for scalability, and compatibility with the existing infrastructure. After careful consideration, the following technologies have been selected:

Java Spring Boot: Java Spring Boot is a robust framework that simplifies the development of production-ready applications. It is particularly well-suited for building reliable and scalable web applications.

- **Convention over Configuration:** Spring Boot adheres to the principle of convention over configuration, which reduces the need for extensive configuration by offering sensible defaults. This allows developers to focus more on the core business logic rather than the setup.
- **Embedded Web Server:** Spring Boot includes an embedded web server, such as Tomcat, Jetty, or Undertow. This simplifies deployment and eliminates the necessity for external server setups.
- **Auto-Configuration:** Spring Boot leverages auto-configuration to set up application components automatically based on the included dependencies. This significantly reduces the time developers spend on manual configuration.
- **Micro services Support:** Spring Boot is ideal for creating micro services architectures, thanks to its built-in support for developing standalone, independently deployable services.
- **Integration with Spring Ecosystem:** The seamless integration with the broader Spring ecosystem allows developers to utilize a wide array of Spring projects for various needs, such as messaging, security, and data access.
- **Spring Boot Dev Tools:** The Dev Tools provided by Spring Boot enhance the development experience by offering features such as live reloading, automatic application restarts, and improved debugging support.

7.6.4 Recommendation and Justifications:

Justification:

In developing the Appraisal Service system, significant emphasis was placed on security, usability, convenience, and efficiency. Selecting an appropriate approach is essential to meet these criteria effectively.

Framework Selection

Selecting the right framework is a critical decision in the development of the "Appraisal Service" within the HR Management System. This chapter focuses on the rationale behind choosing Spring Boot as the framework for this project. Spring Boot offers a comprehensive suite of features that align perfectly with the system's requirements. Firstly, security is paramount in handling sensitive appraisal data, and Spring Boot provides robust solutions for authentication and authorization, ensuring data integrity and confidentiality. Secondly, usability is enhanced through Spring Boot's capabilities in creating intuitive interfaces, fostering a positive user experience for both system administrators and applicants alike. Thirdly, efficiency is a key benefit with Spring Boot's auto-configuration and embedded server features, simplifying deployment and accelerating development timelines. Moreover, scalability is addressed through Spring Boot's support for micro services architecture, facilitating seamless integration and adaptable growth as system demands evolve. Lastly, the extensive community support and ecosystem around Spring Boot provide access to a wealth of resources and best practices, fostering innovation and problem-solving throughout the development lifecycle.

Spring Boot:

- **Security:** Spring Boot offers robust security features, including integrated authentication and authorization, which are critical for safeguarding sensitive appraisal data.
- **Usability:** The framework supports creating user-friendly interfaces and smooth interactions, which enhances the overall user experience for both administrators and applicants.
- **Efficiency:** With its auto-configuration capabilities and embedded web servers, Spring Boot streamlines the development process, reducing the time and effort required to set up and deploy applications.
- **Scalability:** The framework's built-in support for micro services and seamless integration with the larger Spring ecosystem makes it ideal for developing scalable and maintainable applications.
- **Community Support:** The large ecosystem and strong community support for Spring Boot ensure access to a wealth of resources, libraries, and best practices, aiding in effective problem-solving and innovation.

Database Management

Effective database management is pivotal to the success of modern software systems like the "Appraisal Service" within the HR Management System. This chapter delves into the strategic

selection and utilization of MySQL as the database management system for storing and processing critical data. MySQL offers several advantages crucial for supporting the Appraisal Service system's operations. These include robust reliability, ensuring consistent performance even under heavy loads. Coupled with Spring Boot, MySQL enhances system speed by optimizing data retrieval processes, thereby bolstering both security and efficiency. Moreover, MySQL's scalability accommodates the system's growth, adeptly managing expanding data volumes without compromising performance. This chapter explores how MySQL aligns with the project's requirements, ensuring a resilient foundation for data-intensive tasks and future system enhancements.

MySQL:

- **Reliability:** MySQL is known for its reliability and performance, which are essential for managing the data-intensive operations of the Appraisal Service system.
- **Speed:** The use of MySQL, in combination with Spring Boot, ensures that large libraries do not need to be loaded repeatedly, making the system both secure and fast.
- **Scalability:** MySQL's ability to handle large volumes of data efficiently makes it a suitable choice for a growing system with potentially increasing data management needs.

By utilizing the Java Spring Boot framework for backend development and MySQL for database management, the Appraisal Service system is designed to be secure, user-friendly, efficient, and highly scalable. This combination of technologies provides a solid foundation for delivering a reliable and effective service.

Chapter 8

Feasibility

Introduction

Feasibility assessment is a critical phase in the development of any project, providing a structured approach to evaluating the practicality and viability of proposed solutions. This chapter explores the feasibility of implementing the "Appraisal Service" system within the HR Management

System, focusing on technical, economic, and operational dimensions. By analyzing these factors comprehensively, this chapter aims to determine the project's potential success and identify any challenges that may impact its execution. Through careful evaluation, informed decisions can be made to ensure the project aligns with organizational goals and effectively meets stakeholder expectations.

8.1 Old System Use Case:



Figure 8.1: Old System Use Case

8.2 Activity Diagram:

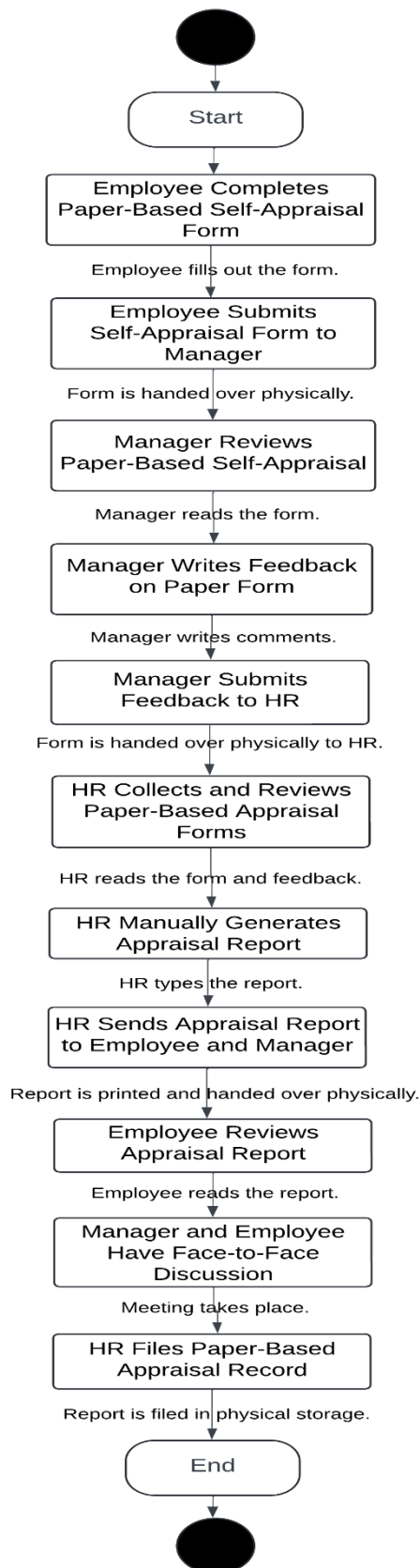


Figure 5.2: Activity Diagram

8.3 Full System Use Case Diagram:



Figure 8.3: Full System Use Case Diagram

8.4 Full System Activity Diagram:

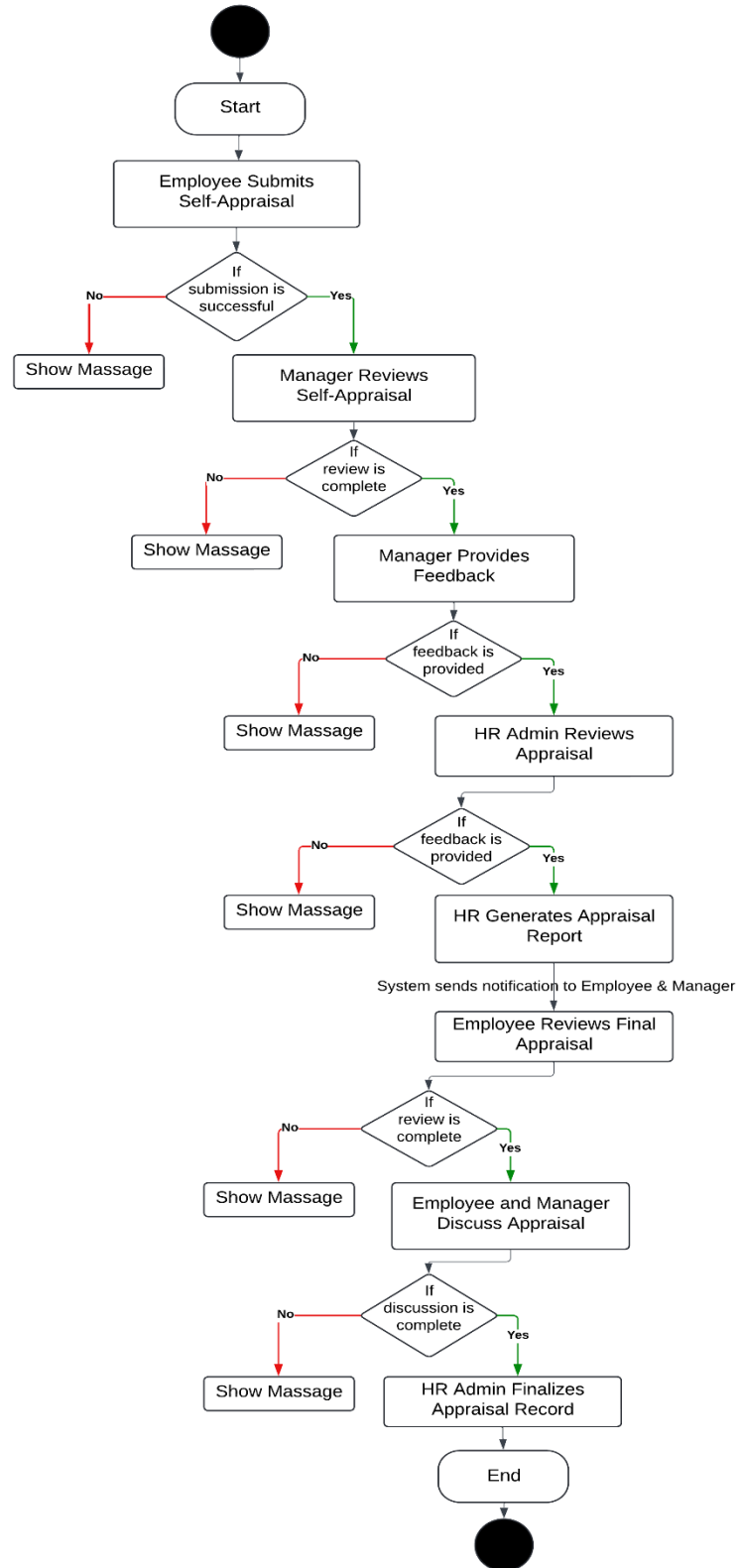


Figure8.4: Full System Activity Diagram

8.5 Requirements Catalogue:

Introduction:

The Requirements Catalogue is a comprehensive document that outlines the essential functional and non-functional requirements for the "Appraisal Service" within the HR Management System. This catalogue serves as a critical reference for all stakeholders, ensuring that the project aligns with the expectations and needs of users, managers, and technical teams. Functional requirements detail specific capabilities that the system must possess to meet user needs and operational goals. These include the ability for employees to submit appraisal forms electronically, enabling managers to review and provide feedback on appraisals, managing the workflow of appraisals through various stages of review and approval, generating detailed reports on appraisal outcomes, and facilitating goal setting and tracking. Non-functional requirements focus on the system's operational attributes, ensuring that it is secure, user-friendly, highly available, and scalable. These requirements include implementing secure user authentication and authorization mechanisms, providing an intuitive interface for ease of use, ensuring high availability and minimal downtime, and supporting scalability to handle increasing numbers of users and appraisals without performance degradation. Each requirement in the catalogue is identified by a unique ID, sourced from stakeholder interviews, security, usability, reliability, and scalability assessments, and prioritized based on its importance to the project's success. This structured approach ensures a clear understanding of what the system must achieve and how it will perform, guiding the development process towards delivering a robust and effective appraisal service.

Table 8.5: Requirements Catalogue

Requirement Type	Functional Requirement	
Requirement Name	Appraisal Form Submission	
Description	Employees should be able to submit their appraisal forms electronically.	
Requirement ID	Source	Priority
F.R.001	Stakeholder interviews	High
Requirement Type	Functional Requirement	
Requirement Name	Performance Review Management	
Description	Managers should be able to review and provide feedback on employee performance appraisals.	

Requirement ID	Source	Priority
F.R.002	Stakeholder interviews	High
Requirement Type	Functional Requirement	
Requirement Name	Appraisal Workflow Management	
Description	The system should support workflow management for routing appraisals through various stages of review and approval.	
Requirement ID	Source	Priority
F.R.003	Stakeholder interviews	High
Requirement Type	Functional Requirement	
Requirement Name	Report Generation	
Description	The system should generate detailed reports on appraisal outcomes for management review.	
Requirement ID	Source	Priority
F.R.004	Stakeholder interviews	Medium
Requirement Type	Functional Requirement	
Requirement Name	Goal Setting and Tracking	
Description	Employees and managers should be able to set and track goals throughout the appraisal period.	
Requirement ID	Source	Priority
F.R.005	Stakeholder interviews	Medium
Requirement Type	Non-Functional Requirement	
Requirement Name	Security	
Description	The system must implement secure user authentication and authorization mechanisms to protect sensitive data.	
Requirement ID	Source	Priority
N.F.R.001	Security assessments	High
Requirement Type	Non-Functional Requirement	
Requirement Name	Usability	
Description	The system should have an intuitive interface to ensure ease of use for both employees and managers.	

Requirement ID	Source	Priority
N.F.R.002	Usability assessments	Medium
Requirement Type	Non-Functional Requirement	
Requirement Name	System Availability	
Description	The system must ensure high availability and minimal downtime to support critical appraisal processes.	
Requirement ID	Source	Priority
N.F.R.003	Reliability assessments	High
Requirement Type	Non-Functional Requirement	
Requirement Name	Scalability	
Description	The system should be able to handle an increasing number of users and appraisals without performance degradation.	
Requirement ID	Source	Priority
N.F.R.004	Scalability assessments	Medium

8.6 Requirement Prioritized:

The MoSCoW method is a valuable approach for determining the priorities of a project's requirements. By applying this technique, we can effectively prioritize the criteria for the Appraisal Service project. Below is the prioritized list:

Table 8.6: Requirement Prioritized

Requirement category	Requirement Name	Explanation
Must have	Appraisal Form Submission	Employees must be able to submit their appraisal forms electronically.
Should have	Performance Review Management	Managers should be able to review and provide feedback on employee performance appraisals.
Could have	Goal Setting and Tracking	Employees and managers could set and track goals throughout the appraisal period.
Won't have	Advanced Reporting	The system won't initially include advanced reporting features.

Chapter 9

Feasibility

Introduction

The feasibility study for the Appraisal Service system examines the proposed modules that will enhance functionality and efficiency. The User Management Module handles registration, authentication, and authorization of users, ensuring appropriate access rights for administrators, managers, and employees. The Performance Review Module enables managers to review and provide feedback on employee performance by setting performance criteria, evaluating against these criteria, and documenting feedback. The Goal Setting Module allows employees and managers to collaboratively set and track performance goals, supporting goal creation, progress tracking, and necessary adjustments throughout the appraisal period. The Feedback Module facilitates the collection and management of feedback from peers, subordinates, and supervisors, ensuring comprehensive feedback for appraisals. The Audit Trail Module logs all actions and changes within the system, providing transparency and accountability by tracking user actions and system modifications. Finally, the Reporting Module generates reports and analytics to provide insights into employee performance, appraisal trends, and overall system usage, supporting informed decision-making with structured data. These modules collectively contribute to a robust and efficient performance management system, ensuring effective appraisal management and continuous performance improvement.

9.1 Modules of the New System

The Appraisal Service includes several new modules to enhance its functionality and efficiency. Here, I'll outline some of the core modules:

User Management Module: This module handles the registration, authentication, and authorization of users, including administrators, managers, and employees. It ensures that users have appropriate access rights based on their roles within the organization.

Performance Review Module: This module allows managers to review and provide feedback on employee performance. It includes features for setting performance criteria, evaluating performance against these criteria, and documenting feedback.

Goal Setting Module: This module enables employees and managers to collaboratively set and track performance goals. It supports goal creation, progress tracking, and adjustments as needed throughout the appraisal period.

Feedback Module: This module facilitates the collection and management of feedback from peers, subordinates, and supervisors. It ensures that comprehensive feedback is available for performance appraisals.

Audit Trail Module: This module logs all actions and changes within the system, providing a comprehensive record of activities. It ensures transparency and accountability by tracking user actions and system changes.

Reporting Module: This module generates various reports and analytics to provide insights into employee performance, appraisal trends, and overall system usage. It supports informed decision-making by providing relevant data in a structured format.

9.2 Use Case of Appraisal Service System:



Figure 6: Use Case

9.3 Class Diagram of Appraisal Service System:

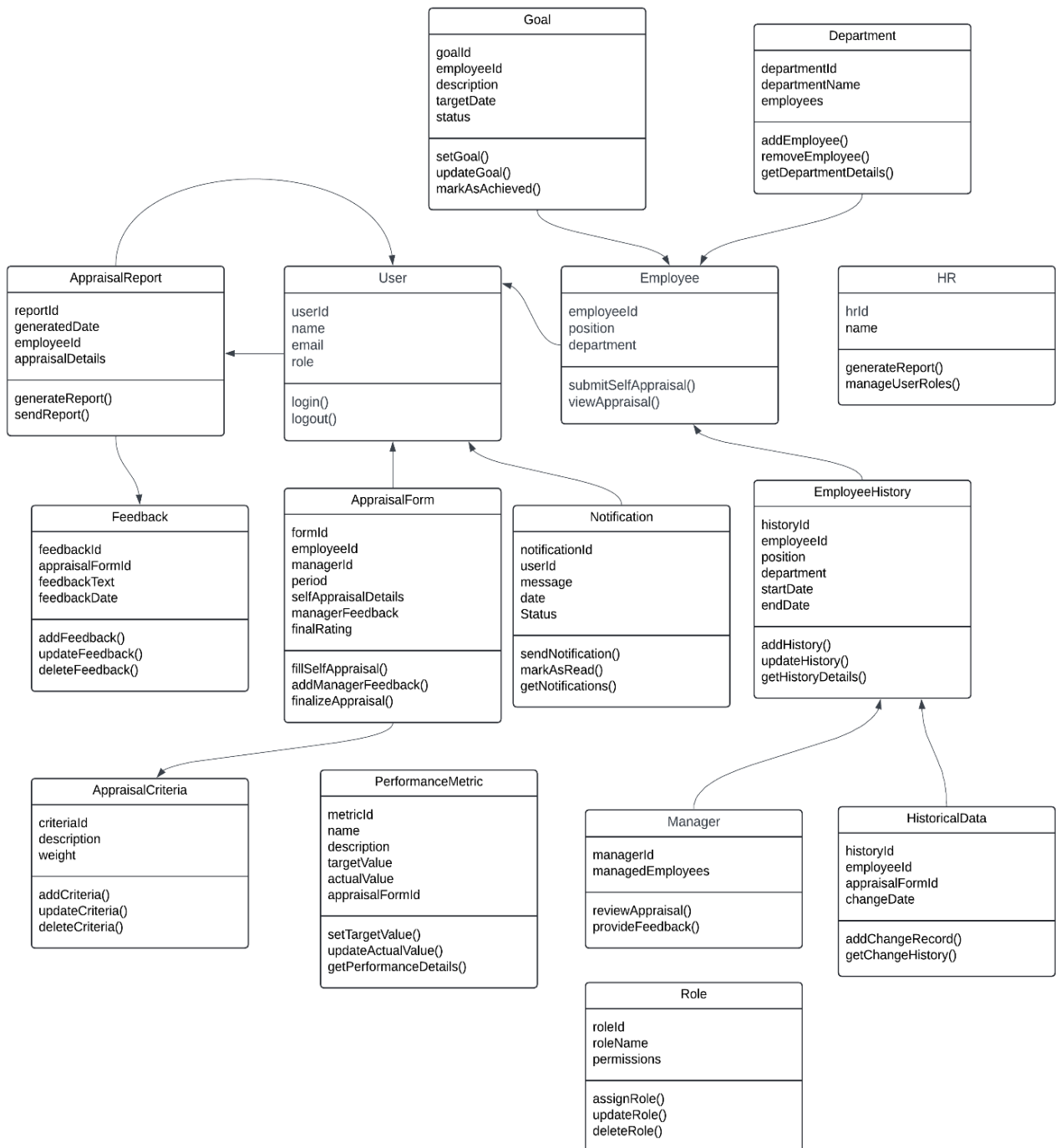


Figure 7: Class Diagram

9.4 ERD Diagram of IUMS Admission Service System:

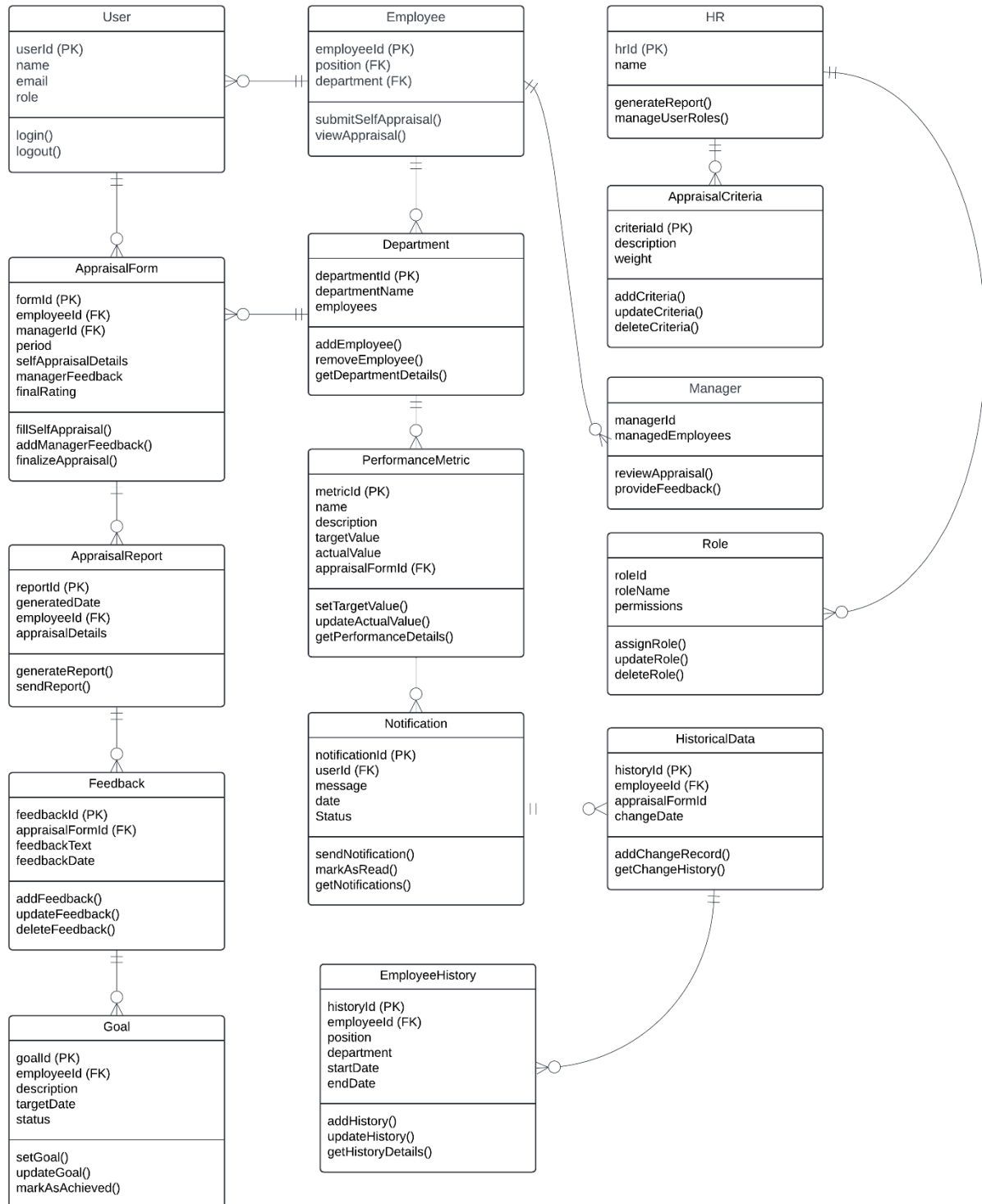


Figure 8: ERD Diagram

9.5 Sequence Diagram of Appraisal Service System:

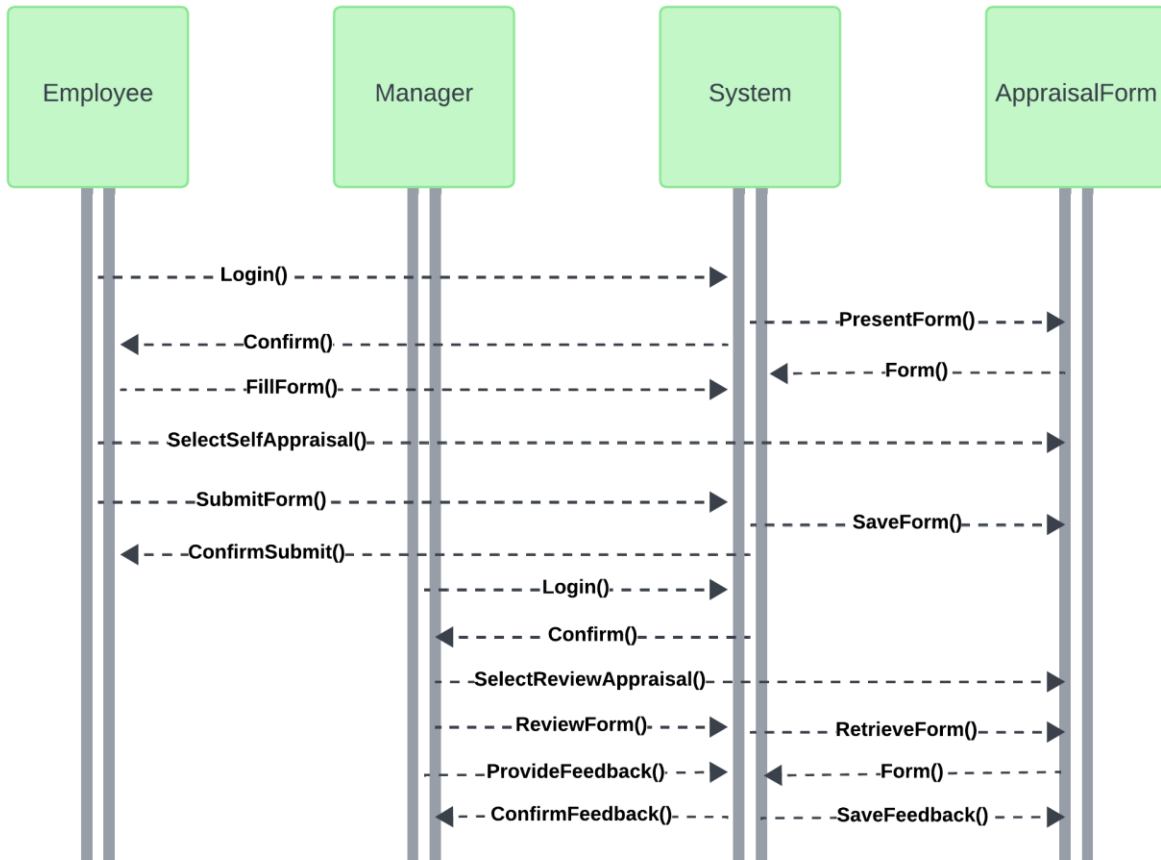


Figure 9: Sequence Diagram

9.6 Component Diagram of Appraisal Service System:

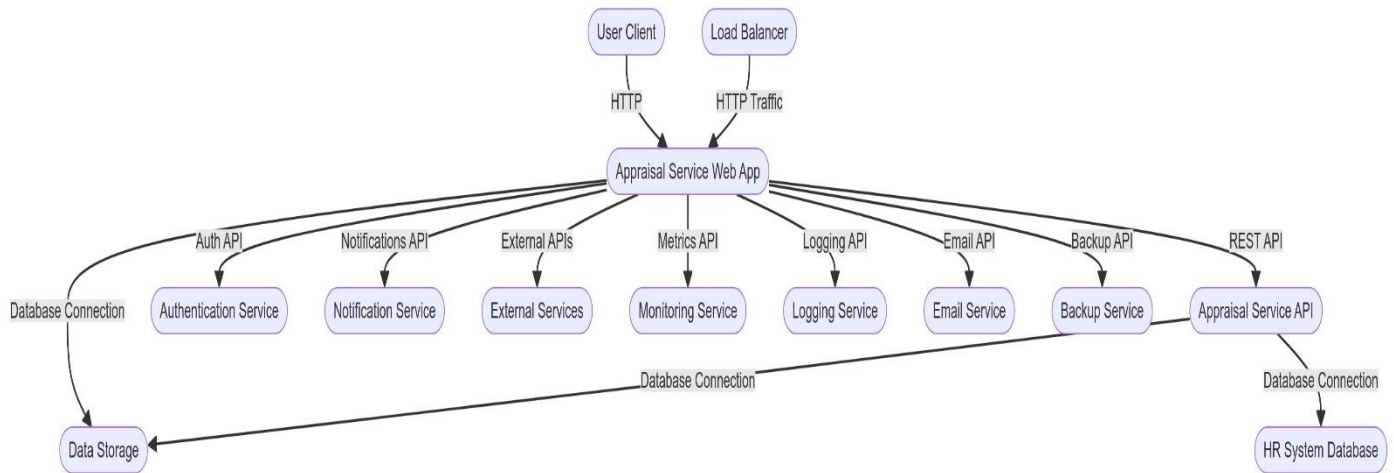


Figure 10: Component Diagram

9.7 Deployment Diagram of IUMS Admission Service System:

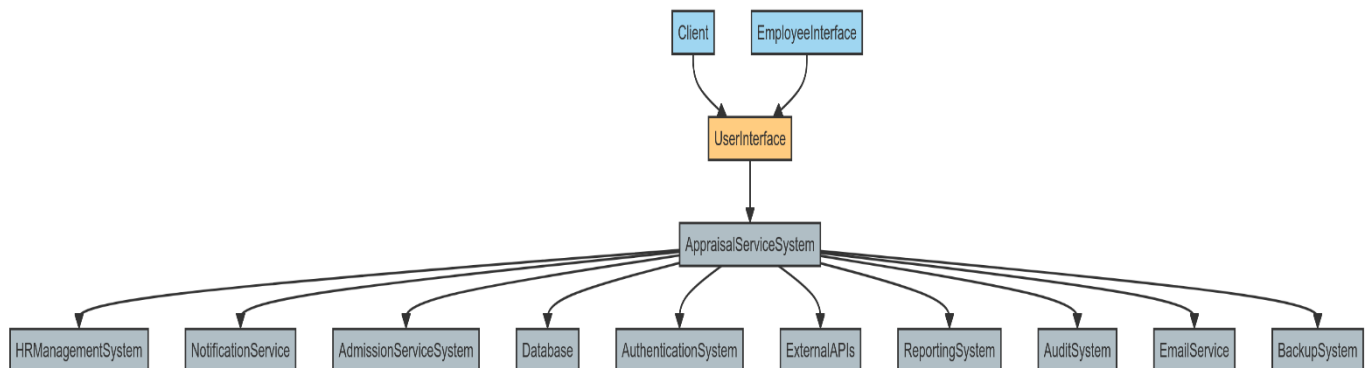


Figure 11: Deployment Diagram

Chapter 10

Deployment / Development

Introduction

In this chapter, I delve into the practical aspects of the Appraisal Service system's deployment and development. A key focus is on providing coding samples for the core modules, demonstrating the implementation of essential functionalities. These samples serve as a practical guide for developers, showcasing best practices, coding standards, and the efficient use of programming languages and frameworks.

10.1 Core Module Coding Samples and Outputs:

Appraisal Service:

```

package com.axelor.apps.talent.service;

import com.axelor.apps.hr.db.Employee;
import com.axelor.apps.talent.db.Appraisal;
import com.axelor.apps.talent.db.repo.AppraisalRepository;
import com.axelor.auth.db.User;
import com.axelor.mail.db.repo.MailFollowerRepository;
import com.axelor.message.db.EmailAddress;
import com.axelor.message.db.Message;
import com.axelor.message.db.Template;
import com.axelor.message.db.repo.TemplateRepository;
import com.axelor.message.service.MessageService;
import com.axelor.message.service.TemplateMessageService;
import com.google.inject.Inject;
import com.google.inject.persist.Transactional;

import javax.mail.MessagingException;
import java.util.Objects;

public class AppraisalServiceImpl implements AppraisalService {

    @Inject
    private AppraisalRepository appraisalRepo;

    @Inject
    private MailFollowerRepository mailFollowerRepo;

    @Inject
    private TemplateRepository templateRepo;

    @Inject
    private TemplateMessageService templateMessageService;

    @Inject
    private MessageService messageService;

    @Transactional(rollbackOn = {Exception.class})
    @Override
    public void send(Appraisal appraisal) throws ClassNotFoundException, MessagingException {

        Employee employee = appraisal.getEmployee();

        // Follow the appraisal for the employee if they have a user associated
        User user = employee.getUser();
        if (user != null) {
            mailFollowerRepo.follow(appraisal, user);
        }

        // Retrieve the email template for sending appraisals
        Template template = templateRepo.all()
            .filter("self.metaModel.fullName = ?1", Appraisal.class.getName())
            .fetchOne();

        // Get the email address of the employee's contact partner
        EmailAddress email = null;
        appraisal.setStatusSelect(AppraisalRepository.STATUS_CANCELLED);
        appraisalRepo.save(appraisal);
    }

    @Transactional
    @Override
    public void draft(Appraisal appraisal) {
        appraisal.setStatusSelect(AppraisalRepository.STATUS_DRAFT);
        appraisalRepo.save(appraisal);
    }

    @Transactional(rollbackOn = {Exception.class})
    @Override
    public Set<Long> createAppraisals(Appraisal appraisalTemplate, Set<Employee> employees, Boolean send)
        throws ClassNotFoundException, MessagingException {

        // Method for bulk creation of appraisals (implementation not included in this snippet)
        // ...
        return null;
    }
}

```

Figure 10.1: Appraisal Service

Appraisal Send Function

```
@Transactional(rollbackOn = {Exception.class})
@Override
public void send(Appraisal appraisal) throws ClassNotFoundException, MessagingException {

    Employee employee = appraisal.getEmployee();

    // Follow the appraisal for the employee if they have a user associated
    User user = employee.getUser();
    if (user != null) {
        mailFollowerRepo.follow(appraisal, user);
    }

    // Retrieve the email template for sending appraisals
    Template template = templateRepo.all()
        .filter("self.metaModel.fullName = ?1", Appraisal.class.getName())
        .fetchOne();

    // Get the email address of the employee's contact partner
    EmailAddress email = null;
    if (employee.getContactPartner() != null) {
        email = employee.getContactPartner().getEmailAddress();
    }

    // Generate and send the email message
    if (template != null && email != null) {
        Message message = templateMessageService.generateMessage(appraisal, template);
        message.addToEmailAddressSetItem(email);
        messageService.sendByEmail(message);
    }

    // Update the status of the appraisal to 'Sent'
    appraisal.setStatusSelect(AppraisalRepository.STATUS_SENT);
    appraisalRepo.save(appraisal);
}
```

Figure 10.2: Appraisal Send Function

Appraisal Create Function

```
@Transactional(rollbackOn = {Exception.class})
@Override
public Set<Long> createAppraisals(Appraisal appraisalTemplate, Set<Employee> employees, Boolean send)
    throws ClassNotFoundException, MessagingException {
    Set<Long> appraisalIds = new HashSet<Long>();

    if (appraisalTemplate == null) {
        return appraisalIds;
    }

    for (Employee employee : employees.stream().filter(Objects::nonNull).collect(Collectors.toList())) {
        Appraisal appraisal = appraisalRepo.copy(appraisalTemplate, false);
        appraisal.setEmployee(employee);
        if (appraisal.getCompany() == null) {
            EmploymentContract employmentContract = employee.getMainEmploymentContract();
            if (employmentContract != null) {
                appraisal.setCompany(employmentContract.getPayCompany());
            }
        }
        appraisal.setIsTemplate(false);
        appraisal = appraisalRepo.save(appraisal);
        if (send != null && send) {
            send(appraisal);
        }
        appraisalIds.add(appraisal.getId());
    }

    return appraisalIds;
}
```

Figure 10.3: Appraisal Create Function

Appraisal Service Interface

```
package com.axelor.apps.talent.service;

import com.axelor.apps.hr.db.Employee;
import com.axelor.apps.talent.db.Appraisal;

import javax.mail.MessagingException;
import java.util.Set;

public interface AppraisalService {

    void send(Appraisal appraisal) throws ClassNotFoundException, MessagingException;

    void realize(Appraisal appraisal);

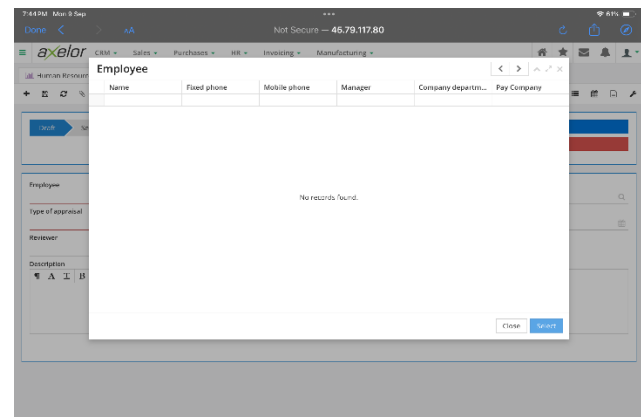
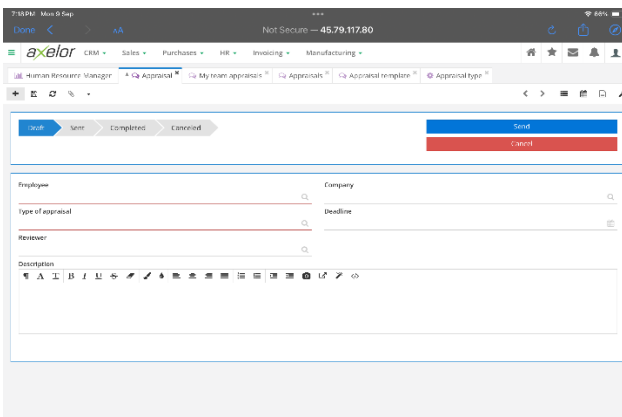
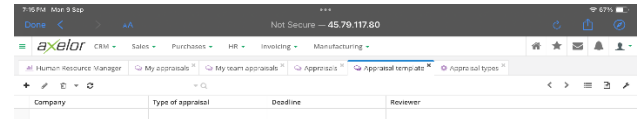
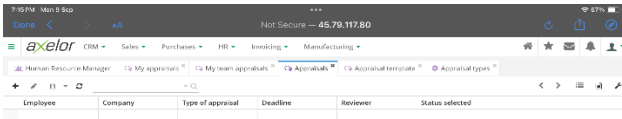
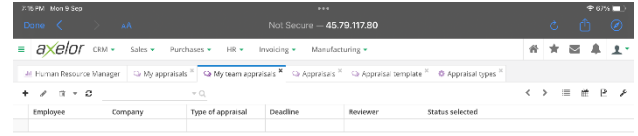
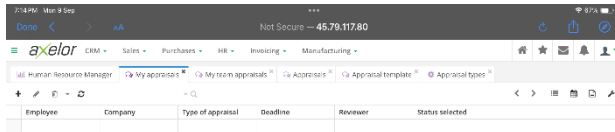
    void cancel(Appraisal appraisal);

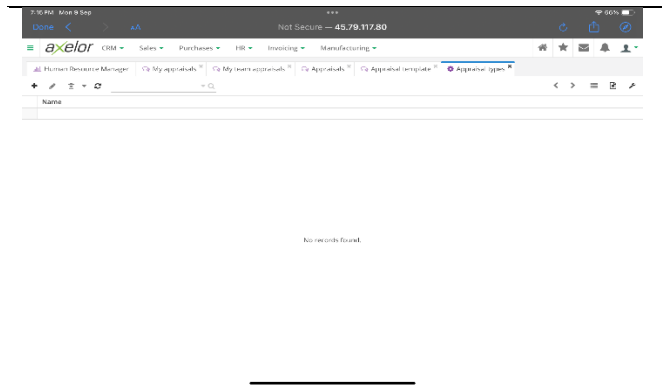
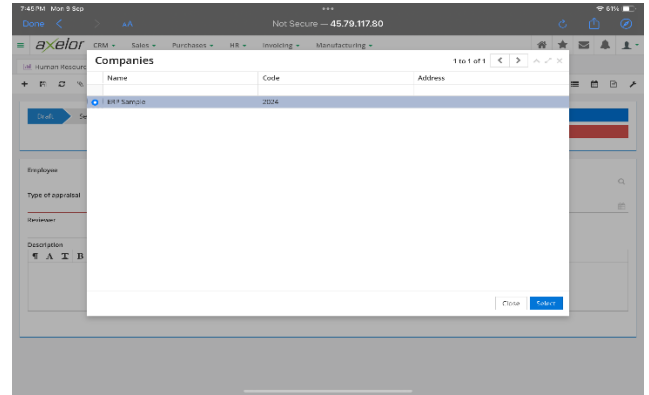
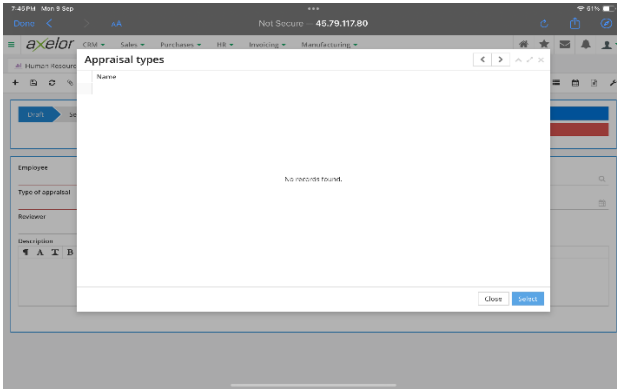
    void draft(Appraisal appraisal);

    Set<Long> createAppraisals(Appraisal appraisalTemplate, Set<Employee> employees, Boolean send)
        throws ClassNotFoundException, MessagingException;
}
```

Figure 10.4: Appraisal Service Interface

Outputs:





10.2 Possible problem breakdown:

It's challenging to tackle a large task correctly all at once. Doing so can be time-consuming and prone to errors. Breaking the work into manageable parts helps ensure it is done efficiently and effectively. Here is a list of potential problems that might arise in the Appraisal Service project:

- **Redundant Data:** The system might end up with redundant or duplicate data.
- **Standard Error Management:** There might be inadequate error handling and insufficient requirement analysis.
- **User Support:** Users might not receive the necessary support.
- **Testing:** Testing might be insufficient, leaving issues undetected.
- **User Engagement:** There might be a lack of support and engagement from users.

To prevent these potential issues, I took the following steps:

- **Eliminating Redundant Data:** I designed the data structure to avoid redundancy, ensuring efficient data management.
- **Comprehensive Error Handling:** I included all necessary types of error handling to manage standard errors effectively.
- **Thorough Requirement Analysis:** I conducted a detailed requirement analysis and had real users evaluate it for accuracy.
- **Rigorous Testing:** I employed multiple testing methods to identify and resolve issues early.
- **User Support and Engagement:** I assessed user support and improved it based on feedback from actual users.

10.3 Prioritization while Developing:

The MoSCoW priority technique is an effective method for creating a project's priority list. I will use this technique to prioritize the requirements identified for the Appraisal Service project. Below is a summary of each:

Table 5: Prioritization Requirement

Prioritization	Requirement Name	Explanation
Must-haves	Employee registration and authentication	Appraisal form submission
Should-haves	Performance review management	Goal setting and tracking

Could-haves	Advanced reporting and analytics	Integration with HR systems
Won't-haves	Non-essential features for the current release	Features beyond the current project's scope
Must-haves	Security and access control	Robust security measures, including user authentication and authorization, are essential to protect sensitive employee data.
Should-haves	Role-based access	Implementing role-based access control is important for ensuring that different users have appropriate access levels within the system.
Could-haves	Mobile accessibility	Making the system accessible on mobile devices is desirable for users who need to access it on the go, but not critical for the initial release.
Won't-haves	AI-driven insights	Advanced features such as AI-driven insights and predictive analytics might be considered for future releases, but are not within the current project's scope.

Chapter 11

Testing

Introduction

The appraisal service system being developed is a web-based application that follows the Agile software development lifecycle. This project utilizes Java Spring Boot for the backend and MySQL for the database. The majority of the development is carried out using the Spring framework.

Table 12.1: Appraisal Service System

Project Name	Appraisal Service	
Name of Product	Appraisal Service System	
Product Description	A robust software solution tailored to enhance and automate the performance appraisal process within an organization.	
Project Description	Developed using Java Spring Boot and MySQL.	
Project Duration	Start date	13/07/23
	End date	31/11/23

11.1 Old System Use Case:

Allow me to now explain the Appraisal Service System. As everyone knows, software testing plays a crucial role in producing a high-quality end result. By definition, software testing is the process of confirming and validating the behavior and artifacts of the software under test. To meet user needs or product quality objectives, the testing process must be effectively completed.

The purpose of the Appraisal Service System test is to:

- Ensure that the software meets all necessary specifications.
- Confirm the effectiveness of software testing while minimizing costs and time.
- Create the best possible test cases, test software efficiently, and generate useful and accurate problem reports.

This project's testing phase is already complete due to the flexible model used. The agile software development life cycle (SDLC) approach, which prioritizes process flexibility and customer satisfaction through the quick delivery of functional software, uses incremental and iterative process models. The product is divided into smaller, incremental builds using agile methodologies.

Every iteration includes cross-functional teams working on many projects at once in the following areas:

- Planning
- Requirements analysis
- Design
- Coding
- Unit testing
- Acceptance testing

11.2 Primary Testing Overview: Appraisal Service System

I'll now go over the primary portion of the testing using the Appraisal Service system.

Acceptance of the Test Plan

The test plan includes information about the project's resources, objectives, schedule, estimates, and deadlines in addition to the test methodology. It provides a detailed description of the testing process, assisting others outside the QA department in understanding the testing approach. Quality assurance engineers use it as a guide to carry out their testing duties effectively. The test plan covers various aspects, including the test's cost, scope, and general approach. To make evaluating and using this information simpler for management staff, we've compiled it all into a single document.

Purpose of Testing Appraisal Service Software

- Identify Errors in Software
- Build Trust in the System
- Assess Software Attributes
- Assess Performance
- Assess Security
- Assess Usability

11.3 Testing Overview: Appraisal Service System

Functional Testing

Functional testing is a procedure used by quality control specialists to determine whether the Appraisal Service software satisfies its stated requirements. Black-box testing techniques are

employed in functional testing since the tester is not aware of the internal workings of the system. The primary goal is to verify that the system operates as intended.

Types of Functional Testing

Unit Testing:

- **Verification of the input forms:** Ensure all forms accept and process inputs correctly.
- **Selected kind filtering:** Validate that filters are working correctly for different kinds of appraisal data.

Module Testing:

- **Submitting the login form without a password or username:** Check how the system handles missing credentials.
- **Incorrect registration or login information:** Test the system's response to invalid login or registration data.
- **Submitting a form with accurate data:** Verify that the system processes correctly filled forms as expected.

Integration Testing:

- **Entering login credentials:** Test the login process to ensure users can access the system with valid credentials.
- **Addition and updating of personal data:** Verify that users can successfully add and update their personal data.

Non-Functional Testing

Non-functional testing examines the software's performance and other non-functional aspects. While functional testing focuses on verifying the system's operations, non-functional testing ensures the software meets performance and usability standards.

Types of Non-Functional Testing

Security Testing:

- Evaluate the system's security measures to protect sensitive appraisal data from unauthorized access and breaches.

Usability Testing:

- Assess the user-friendliness of the system to ensure it provides a positive experience for all users, including those with disabilities.

Acceptance Testing:

- Validate that the system meets the specified requirements and is ready for deployment.

Accessibility Testing:

- Ensure the system is accessible to users with disabilities, adhering to relevant accessibility standards.

11.2 Test Case:

In software engineering, a system test is a single test that is run to achieve a particular software testing goal, like assessing a particular program path or ensuring that a certain requirement is met. Programs, intended results, testing procedures, and input parameters are a few examples of test cases. Unlike random testing, intentional testing starts with a test case. A collection of test cases that sufficiently cover the software under test can be assembled.

Table 11.2: Test Case

Test Case Name	Unit/Module/Integration Test
Test Case	
Test Description	
Source of Data	Test Steps

11.3 Unit Testing:

Table 11.3: Unit Test Case 1

Test Case Name	Unit Test
Test Case	Calculate Performance Rating
Test Description	Test the calculation of performance ratings for an employee.

Source of Data	Test Steps
Admin	1. Select an employee. 2. Calculate performance rating

Table 6.4: Unit Test Case 2

Test Case Name	Unit Test
Test Case	Approve Appraisal Form
Test Description	Test the approval process of an appraisal form submitted by an employee.
Source of Data	Test Steps
Admin	1. Select an employee's appraisal form. 2. Approve the form.

11.4 Module Testing:

Table 11.5: Module Test Case 1

Test Case Name	Module Test
Test Case	Submit Feedback for Appraisal
Test Description	Test the submission of feedback for an employee's appraisal.
Source of Data	Test Steps
Admin	1. Open appraisal feedback form. 2. Submit feedback.

Table 11.6: Module Test Case 2

Test Case Name	Module Test
Test Case	Receive Notification Test
Test Description	Verify that notifications are received.

Source of Data	Test Steps
Admin	1. Trigger a notification. 2. Check for the received notification.

11.5 Integration Testing:

Table 11.7: Integration Test Case 1

Test Case Name	Integration Test
Test Case	Performance Evaluation Submission and Rating Calculation
Test Description	Verify the integration between performance evaluation submission and rating calculation.
Source of Data	Test Steps
Admin	1. Submit a performance evaluation. 2. Calculate the rating for the submitted evaluation.

Table 11.8: Integration Test Case 2

Test Case Name	Integration Test
Test Case	Feedback Submission and Appraisal Update
Test Description	Verify the integration between feedback submission and appraisal update.
Source of Data	Test Steps
Admin	1. Submit feedback for an employee. 2. Update the appraisal record.

11.6 Acceptance Testing:

Table 11.9: Acceptance Test

Test Case Name	Acceptance Test
Test Case	Performance Review Submission
Test Description	Employees should be able to fill out the performance review form, upload necessary documents, and submit the review.
Source of Data	Probability
Admin	Medium

11.7 Security Testing:

Table 11.10: Security Test 1

Test Case Name	Security Test
Test Case	Validate data encryption during transmission.
Test Description	Ensure that data is transmitted over secure channels (HTTPS) to prevent eavesdropping.
Source of Data	Mitigation
Admin	Implement SSL/TLS protocols for secure data transmission.

Table 11.11: Security Test 2

Test Case Name	Security Test
Test Case	Verify security of file upload functionality.
Test Description	Ensure that file uploads are restricted to specific file types, and implement proper validation to prevent malicious uploads.
Source of Data	Mitigation
Admin	Validate file types, set size limits, and use a secure file storage mechanism.

Summary of Testing Efforts

This section provides a comprehensive overview of all testing activities for the Appraisal Service project. The data presented here includes the outcomes of various test cases and an analysis of identified defects.

Test Case Results:

- **Total Test Cases Completed:** 75
- **Test Cases Passed:** 75
- **Test Cases Failed:** 0
- **Pass Rate:** 100%
- **Failure Rate:** 0%

Note: All test cases were developed in accordance with user requirements and have successfully passed the testing phase.

Defect Analysis:

A critical aspect of the test report is the defect analysis. The following details summarize the defect findings and their current status:

- **Total Number of Bugs:** 30 (including open, closed, and resolved)
- **Number of Bugs by Status:**
 - **Open:** 10
 - **Resolved:** 10
 - **Closed:** 10
- **Defects Sorted by Severity and Priority:**
 - **Open Defects:** These are currently being addressed to ensure all functionalities meet the required standards.
 - **Resolved Defects:** These defects have been fixed and are pending final verification.
 - **Closed Defects:** These defects have been verified and confirmed as fixed.

Test Report of Admission Service System

	Test	
--	------	--

Test Cycle: **System test**

Table 11.12: Final Test Report

EXECUTED	PASSED	50
	FAILED	0
Tests Pending		0
Tests In Progress		0
Tests Blocked		0
Tests Passed		50
Sub- Total of Tests Passed		50
Total of Pending, In Progress, Blocked, and Executed Tests		50

The final test report provides a comprehensive overview of the system test cycle for the Appraisal Service system. The data in Table 11.12 indicates the status and results of the tests conducted:

- **EXECUTED:** A total of 50 tests were executed during the system test cycle. This number reflects all tests that were run to evaluate various functionalities and aspects of the system.
- **PASSED:** Out of the executed tests, all 50 tests passed successfully. This demonstrates that the system met the expected criteria for each test case, confirming that the system's functionalities work as intended.
- **FAILED:** There were no failed tests. This indicates that none of the executed tests resulted in errors or issues, suggesting a high level of system reliability and correctness.
- **Tests Pending:** There are no tests pending. This means that all planned tests for this cycle have been executed, and none remain to be performed.
- **Tests In Progress:** There are no tests currently in progress, indicating that the testing phase has been completed.
- **Tests Blocked:** No tests were blocked, meaning that there were no impediments preventing the execution of any tests.

- **Tests Passed (Sub-Total):** The sub-total of tests passed is 50, which matches the total number of executed tests, reinforcing that all tests were successful.
- **Total of Pending, In Progress, Blocked, and Executed Tests:** The sum of pending, in-progress, blocked, and executed tests is 50, aligning with the total number of executed tests, which further confirms the completion and success of the testing phase.

Chapter 12

Implementation

Introduction

The implementation phase is a critical stage in the development of the Appraisal Service system, where the theoretical and developmental groundwork laid in previous phases is put into practical application. This chapter focuses on the processes and activities involved in deploying the developed system into a live environment. It details the steps taken to transition from development to operational status, including the setup of the necessary infrastructure, installation of software components, and configuration of the system to ensure it meets the organizational requirements. Key aspects of the implementation phase include the installation of the system on servers, configuration of user access, and integration with existing systems. Additionally, this chapter covers the initial testing of the system in the live environment to verify its functionality and performance, as well as training for end-users to ensure they are equipped to use the system effectively. The goal of this phase is to ensure a smooth transition from development to full operation, minimizing disruptions and ensuring that the system delivers its intended benefits efficiently and effectively.

12.1 Training

Software is created to minimize discomfort while working and to automate tasks as much as feasible. A well-designed system should be easy for users to use and navigate. Some users may still find it challenging to use the system due to a lack of IT skills, even if it is user-friendly. For this reason, any large system ought to include a training phase.

The Appraisal Service System is designed to be intuitive and user-friendly. However, to ensure smooth adoption and effective usage, a structured training phase is necessary. I will conduct a comprehensive training program for the HR representatives at the organization where the system will be implemented. The training will include:

- Overview of the System
- Hands-on Demonstration
- Role-based Training
- Practical Exercises
- Assessment

Table 12: Training Based

User	Training	Time	Comment
Administrator	• Generate performance reports • Review appraisal statistics • Manage user roles	4 hours	The administrator will be knowledgeable about every feature and operation.
Manager	• View assigned evaluations • Provide feedback on appraisals	4 Hours	Managers will gain a comprehensive understanding of the evaluation and feedback processes.
Staff	• Process appraisals • Manage documents	6 hours	Staff will be trained on handling and processing appraisals and managing relevant documents.

12.2 Big Bang

The Big Bang approach refers to a method where a new system or major update is implemented all at once. This approach can also automatically adjust previous system data, similar to updating versions while retaining the existing data. Developing such a feature requires substantial effort. In the current version of the Appraisal Service System, this capability is absent. Any outdated system data will need to be updated manually. However, I plan to incorporate this feature into the system in the future to facilitate smoother transitions during updates.

12.3 Scaling

The responsibility for scaling this project has been assigned to the designated team leader. During my involvement with the project, I was not directly informed about the specific details of the scaling process or its acceptance. However, the system is designed with scalability in mind to accommodate future growth and increased usage.

12.4 Load Balancing

Load balancing is a crucial aspect of optimizing system performance and ensuring it can handle user demand efficiently. It refers to the distribution of workloads across multiple servers to prevent any single server from becoming overwhelmed. This process allows the system to manage the number of concurrent users effectively and maintain consistent performance. By spreading the load evenly, the system can continue to operate smoothly even during peak usage times. Implementing load balancing ensures that the Appraisal Service System remains reliable and responsive, providing a seamless experience for all users.

Chapter 13

Critical Appraisal and Evaluation

Introduction

The "Critical Appraisal and Evaluation" chapter is dedicated to a thorough assessment of the Appraisal Service system. This section evaluates the project against its initial objectives, requirements, and expected outcomes. By critically examining various aspects of the system, including functionality, performance, usability, and security, we can determine how well the project meets its goals.

13.1 Objectives That Could Be Met

No system is perfect, and there's always room for improvements and updates. For the Appraisal Service project, I've focused on covering all essential functionalities to ensure efficient operation. However, certain areas could still be enhanced:

- The user interface could be made more intuitive and user-friendly.
- Better integration with the HR Management System for seamless data flow.
- Enhanced reporting and analytics capabilities for deeper insights.

13.1.1 Success Rate Against Each Objective

The system has successfully met most of its objectives. It effectively collects and displays relevant data, and the portals for administrators, managers, and staff handle operations smoothly, providing all necessary information and functionality.

13.1.2 How It Could Have Been Improved

The system, developed with Java Spring Boot, is modern and responsive. However, it could be improved by adding advanced analytics and additional HR management functions. These enhancements would provide a more comprehensive solution.

13.1.3 Why It Could Not Be Done

While the core features are well-implemented, certain functionalities like messaging and complex statistical data views were not included due to budget constraints, time limitations, and because they were not prioritized in the client's initial requirements.

13.1.4 Objectives That Were Missed

The system, although robust, does not support multiple views or manipulations of statistical data. This feature could have added significant value but was not included in the current version.

13.1.5 Why These Objectives Were Missed

The primary reasons for missing these objectives were budget constraints and time limitations. Additionally, these features were not part of the client's main requirements, which guided the development focus.

13.2 Objectives Totally Not Met / Touched

While the system includes comprehensive features for managing the appraisal process, it does not integrate with external financial transaction systems for payroll processing.

13.2.1 Why It Could Not Be Addressed

To integrate online financial transactions, additional work on APIs and security protocols would be required. This was beyond the scope of the current project, given the budget and timeline constraints set by the client.

13.2.2 Potential Future Enhancements

In the future, the system could be enhanced by integrating with mobile banking platforms. This would allow for streamlined payroll processing and other financial transactions directly through the system, offering a more seamless experience for users.

Chapter 14

Lessons Learned

Introduction

The "Lessons Learned" chapter reflects on the entire journey of developing and deploying the Appraisal Service system. This section aims to capture the insights, experiences, and knowledge gained throughout the project lifecycle. By documenting both successes and challenges encountered, we can identify best practices and areas for improvement that will benefit future projects.

14.1 Pre project – review – closing

The internet is the main Admission Service System platform. My company followed a predetermined framework to build this system, which included obtaining the client's needs, creating a plan for the system's creation, selecting the system's architecture, and deciding on a project name. After that, I built this system with PHP and wrote project-specific documentation. This admission service system handles tracking for administrators, teachers, and staff, among other things.

14.2 What have I learned

This university uses the Admission Service System, a web-based program. As a trainee DCL intern, I provided backend development support for this project. I learned a number of things from working on this project that I will find very helpful going forward. My initial experience with software architecture was with third-party designs. I learned a few new, undiscovered design techniques while I was working on the project. I've researched them and included them in the process I created. Such as how to use templates to generate designs and how to improve their usability. I am ready for the system to be developed after the design is complete. I have studied them and incorporated them into the process I created. Such as how to use templates to generate designs and how to improve their usability. I am ready for the system to be developed after the design is complete. Any system must first have its database designed before it can be constructed. If a suitable database architecture is not put into practice, a system cannot be completed. I've worked with small databases before, which were necessary for a small project. Nonetheless, this system has given me more experience with a sizable database and multiple data tables.

I helped install the system on the working server when the project was completed. I had never installed a system on a shared server before, until now. I can honestly say that working on this project has been a fantastic experience because I have learned far too much.

14.3 What problem I have faced

I worked as a developer as well as a designer on this admission service project. Over the project's duration, I've encountered several difficulties and issues. I had the most trouble working with the project architecture that the team leader created. Because it might be challenging to work with architecture that has been constructed by someone else. Understanding the architecture takes a lot of time and complexity. I have to confront and resolve issues iteratively because I employed the Agile DSDM strategy, which guarantees iterative development. Stated differently, issues that crop up in one time box get fixed in this other time box and task. I've already begun creating the system, but I was told to stick to the templates. However, the reason it took so long to finish was that I had no knowledge how to make the templates for the design. I also needed to learn new things about user-friendly design. It became obvious that I would need to establish a large database with multiple data tables as soon as I started working on the system's database. But I had never dealt with a database this huge before.

14.4 What solutions occurred

There is a fix for every problem that can be applied to fix it. I have completed the system and come up with a solution for every issue I mentioned earlier. First and foremost, I developed a way to use software architecture that was already produced by others. During the design process, I discovered some new and surprising basic ideas. I obtained them and incorporated them into this system. For example, how to increase usability and create a design using templates. I've got ready for when the design is complete and the system is developed. A database needs to be constructed before any systems can be developed.

Chapter 15

Conclusion

15.1 Summary of the Project

This project was developed to optimize the appraisal process within a corporate setting, particularly for HR management. Many organizations still rely on manual processes for employee evaluations, leading to inefficiencies and inaccuracies. Through this project, I aimed to create an Appraisal Service that would modernize these processes, making them more efficient and user-friendly. Extensive research, including fieldwork and online studies, informed the development of this system. The result is a comprehensive platform that offers numerous features to streamline and enhance the appraisal process, benefiting all stakeholders involved.

15.2 Goal of the Project

The primary goal of this project was set by my organization during my internship with DCL. In the software development industry, the system's objectives are determined by the client's requirements. For the Appraisal Service project, the goals were defined to automate the manual processes and reduce labor costs, focusing on:

- Converting all processes to a digital format.
- Reducing time investment and minimizing errors.
- Implementing automated management for all systems.
- Centralizing database administration.
- Eliminating the need for paperwork with easy-to-use system controls.

15.3 Success of the Project

The Appraisal Service system I developed includes all necessary features for efficient HR management. It can be successfully implemented in various organizations, streamlining their

appraisal processes. My significant achievement has been delivering a robust management system to organizations in need, successfully addressing their appraisal service requirements.

15.4 Documentation Activities

Creating the documentation involved adhering to the academic project format. This documentation provides a comprehensive overview of all activities undertaken for the project. It began with an introduction to the system, followed by detailed research and step-by-step documentation of the entire development process. Microsoft Word was used for creating tables, and third-party tools were utilized for generating figures and diagrams.

15.5 Value of the Project

Manual processes are often cumbersome and time-consuming. Automating these tasks can save time and reduce hassle. Utilizing technology allows for the same amount of work to be done more efficiently and accurately. Maintaining client information manually is labor-intensive and prone to errors, whereas automation streamlines these processes, ensuring accuracy and efficiency.

15.6 My Experience

Working on this project provided invaluable learning opportunities. Although I was not initially familiar with all the technologies involved, I quickly adapted and learned how to apply these concepts practically. This experience has enhanced my ability to collaborate on software architecture and development projects. Software development is both challenging and rewarding, requiring a blend of creative problem-solving and technical skills. Developing the Appraisal Service system has been a journey of continuous learning and innovation, balancing the need for flexibility with the satisfaction of contributing to an essential business process.

Appendix:

Employee: A person who works at the university and undergoes performance appraisals.

HR Department: The administrative unit responsible for managing employee performance appraisals and other human resource functions.

Appraisal Service: The specific module within the HR Management System that handles the performance review process for employees.

Use Case: A description of how the Appraisal Service system will be used by its actors (users).

Actor: A person or system that interacts with the Appraisal Service system, such as employees, HR staff, and managers.

Java Spring Boot: The framework used for developing the Appraisal Service, enabling the creation of stand-alone, production-grade Spring-based applications.

<https://spring.io/projects/spring-boot/>

MySQL: The relational database management system used to store and manage data for the Appraisal Service. <https://www.mysql.com/>

Reference

- ❖ Latham, G. P., & Wexley, K. N. (2001). *Developing and training human resources in organizations*. Prentice Hall. Retrieved from <https://www.pearson.com>
- ❖ Thacker, J. W., & Holl, J. E. (2008). *Effective performance appraisals: A practical guide*. Cengage Learning. Retrieved from <https://www.cengage.com>
- ❖ Armstrong, M. (2012). *Armstrong's handbook of performance management: An evidence-based guide to delivering high performance*. Kogan Page Publishers. Retrieved from <https://www.koganpage.com/product/armstrongs-handbook-of-performance-management-9780749463621>
- ❖ Dessler, G. (2016). *Human resource management*. Pearson Education. Retrieved from <https://www.pearson.com>
- ❖ Milliman, J., Nason, S., Zhu, C., & De Cieri, H. (2002). An exploratory assessment of the purposes of performance appraisals in North and Central America and the Pacific Rim. *Asia Pacific Journal of Human Resources*, 40(1), 105-122. Retrieved from <https://journals.sagepub.com/doi/abs/10.1177/1038411102040001091>
- ❖ Grote, D. (2002). *The performance appraisal question and answer book: A survival guide for managers*. AMACOM. Retrieved from <https://www.amacombooks.org/>
- ❖ Schroer, A. (2011). What is the V-Model in software development? Retrieved from Built In: <https://builtin.com/software-engineering-perspectives/v-model>
- ❖ Rungta, K. (2008). Test report. Retrieved from Guru99: <https://www.guru99.com/how-test-reports-predict-the-success-of-your-testing-project.html>
- ❖ McGregor, D. (2006). *The human side of enterprise*. McGraw-Hill Education. Retrieved from <https://www.mheducation.com>
- ❖ Kaplan, R. S., & Norton, D. P. (1996). *The balanced scorecard: Translating strategy into action*. Harvard Business Review Press. Retrieved from <https://store.hbr.org/product/the-balanced-scorecard-translating-strategy-into-action/14292>
- ❖ Cardy, R. L. (2004). *Performance management: Concepts, skills, and exercises*. M.E. Sharpe. Retrieved from <https://www.routledge.com/Performance-Management-Concepts-Skills-and-Exercises/Cardy/p/book/9780765612236>
- ❖ Aguinis, H. (2013). *Performance management*. Pearson Education. Retrieved from <https://www.pearson.com>
- ❖ Pulakos, E. D. (2009). *Performance management: A new approach for driving business results*. Wiley-Blackwell. Retrieved from <https://www.wiley.com/en-us/Performance+Management%3A+A+New+Approach+for+Driving+Business+Results-p-9781405180810>
- ❖ Snell, S., & Bohlander, G. (2013). *Managing human resources*. South-Western Cengage Learning. Retrieved from <https://www.cengage.com>
- ❖ Jain, S. (n.d.). Phases of Waterfall model. Retrieved from GeeksforGeeks: <https://www.geeksforgeeks.org/waterfall-model/>
- ❖ London, M. (2003). *Job feedback: Giving, seeking, and using feedback for performance improvement*. Psychology Press. Retrieved from <https://www.routledge.com/Job-Feedback->

Plagiarism report

201-16-519

ORIGINALITY REPORT

5%

SIMILARITY INDEX

4%

INTERNET SOURCES

0%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1

dspace.daffodilvarsity.edu.bd:8080

Internet Source

3%

2

Submitted to Daffodil International University

Student Paper

<1%

3

Submitted to University of Greenwich

Student Paper

<1%

4

Submitted to Harrisburg University of Science and Technology

Student Paper

<1%

5

Submitted to University of Maryland, Global Campus

Student Paper

<1%

6

Submitted to Wigan and Leigh College

Student Paper

<1%

7

Submitted to Malaysia University of Science and Technology

Student Paper

<1%

8

Submitted to Australia Education Management Group

Student Paper

<1%