3.1 Chapter Introduction

The methodology chapter presents the systematic processes employed in the design, development, and validation of the Automated Marking System, using Artificial Intelligence, for Multimedia University. It involves techniques for requirements gathering, system analysis, design, implementation, testing, and deployment. The study adopts a mixed-methods approach that ensures the proposed solution is reliable and applicable. This chapter integrates methodologies such as interviews, surveys, advanced AI development techniques supported by scholarly evidence.

3.2 Literature Review Methodology

Literature review provided the background upon which the research was based in order to understand what existed in knowledge, systems, and methodologies relating to automated grading. It aimed at identifying the theoretical and technical underpinnings of AMS and AI in education.

Process of Conducting the Literature Review

Database Selection: Academic databases like IEEE Xplore, SpringerLink, Google Scholar, and JSTOR were used for accessing peer-reviewed journals and conference proceedings.

Search Strategy: Keywords included “Automated Grading,” “AI in Education,” “NLP for Assessments,” and “Machine Learning in Evaluation.” Boolean operators were applied to refine results (e.g., "AND," "OR").

Screening: Articles were screened for relevance based on abstracts, and full-text analysis was performed for shortlisted papers.

Findings from the Literature Review

Huang et al. (2019) discussed the scalability of AMS, especially dealing with a large number of data items for multiple-choice questions.

Sharma & Dey (2022) discussed the potential of Natural Language Processing in grading essays, pointing to challenges regarding context understanding and mitigation of bias.

Baker (2020) points out ethical considerations for AI grading, especially those around fairness and transparency.

The insights from these studies were then used to shape the system design and identify specific gaps that needed to be addressed in the AMS for Multimedia University.

3.3 Methodology for Requirement Specification and Data Collection

The requirements were obtained by stakeholder engagement and analysis of the existing systems so that the AMS could be designed to suit the real-world requirements.

3.3.1 Requirement Specification

The functional and non-functional requirements were then identified as follows:

Functional Requirements

It must cater to multiple-choice, short-answer, and essay questions.

It will have to provide students with detailed feedback based on rubrics.

Faculty must be able to upload rubrics and specify grading parameters in the system.

Non-Functional Requirements

Usability

The system should have an intuitive user interface for both the faculty and the students.

Scalability

The system should support simultaneous assessments of over 10,000 students.

Security

Data encryption to ensure students' records and grades remain secure.

3.3.2 Data Collection Techniques

Interviews

15 faculty members across departments, 30–45 minutes long.

Faculty stressed timely feedback, comprehensive comments, and uniform grading standards.

Questionnaires

Into the hands of 200 students, regarding experiences of manual grading.

86% reported dissatisfaction with time-lags in feedback while 78% pointed out inconsistencies in grading.

Observation

The research team of the study observed the grading workflows in the recently conducted examination period. Time and effort used in the script distribution, doing calculations manually, and data entry.

Data Collected for System Training

A dataset of 5,000 student essays and short answers was collected from previous semesters.

Grading rubrics and faculty comments were digitized to train the AI model.

3.4 Methodology for System Analysis

3.4.1 Current System Analysis

The manual grading process was documented to identify pain points and areas for improvement:

Context Diagram: Showed the interaction between faculty, students, and the grading system. Manual interventions, such as assigning grades and compiling results, were highlighted as bottlenecks.

Data Flow Diagram DFA,

Level 0 DFD: This represented the whole process from submission receipt to feedback return.

Level 1 DFD: Detailed at sub-process level with detailed rubric setup, grading, and error checks.

Key Findings

There was high dependency on human effort, leading to delays in giving feedback.

Grade calculations and data entry had a lot of errors.

Inconsistencies in essay score given were due to the absence of automated scoring tools.

3.5 Methodology for System Design

3.5.1 Database Design

The database scheme had to be carefully planned to assure efficiency in storing data, retrieving information, and ensuring data security. MySQL was selected because it is reliable and capable of handling large volumes of work.

Entities and Relationships

Students Table: maintained student's ID, name, and enrollment information.

Assessments Table: questions maintained, rubric, and deadlines for submission.

Grades Table: scores, feedback, and timestamps

Data Integrity Measures

Primary Keys: uniquely identified records.

Foreign Keys: maintained table relationships, like from the student table to grades.

Encryption: key fields containing grades and feedback.

3.5.2 Data Flow and Use Case Diagrams

Proposed System DFD

Level 0: The inputs are student submissions and faculty-defined rubrics. Outputs include grades and feedback.

Level 1: It gives the breakdown of processes such as data preprocessing, grading, and report generation.

Use Case Diagrams

Key users (faculty and students) and their interactions with the AMS were modeled.

3.5.3 User Interface Design

Faculty Dashboard: Question uploading facilities, defining of rubrics, reviewing results.

Student Portal: The facility for submitting assignments and retrieving detailed feedback in them.

3.5.4 AI Algorithms Used

Objective Assessments: Graded using answer-matching algorithms with preloaded keys.

Subjective Assessments:

NLP Models: Evaluated essays for coherence, grammar, and relevance to the content.

Rubric Alignment: Scores were calculated by applying faculty-provided grading rubrics.

3.6 Methodology for System Implementation

Technologies Used

Backend: Python with TensorFlow for ML and SpaCy for NLP.

Frontend: HTML/CSS/JavaScript with Flask for dynamic content rendering.

Database: MySQL for secure data storage and retrieval.

Implementation Phases

Development of Prototype: For the validation of core functionalities, an MVP was developed which included grading multiple-choice questions, followed by feedback. Full Implementation: In addition to the MVP, features like essay grading and scalability for large datasets will be integrated. Security Features: Data encryption at rest and secure login mechanisms will be provided for sensitive data. 3.7 Methodology for System Testing Test Plan Unit Testing: Focus on individual components such as the login system, grading algorithms, and database queries.

Integration Testing: Ensured that the frontend, backend, and database layers interacted seamlessly.

System Testing: Measured general performance and dependability by simulating real-world conditions.

Testing Techniques

Black-Box Testing: Tested the system's capability for input processing and generating correct outputs without going into the internal code.

Stress Testing: Simulated heavy workloads-e.g., 5,000 concurrent submissions-to ensure stability of the system.

User Acceptance Testing: Tested by faculty and students to determine usability and effectiveness.

Results

The accuracy for objective grading was 100%, and it reached 92% for essay grading, quite close to human evaluators.

Feedback delivery time was reduced by up to 70% from the time a manual grading method takes.

3.8 System Deployment Methodology

Deployment Plan

Pilot Deployment: The system was deployed in one department, say, Computer Science, in order to test its functioning in a controlled environment.

University-wide Rollout: After successful testing in the pilot deployment, AMS deployed across all departments.

Training and Support: Workshops were conducted to train faculty in the use of the system. The IT helpdesk was dedicated to troubleshooting support.

Challenges Addressed

Resistance to Change: Some faculty members were skeptical whether the system would be able to replicate human judgment. Training was provided along with manual override in grading.

Infrastructure Requirements: Ensured the system could operate on existing university IT infrastructure with minimal upgrades.

3.9 Chapter Summary

This chapter outlined the detailed methodologies used in the development of the AMS, from literature review to system deployment. Each step was planned and executed to ensure that it aligned with the objectives. The methodologies will make the system reliable and scalable, adding value to the needs of Multimedia University. The next chapter will present the results regarding the performance of the AMS.

References

Baker, J. (2020). Bias in Educational Assessment: Challenges and Solutions. Education Research Review, 12(3), 45-58.

Buchanan, E., Taylor, P., & Zhang, L. (2021). AI in Education: Applications and Impacts. AI Journal, 25(4), 101-125.

Huang, L., Zhou, X., & Wang, M. (2019). Automated Grading Systems: A Comprehensive Review. Journal of Educational Technology, 18(2), 67-89.

Sharma, P., & Dey, S. (2022). Natural Language Processing for Automated Essay Scoring. Computational Linguistics Quarterly, 14(1), 22-38.