QURAN MEMORIZATION PLAN GENERATION USING ARTIFICIAL INTELLIGENCE

AHMAD OSAMA KENTAR

MOHAMMAD YAHYA BATHEEB

ABDULKAREEM YAHYA ALMAHLAWI

DEPARTMENT OF COMPUTER SCIENCE

KING ABDULAZIZ UNIVERSITY

DECEMBER 2024

QURAN MEMORIZATION PLAN GENERATION USING ARTIFICIAL INTELLIGENCE

AHMAD OSAMA KENTAR

MOHAMMAD YAHYA BATHEEB

ABDULKAREEM YAHYA ALMAHLAWI

THIS REPORT IS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE BACHELOR DEGREE IN COMPUTER SCIENCE

DR. ABDULLAH MARISH ALI

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF COMPUTERS INFORMATION TECHNOLOGY

KING ABDULAZIZ UNIVERSITY

DECEMBER 2024

Declaration of Originality

We hereby declare that this project report is based on our original work except for citations and quotations, which had been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at KAU or other institutions.

Student Name	ID	Signature	Date
Ahmad Osama Kentar	2148843		December 2024
Mohammad Yahya Batheeb	2148844		December 2024
Abdulkareem Yahya Almahlawi	2148850		December 2024

Abstract

The project focuses on developing an AI-based solution to improve the Quran memorization process through personalized and adaptive learning plans. Traditional memorization methods often rely on fixed schedules, which lack in addressing individual differences in students' learning speeds, retention abilities, and availability. This inflexibility can lead to discouragement or disengagement among students. By leveraging machine learning, particularly Long Short-Term Memory (LSTM) networks and Reinforcement Learning (RL), the proposed system creates customized memorization, and revision plans that adapt dynamically to each student's performance and needs. The methodology includes analyzing historical data on student recitations and real-time performance metrics to train the AI model. LSTM networks are employed to understand sequential learning patterns and predict appropriate memorization goals, while RL fine-tunes these goals based on ongoing feedback, promoting flexibility and retention. The data-driven approach incorporates student demographics, recitation history, and session-specific metrics to ensure accurate and effective plan generation. Key tools used in the project include Python for model implementation, TensorFlow for machine learning, and databases such as MySQL for data storage and retrieval. Performance metrics such as accuracy and error rates are expected to guide the model's continuous adaptation, ensuring alignment with students' evolving needs. The AI-based system is expected to address the limitations of traditional Quran memorization methods, improving learning efficiency and retention ability. The conclusions highlight the potential benefits of employing AI-powered educational tools to support Quranic learning, aiming to use AI for improving this process and paving the way for future applications that may contribute to its advancement.

المستخلص

يركز هذا المشروع على تطوير حل قائم على الذكاء الاصطناعي لتطوير عملية حفظ القرآن الكريم من خلال خطط تعلم مخصصة وقابلة للتكيف. تعتمد طرق الحفظ التقليدية غالبًا على جداول ثابتة لا تأخذ في الاعتبار الفروقات الفردية في سرعات التعلم وقدرات الاحتفاظ ووقت التفرغ لدى الطلاب. يؤدي هذا النقص في المرونة إلى الإحباط أو فقدان الدافع لدى العديد من الطلاب. باستخدام تقنيات التعلم الآلي، وخاصة الشبكات العصبية طويلة المدى (LSTM) والتعلم المعزز (RL)، يعمل النظام المقترح على إنشاء خطط مخصصة للحفظ والمراجعة تتكيف ديناميكيًا مع أداء الطلاب واحتياجاتهم. تشمل المنهجية تحليل البيانات السابقة لتسميع الطلاب ومقاييس الأداء في الوقت الفعلي لتدريب نموذج الذكاء الاصطناعي. يتم استخدام الشبكات العصبية طويلة المدى (LSTM) لفهم أنماط التعلم المتسلسلة وتوقع أهداف الحفظ المناسبة، في حين يقوم التعلم المعزز (RL) بضبط هذه الأهداف بناءً على التغذية الراجعة المستمرة، مما يعزز المرونة والاحتفاظ بالمعلومات. تعتمد هذه المنهجية على البيانات التي تشمل معلومات الطلاب الأساسية، وسجلات التسميع، والمقاييس الخاصة بكل تسميع لضمان دقة وفعالية الخطط المنتجة. تشمل الأدوات المستخدمة في المشروع Python لتطبيق النموذج، وTensorFlow للتعلم الآلي، وقواعد بيانات مثل MySQL لتخزين البيانات واسترجاعها. يتم توجيه التعديلات المستمرة للنموذج باستخدام مقاييس الأداء مثل الدقة ومعدلات الأخطاء لضمان توافق الخطط مع احتياجات الطلاب المتغيرة باستمرار. نتوقع أن يعالج النظام القائم على الذكاء الاصطناعي المحدودية في الطرق التقليدية لحفظ القرآن الكريم، مما يعزز كفاءة التعلم وقدرة الاحتفاظ. تؤكد الاستنتاجات إلى الفوائد المحتملة لاستخدام الأدوات التعليمية المعتمدة على الذكاء الاصطناعي في دعم تعليم القرآن الكريم. يهدف هذا المشروع إلى توظيف الذكاء الاصطناعي لتحسين عملية تعليم القران الكريم، مما يفتح المجال لتطبيقات مستقبلية قد تسهم في تطوير هذه العملية.

Contents

1. INTRODUCTION 1
1.1. Introduction 1
1.2. Aim 1
1.3. Problem Definition 1
1.4. Target Users 3
1.5. Suggested Solution 3
1.5.1. Key Features of the Suggested Solution: 4
1.6. Initial System Overview 4
1.6.1. Objective 4
1.6.2. Key Functionalities 5
1.6.3. Input Data 5
1.6.4. Output 6
1.6.5. Development Activities 6
1.6.5.1. Requirements Specification 6
1.6.5.2. Data Collection & Preprocessing 6
1.6.5.3. AI Model Design and Training 7
1.6.5.4. Model Testing and Validation 7
1.6.5.5. Real-Time Plan Adjustment Implementation 7
1.6.5.6. Continuous Learning & Improvement 7
1.7. Project Scheduling 8
2. LITERATURE REVIEW 10
2.1. Introduction 10
2.2. Gaps in Quran Memorization Technologies 10

2.3. Personalized Learning: The Role of AI in Education 11
2.4. Adaptive Learning Systems and Real-Time Feedback 12
2.5. The Importance of Revision in Quran Memorization 12
2.6. Conclusion 13
3. DATA COLLECTION AND ANALYSIS TECHNIQUES 14
3.1. Methods to Find Information 14
3.1.1. Research 14
3.1.2. Interviews with subject matter experts 14
3.2. Requirement Specification 15
3.2.1. Functional Requirements 15
3.2.2. Non-Functional Requirements 16
3.2.3. Data Requirements 17
3.2.4. Software Requirements 17
3.2.5. Hardware Requirements 18
3.2.6. Security Requirements 19
3.3. Acquiring Data 19
3.4. Dataset Overview 20
3.4.1. Student Lesson History 20
3.4.2. Surahs 21
3.4.3. Verses 22
3.5. Dataset Utilization in the Project 23
4. ARCHITECTURE AND DESIGN 25
4.1. Machine Learning Approach 25
4.2. What are LSTM and RL? 25
4.2.1. Long Short-Term Memory (LSTM) 25

4.2.2. Reinforcement Learning (RL) 26
4.3. Why use LSTM and RL in our project? 26
4.3.1. Long Short-Term Memory (LSTM) 26
4.3.2. Reinforcement Learning (RL) 27
4.4. Key Model Inputs 28
4.5. Data Processing Workflow 28
4.6. Summary of the Approach 29
4.7. Needed Skills 29
4.7.1. Main Skills Needed 29
4.7.1.1. Machine Learning and Deep Learning 29
4.7.1.2. Data Processing and Analysis 30
4.7.1.3. Database Management and SQL 30
4.7.1.4. Python Programming 30
4.7.2. Skills to Be Developed 31
4.7.2.1. Long Short-Term Memory (LSTM) 31
4.7.2.2. Reinforcement Learning (RL) 31
4.7.2.3. Data Processing and Feature Engineering 31
4.7.2.4. Backend Development 31
4.8. The Initial Design 32
4.8.1. Use Case Diagram 32
4.8.2. ER Diagram 34
4.8.3. Data Flow Diagram 36
4.8.4. Flowchart 38
4.8.5. Sequence Diagram 41
4.8.5.1. Add New Student 41

4.8.5.2. Request Initial Plan 42
4.8.5.3. Record Recitation Feedback 43
4.8.5.4. Pause or Modify an Ongoing Plan 44
4.8.6. Prototype Design 45
4.8.6.1. Students' Plans 45
4.8.6.2. Add New Student 46
4.8.6.3. Student's New Plan 47
4.8.6.4. Create a New Plan 48
4.8.6.5. View Student's Plans 50
5. Conclusion 51
6. References 53

List Of Figures

Figure 1: Student Lesson History Table 21
Figure 2: Surah Table 22
Figure 3: Verses Table 23
Figure 4: Use Case Diagram 33
Figure 5: ER Diagram 35
Figure 6: Data Flow Diagram 36
Figure 7: Flowchart (Part 1) 38
Figure 8: Flowchart (Part 2) 39
Figure 9: Sequence Diagram - Add New Student 41
Figure 10: Sequence Diagram - Request Initial Plan 42
Figure 11: Sequence Diagram - Record Recitation Feedback 43
Figure 12: Sequence Diagram - Pause or Modify an Ongoing Plan 44
Figure 13: UI - Students' Plans ....خطأ! الإشارة المرجعية غير معرّفة.
Figure 14: UI - Add New Student ....خطأ! الإشارة المرجعية غير معرّفة.
Figure 15: UI - Student's New Plan ....خطأ! الإشارة المرجعية غير معرّفة.
Figure 16: UI - Create a New Plan ....خطأ! الإشارة المرجعية غير معرّفة.
Figure 17: UI - View Student's Plans ....خطأ! الإشارة المرجعية غير معرّفة.

List Of Tables

Table 1: Project Scheduling 9

CHAPTER 1

INTRODUCTION

1.1. Introduction

In many traditional Quran learning environments, students are expected to follow a fixed memorization schedule, which doesn’t account for their unique abilities. As a result, some students may struggle to keep up, while others aren’t being challenged enough. This project explores how AI can offer a more personalized approach to Quran memorization, improving the learning experience for each student.

1.2. Aim

This project aims to propose and design a machine-learning model that provides personalized and adaptive Quran memorization plans for individual students. Using artificial intelligence, the system will analyze user-specific data, such as current memorization level and student performance, to create an efficient, manageable, and flexible memorization plan. The goal is to enhance the students’ progress through optimized scheduling, making memorization more structured, personalized, and achievable.

1.3. Problem Definition

Memorizing the Quran is a cherished tradition in Islam, with countless Muslims dedicating themselves to this sacred pursuit. However, in today's world, memorizing the Quran presents significant challenges for students. Most Quran memorization programs follow fixed schedules that do not account for individual differences in learning abilities. As a result, students with varying memorization speeds, cognitive abilities, and learning preferences are forced to follow a standardized approach that may not suit their unique needs.

In this traditional approach, students are typically expected to memorize a set portion of the Quran each day and review previously memorized sections periodically. While this approach may work for some, many students face difficulties keeping up with the assigned pace. Some struggle due to personal factors such as limited time availability, weaker memorization ability, or slower memorization speed. These students may fall behind, becoming discouraged and less motivated, which can ultimately affect their progress and confidence. On the other hand, students who can memorize quickly and efficiently often find themselves under-challenged, leading to disengagement and limited learning. Both scenarios highlight the inefficiencies in the fixed memorization plans.

Furthermore, one of the key challenges in Quran memorization is retention. Even if students manage to memorize new material, retaining it over time requires regular revision. Traditional systems tend to implement fixed revision schedules that do not consider individual retention capabilities. Without an adaptive mechanism to adjust revision plans based on each student's needs, students may either forget previously memorized sections due to inadequate review or spend too much time revising verses they have already mastered, limiting their overall progress.

The absence of intelligent feedback systems makes the lack of personalization in Quran memorization programs worse. Teachers often rely on their judgment to assess a student's progress, which can be subjective and inconsistent. There's no structured way to adjust memorization plans based on a student's ongoing performance, leaving many students frustrated and stuck in ineffective learning.

Given these challenges, there is a growing need for an intelligent system to offer a more personalized approach to Quran memorization. Artificial intelligence and machine learning technologies have the potential to transform this process by analyzing individual learning patterns and generating tailored memorization plans. These technologies can assess a student's current level, pace, and performance, adjusting the memorization and revision schedules dynamically to ensure that each student receives the appropriate level of challenge and support. Such a system would address the issue of varying memorization speeds and enhance long-term retention through customized revision strategies.

1.4. Target Users

1. **Students of Quran Memorization**: Individuals of all ages and backgrounds aiming to memorize the Quran, whether beginners, intermediates, or advanced learners. These students may vary in memorization speed, learning ability, and daily availability, making personalized plans critical for effective progress.

2. **Quran Teachers/Instructors**: Educators who supervise students' Quran memorization can use the system to track student progress, get performance insights, and adjust learning strategies for each student more effectively.

3. **Islamic Educational Institutions**: Islamic schools and online Quran learning platforms that want to offer their students a modern and personalized Quran memorization tool, improving overall learning outcomes.

4. **Parents/Guardians**: Those responsible for guiding their children through Quran memorization, seeking a tool that provides structured, personalized plans to ensure effective learning and retention.

1.5. Suggested Solution

The proposed solution is an **AI-based Quran Memorization Plan Generator** that designs personalized memorization and revision plans based on a student's unique learning profile, performance data, and retention ability. The system will analyze user-specific inputs such as current memorization level, daily availability, and revision needs, to generate and adapt a flexible memorization plan that balances new verses with structured revision.

1.5.1. Key Features of the Suggested Solution:

1. **Personalized Memorization Plans:** Create customized plans for each student, considering their pace, availability, and progress. The system generates new verses to memorize, alongside minor and major revisions based on what the student has already memorized.
2. **Performance-Based Adjustments:** The AI model adjusts memorization, and revision plans automatically based on the student's performance, including speed, accuracy, and retention rates. The system can scale up or slow down the workload to match the students' capabilities.
3. **Adaptive Revision Plans:** The system ensures that students regularly review previously memorized verses, using personalized intervals for revision. Minor revisions cover recently memorized verses, while major revisions periodically improve the retention of large parts of previously memorized sections.
4. **Flexible and Dynamic Scheduling:** The system adapts to changes in the student's availability or challenges, allowing them to pause, adjust, or modify their memorization plan without losing progress.

1.6. Initial System Overview

The project focuses on developing an AI model that generates personalized Quran memorization plans based on a student's performance data. The model will operate without a complex system or UI, but with a simple interface to input performance data and output the next memorization plan. Below is a detailed overview of the project:

1.6.1. Objective

The AI model will create three types of plans:

- **New Memorization Plan**: Assign new Quranic verses for students to memorize.

- **Minor Revision Plan**: Focuses on revising recently memorized portions.
- **Major Revision Plan**: Periodic review of larger portions of memorized Quran to ensure long-term retention.

1.6.2. Key Functionalities

The AI model will have the following functionalities:

1. **Initial Training on Historical Data**: Using a dataset that contains records of student recitations, the model will learn patterns from past performances.
2. **Real-Time Plan Adaptation**: After each student's recitation, the model will analyze performance data (e.g., mistakes made, and whether the recitation was accepted as valid) to adjust future memorization and revision plans.
3. **Dynamic Plan Generation**: The model continuously refines its predictions, generating new, minor, and major revision plans based on ongoing performance tracking.

1.6.3. Input Data

- **Training Dataset**: Contains recitation records of students, with details such as:
  - Portion requested for recitation.
  - Number of mistakes made.
  - Whether the recitation was accepted as valid.
- **Real-Time Performance Data**: Captured after each recitation session, this data will be used to modify the student's plans. It includes:
  - Performance on the most recent recitation.
  - Mistake count.
  - Feedback on the validity of the recitation.

1.6.4. Output

- **Memorization Plan**: Assign new Quranic verses based on the student's capacity.
- **Minor Revision Plan**: Suggests portions of recently memorized content for revision to reinforce short-term memory.
- **Major Revision Plan**: Periodically suggests larger sections for review to support long-term retention.

1.6.5. Development Activities

1.6.5.1. Requirements Specification

- Identify key metrics for evaluating student performance, such as number of mistakes, pace of recitation, and memorization capacity.
- Define how often plans should be adjusted based on recitation feedback.

1.6.5.2. Data Collection & Preprocessing

- Data Collection: Collect historical data on student recitations, including mistakes and feedback.
- Data Preprocessing: Clean and structure the dataset to ensure consistent input for model training, including handling missing data, normalizing performance metrics, and organizing data by individual student records.

1.6.5.3. AI Model Design and Training

- Model Selection: Choose an appropriate machine learning model (e.g., decision tree, neural network) capable of handling sequential data for learning progression and retention patterns.
- Training: Use the preprocessed dataset to train the AI model. The model will learn from past performance data, identifying trends in student memorization speed, error rates, and retention.

1.6.5.4. Model Testing and Validation

- Testing: Validate the AI model using unseen data to ensure it provides accurate memorization plans and adjusts based on student performance.
- Evaluation Metrics: Use evaluation metrics such as accuracy in predicting next verses and F1-score for mistake prediction to ensure robustness.

1.6.5.5. Real-Time Plan Adjustment Implementation

- Integration: Incorporate a feedback loop where new student performance data is fed into the model after each recitation, allowing real-time updates to memorization and revision plans.
- Performance Monitoring: Continuously monitor the model's predictions to ensure the plans are effectively adapted to each student’s learning pace.

1.6.5.6. Continuous Learning & Improvement

- Use reinforcement learning or a similar approach to enable the AI model to learn and improve its predictions over time based on feedback from real-time recitations and evaluations.

1.7. Project Scheduling

Waterfall Phase	Milestones	Tasks	Week
Analysis	Initial Project Proposal	Finalize project title, objectives, and scope	Week 3
		Prepare project significance overview	
		Conduct preliminary research	Week 4
		Gather sources for the literature review	
		Prepare slides for initial presentation	Week 5
		Draft and refine the project proposal	
	Project Proposal and Literature Review	Begin writing Report #1	Week 6
		Research existing Quran memorization tools and AI applications	
		Conduct literature review	Week 7
	Initial Requirements Analysis	Conduct initial requirements analysis	Week 8
		Identify user needs and technical requirements	
		Finishing writing Report #1	
	Presentation #1 Preparation	Prepare slides for Presentation #1	Week 9
Design	Detailed Requirements and System Design	Conduct detailed requirements analysis	Week 10
		Start data collection and description	
		Begin writing Report #2	Week 11
		Begin system design and architecture	
		Develop initial prototypes	

	Report#2 and Presentation#2 Submission	Finishing writing Report #2	Week 12
		Prepare slides for Presentation #2	
	Final Report Submission	Compile and finalize all chapters for the final report	Week 13
		Ensure system design and prototyping documentation is complete	
		Review and refine the final report for submission	
		Complete all necessary revisions	
		Final Report (All Chapters) Submission	Week 14
	Poster Design and Preparation	Design the project poster	
		Finalize the key points and visuals for the poster presentation	Week 15
	Final Presentation Preparation	Prepare the slides for the final presentation	
		Review all phases of the project to present key findings, challenges, and outcomes	

Table 1: Project Scheduling

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

Integrating artificial intelligence (AI) into personalized learning environments has become a transformative approach in modern education. AI systems provide dynamic, real-time adaptation of learning plans to individual needs, making learning more efficient and targeted. In Quran memorization, an area that requires continuous revision and personalized pacing, traditional methods often fall short in addressing the unique requirements of learners. This literature review explores themes around the gaps in existing Quran memorization technologies, personalized learning, and adaptive AI-driven educational models, positioning them within the scope of developing an AI-based Quran Memorization Plan Generator.

2.2. Gaps in Quran Memorization Technologies

Despite the increasing use of technology in Quran memorization, significant gaps remain, particularly in personalized, adaptive systems. Most current tools offer static memorization schedules or simple tracking features, lacking the ability to adjust to the individual needs of each learner.

Haryono et al. (2023)[1] provide a comprehensive review of Quran memorization technologies and highlight the absence of AI-driven tools that offer real-time adaptation and personalization. Existing systems are often limited to tracking progress and providing access to Quranic text but fail to dynamically adjust to the learner's memorization pace or offer tailored revision schedules. This gap points to the need for more intelligent systems to analyze student data and adjust the learning path accordingly.

The proposed AI-based Quran Memorization Plan Generator would fill this gap by offering a more personalized and adaptive approach. Using AI to create dynamic, flexible plans based on individual performance would make Quran memorization more accessible and effective, especially for students with varying memorization and retention abilities. By addressing the limitations identified by **Haryono et al. (2023)**[1], the project aims to revolutionize how students approach Quran memorization, making the process more structured and personalized.

2.3. Personalized Learning: The Role of AI in Education

Traditional teaching methods often follow strict, standardized approaches that may not consider individual student differences. However recent advancements in AI offer solutions that can dynamically adjust to each learner's pace, abilities, and progress.

Maghsudi et al. (2021)[2] discuss how AI can provide personalized learning experiences by analyzing student data to deliver tailored content and feedback. This aligns with the needs of Quran memorization, where learners progress at different rates, and fixed memorization schedules can hinder learning. In Quranic studies, an AI-driven system can continuously monitor a student's progress, adjusting both the number of new verses assigned and the frequency of revision sessions to optimize learning outcomes. Similarly, **Anil et al. (2019)**[3] proposes dynamic learning plan generators that adjust to individual learning profiles, emphasizing how real-time student data can influence personalized learning paths. This dynamic approach, when applied to Quran memorization, could enable learners to move at their own pace while ensuring consistent review of previously memorized content.

The significance of personalized learning systems in Quran memorization is that they break away from one-size-fits-all approaches, offering a more flexible, student-centered model. This would allow each student to follow a plan that matches their memorization ability, ensuring both progression and retention, key challenges in Quran learning.

2.4. Adaptive Learning Systems and Real-Time Feedback

The concept of adaptive learning systems builds on personalization by adding continuous, real-time adaptation based on performance. In Quran memorization, this is particularly important because learners often need to balance learning new material with revisiting previously memorized content to strengthen retention. AI's capability to dynamically adjust learning paths in real time is a key enabler of more effective Quran memorization practices.

Somasundaram et al. (2020)[4] introduce an AI-based system for managing learning paths, where student performance is tracked, and plans are adjusted according to real-time feedback. This continuous adaptation is especially relevant to Quran memorization, where an AI system could monitor how well students are retaining verses and suggest revision sessions when needed.

Oussama et al. (2022)[5] also emphasize the importance of adaptability in AI-driven education systems, suggesting that personalized learning paths should evolve based on ongoing assessments of student progress.

For Quran memorization, a similar model could be developed where AI tracks factors such as recitation accuracy, speed of memorization, and long-term retention. This would ensure that memorization plans remain responsive to the learner's needs, providing a flexible and sustainable learning journey. By automatically adjusting revision intervals and assigning new verses when appropriate, AI systems can help prevent learners from becoming overwhelmed, thus improving memorization effectiveness.

2.5. The Importance of Revision in Quran Memorization

A significant challenge in Quran memorization is retention, and effective revision strategies are essential for long-term success. Research into AI-based learning systems shows that spaced repetition and revision are critical components of adaptive learning models, ensuring that learners retain what they have memorized.

Zhang et al. (2020)[6] focus on the implementation of personalized learning systems and highlight the importance of structured revision to reinforce learning. In the Quran memorization, the need for a consistent review of previously learned material cannot be overstated. An AI-based system can be designed to incorporate these findings by scheduling both minor and major revisions based on student performance data. The system would ensure that learners are regularly revisiting past lessons to reinforce their memorization, preventing long-term forgetting. By combining new memorization with periodic review, the AI-based system could effectively support both short-term learning goals and long-term retention.

This combination of new learning and targeted revision allows students to maintain a steady pace of progress while ensuring that older content is regularly revisited. It addresses a key issue in Quran memorization—balancing the need for continuous progress with the requirement for periodic review to reinforce long-term memory.

2.6. Conclusion

The literature on AI in education consistently highlights the potential of personalized, adaptive learning systems to enhance student outcomes by tailoring the learning experience to individual needs. In the context of Quran memorization, where learners require both flexibility and structure, the development of an AI-based memorization plan generator addresses several key challenges. The studies reviewed provide a strong theoretical foundation for applying AI to personalize learning, optimize revision schedules, and improve retention. By incorporating real-time performance feedback and adaptive scheduling, this project can significantly improve the efficiency and effectiveness of Quran memorization, bridging the gap identified in current Quran memorization technologies.

CHAPTER 3

DATA COLLECTION AND ANALYSIS TECHNIQUES

3.1. Methods to Find Information

To gather the comprehensive information required for our project, we employed a multifaceted approach that combined research with insightful interviews with subject matter experts from a Quranic education management company and teachers of the Quran.

3.1.1. Research

To collect system requirements, we utilized research as a foundational method for identifying functional, non-functional, hardware, and software needs. Research allowed us to define functional requirements by studying the key features needed for personalized Quranic memorization, such as tracking student progress, creating adaptive memorization plans, and supporting specific learning methodologies. It also provided insights into non-functional requirements, helping us set clear standards for system performance, scalability, usability, and data security.

Additionally, research guided us in understanding the hardware requirements, such as the computational resources necessary to train and deploy our models effectively, as well as the storage needed for handling large datasets. On the software side, it helped us determine the tools, frameworks, and platforms best suited for developing and deploying the system efficiently. By systematically analyzing available information, research enabled us to establish a comprehensive and realistic set of system requirements aligned with the project's goals.

3.1.2. Interviews with subject matter experts

Interviews with subject matter experts played a crucial role in gathering comprehensive system requirements, including functional and non-functional aspects. These discussions provided practical, experience-driven insights that were instrumental in defining the system's functional requirements. Experts highlighted key features necessary for effective Quranic memorization, such as personalized learning plans, real-time feedback on recitations, and tools for tracking student progress. Their input also clarified the importance of incorporating features that address common challenges, like frequent errors in pronunciation or retention, ensuring the system aligns closely with real-world teaching practices.

For non-functional requirements, experts emphasized the need for a user-friendly interface suitable for both teachers and students, as well as the importance of system reliability and accuracy in evaluating recitations. They also provided feedback on the system's response times, scalability for handling multiple users and maintaining data privacy, especially for student performance records.

3.2. Requirement Specification

3.2.1. Functional Requirements

The functional requirements of the model are designed to support adaptive and personalized learning experiences for each student. The main functional requirements include:

1. **Generate Daily Memorization Plans**:
    - The model should generate a personalized daily plan for memorization and revision for each student based on his historical performance and current needs.
2. **Adjust Plans in Real-Time**:
    - The system should dynamically adjust the memorization and revision goals based on the student's last recitation session, including errors and accuracy. Real-time adaptation is achieved through Reinforcement Learning (RL).

3. **Track and Record Performance Data**:
   - The system should record the data of each recitation session, including the verses recited, errors, and the type of recitation (new memorization, major revision, minor revision).
4. **Historical and Cluster-Based Analysis**:
   - For new students for which there is no personal historical data, the model should use performance data from similar students to determine a suitable starting goal.
5. **Provide Performance Feedback**:
   - Provide feedback to teachers based on errors and accuracy and guide them to areas that need to be revised or improved.

3.2.2. Non-Functional Requirements

These requirements ensure the usability, performance, and reliability of the system :

1. **Usability**:
   - The model should have a simple user interface which will make it easier for teachers to enter student data and retrieve daily plans.
2. **Performance**:
   - Adjustments should occur in real-time within seconds to ensure a smooth and responsive operation process.
3. **Scalability**:
   - The model should be able to handle several students at once, which will allow the system to grow as the number of users increases.
4. **Reliability**:
   - The system should work constantly while ensuring data integrity and accuracy in generating plans and tracking performance.

5. **Maintainability**:
    - The architecture of the system should be modular and easy to update, which allows it to adjust and improve efficiently.

3.2.3. Data Requirements

The dataset needs to capture detailed records of each student's recitation history and real-time performance data:

1. **Student Data**:
    - Basic information like student ID, name, and memorization start date.
    - Historical recitation data, including verses recited and type of recitation.
2. **Session Performance Data**:
    - Data on recitation sessions, such as date, type, errors, acceptance, and pace (letters and pages recited).
3. **Historical and Cluster-Based Data**:
    - Aggregated performance data from similar students, organized by clusters, for new sections with no specific student history.
4. **Generated Feedback Data**:
    - Referring to the data that captures any feedback or adjustments made to a student's daily memorization and revision plan based on their recent recitation performance. This feedback includes information on how the system responded to the student's session—whether the target was adjusted, the type of adjustment made, and the reasons behind it.

3.2.4. Software Requirements

The project requires a combination of machine learning libraries, data management tools, and programming environments to develop, deploy, and maintain the system.

1. **Programming Languages**:
    - **Python**: Primary language for model development, data processing, and integration.
2. **Machine Learning Libraries**:
    - **TensorFlow or PyTorch**: For building the machine learning model.
    - **scikit-learn**: For clustering similar students based on performance metrics.
3. **Database Management**:
    - **MySQL or PostgreSQL**: To store and retrieve student records, session logs, and performance metrics.
4. **Data Processing Libraries**:
    - **Pandas, NumPy**: For data cleaning, manipulation, and analysis.
5. **Environmental and Deployment Tools**:
    - **Jupyter Notebook**: For model development and testing.
    - **Docker**: For containerizing the application for easy deployment and scalability.

3.2.5. Hardware Requirements

1. **Server Requirements**:
    - **CPU**: Multi-core processors to handle model training and data processing.
    - **GPU**: Required for efficient training of the models as we are working with a large dataset.
    - **RAM**: Minimum 16GB to support data handling, model training, and database operations.

2. **Local System Requirements**:
    - Personal computers for development with at least 8GB RAM and a modern CPU.
3. **Cloud Infrastructure (Optional)**:
    - **AWS** or **Google Cloud Platform**: For scalable deployment and storage of large datasets, with GPU capabilities for model training.

3.2.6. Security Requirements

Security is essential to protect sensitive student data and ensure the system's integrity. Key security requirements include:

1. **Access Control**:
    - Role-based access control (RBAC) to limit data access based on user roles, allowing only authorized teachers and administrators to access or update student records.
2. **Data Backup and Recovery**:
    - Regular backup of the database to prevent data loss, with a recovery plan in case of accidental data deletion or corruption.
3. **Authentication**:
    - Implement secure login and authentication methods for users to access the system, such as two-factor authentication (2FA) for added protection.

3.3. Acquiring Data

To collect the data, we needed for our project, we teamed up with a company that specializes in managing Quranic education. This company's system keeps detailed records of student recitations, including their performance, memorization progress, and revision habits. By working with them, we got access to this organized data, which became the backbone of our personalized Quran memorization model.

We started our data-gathering journey by chatting with the company's team. These conversations gave us valuable insights into how their system tracks and evaluates student performance, manages recitation logs, and organizes memorization and revision cycles. The information they shared helped us understand the real-world aspects of Quranic education, which was crucial for identifying relevant data points and grasping the context of recitation records. This knowledge shaped our approach, ensuring that our data collection process aligned with the specific needs of our model.

While we also searched online for publicly available data on student recitation patterns and memorization processes, we found limited resources that fit our project's unique requirements. Most online datasets either lacked the necessary detail or weren't specifically focused on Quranic memorization. As a result, our collaboration with the Quranic education company became essential, providing us with both the required data and an understanding of its practical application in educational settings. This partnership not only enriched our dataset but also guided our methodology, ensuring that our project remained closely connected to real-world Quranic memorization practices.

3.4. Dataset Overview

The dataset contains three main tables that provide a complete view of each student's journey in memorizing and revising the Quran. These tables are:

3.4.1. Student Lesson History

It tracks individual recitation sessions for each student and explains the specific verses recited, the type of recitation, and performance indicators. The main columns include:

- **student_id**: Defines each student uniquely.
- **start_verse_id** and **end_verse_id**: Determines the range of verses recited in each session.
- **pillar_id**: Indicates the type of recitation (for example, 1 for new memorization, 2 for major revision, 3 for minor revision). This column is

useful for classifying types of recitation, which helps the model distinguish between new memorization sessions and revision.

- **teacher_id**: Defines each teacher uniquely.
- **date_of**: Date of the recitation session.
- **letters_count** and **pages_count**: Describes the amount of recitation based on the range of verses recited in each session. These columns are important for tracking the amount of conservation.

B	C	D	E	F	G	H	I
student_id	start_verse_id	end_verse_id	pillar_id	teacher_id	date_of	calculated_letters_count	calculated_pages_count
424	6231	6225	3	422	8/20/2023	257	1.000001
452	6080	5993	3	449	8/21/2023	2154	4.590018
479	6231	5932	3	449	8/22/2023	5580	13.455782
334	3971	3793	2	523	8/23/2023	3187	5.413072
331	5623	5640	2	522	8/24/2023	263	0.516698
331	5623	5640	2	83	8/24/2023	263	0.516698
331	6231	6234	2	523	8/24/2023	66	0.256809
332	6231	6226	2	522	8/24/2023	132	0.513619
332	5673	5702	3	522	8/24/2023	519	1.000003

Figure 1: Student Lesson History Table

3.4.2. Surahs

It stores constant information about each Surah (chapter), including the number of verses. The main columns include:

- **surah_id**: A unique identifier for each Surah.
- **name**: The name of the Surah in the Quran.
- **no_verses**: The total number of verses in each Surah.

A	B	C
surah_id	name	no_verses
1	الفاتحة	7
2	البقره	286
3	آل عمران	200
4	النساء	176
5	المائده	120
6	الأنعام	165
7	الأعراف	206
8	الأنفال	75
9	التوبة	129
10	يونس	109
11	هود	123
12	يوسف	111
13	الرعد	43
14	إبراهيم	52
15	الحجر	99
16	النحل	128
17	الإسراء	111
18	الكهف	110
19	مريم	98
20	طه	135
21	الأنبياء	112
22	الحج	78

Figure 2: Surah Table

3.4.3. Verses

It provides verse-specific details, including the order in the Quran and each Surah, as well as the number of letters and page locations. The main columns include:

- **verse_id**: A unique identifier for each verse.
- **surah_id**: Connects each verse with its own Surah.
- **order_in_quraan**: The sequential order of the verse in the Quran.

- **reverse_index**: The reverse order of verses.
- **order_in_surah**: The sequential order of the verse in the Surah.
- **page_no**: The location of the page where the verse is located.
- **letters_count**: The total number of letters in each verse, used to analyze the speed of memorization.
- page_no, letters_count, and weight_on_page: Details on where the verse is located, its length, and its weight on the page.

A	B	C	D	E	F	G	H	I
verse_id	surah_id	begin_verse	order_in_quraan	reverse_index	order_in_surah	page_no	letters_count	weight_on_page
1	1	بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ	1	6230	1	1	19	0.132867
2	1	ٱلۡحَمۡدُ لِلَّهِ رَبِّ	2	6231	2	1	18	0.125874
3	1	ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ	3	6232	3	1	12	0.083916
4	1	مَٰلِكِ يَوۡمِ ٱلدِّينِ	4	6233	4	1	12	0.083916
5	1	إِيَّاكَ نَعۡبُدُ وَإِيَّاكَ	5	6234	5	1	19	0.132867
6	1	ٱهۡدِنَا ٱلصِّرَٰطَ ٱلۡمُسۡتَقِيمَ	6	6235	6	1	19	0.132867
7	1	صِرَٰطَ ٱلَّذِينَ أَنۡعَمۡتَ	7	6236	7	1	44	0.307692
8	2	بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ	8	5944	1	2	22	0.121547
9	2	ذَٰلِكَ ٱلۡكِتَٰبُ لَا	9	5945	2	2	27	0.149171
10	2	ٱلَّذِينَ يُؤۡمِنُونَ بِٱلۡغَيۡبِ	10	5946	3	2	47	0.259669
11	2	وَٱلَّذِينَ يُؤۡمِنُونَ بِمَآ	11	5947	4	2	52	0.287293
12	2	أُوْلَٰٓئِكَ عَلَىٰ هُدٗى	12	5948	5	2	33	0.18232
13	2	إِنَّ ٱلَّذِينَ كَفَرُواْ	13	5949	6	3	47	0.083929
14	2	خَتَمَ ٱللَّهُ عَلَىٰ	14	5950	7	3	53	0.094643
15	2	وَمِنَ ٱلنَّاسِ مَن	15	5951	8	3	47	0.083929
16	2	يُخَٰدِعُونَ ٱللَّهَ وَٱلَّذِينَ	16	5952	9	3	49	0.0875
17	2	فِي قُلُوبِهِم مَّرَضٞ	17	5953	10	3	51	0.091071
18	2	وَإِذَا قِيلَ لَهُمۡ	18	5954	11	3	43	0.076786
19	2	أَلَآ إِنَّهُمۡ هُمُ	19	5955	12	3	29	0.051786
20	2	وَإِذَا قِيلَ لَهُمۡ	20	5956	13	3	77	0.1375
21	2	وَإِذَا لَقُواْ ٱلَّذِينَ	21	5957	14	3	73	0.130357
22	2	ٱللَّهُ يَسۡتَهۡزِئُ بِهِمۡ	22	5958	15	3	34	0.060714
23	2	أُوْلَٰٓئِكَ ٱلَّذِينَ ٱشۡتَرَوُاْ	23	5959	16	3	57	0.101786
24	2	مَثَلُهُمۡ كَمَثَلِ ٱلَّذِي	24	5960	17	4	72	0.125217
25	2	صُمُّۢ بُكۡمٌ عُمۡيٞ	25	5961	18	4	19	0.033043
26	2	أَوۡ كَصَيِّبٖ مِّنَ	26	5962	19	4	86	0.149565
27	2	يَكَادُ ٱلۡبَرۡقُ يَخۡطَفُ	27	5963	20	4	102	0.177391
28	2	يَٰٓأَيُّهَا ٱلنَّاسُ ٱعۡبُدُواْ	28	5964	21	4	53	0.092174
29	2	ٱلَّذِي جَعَلَ لَكُمُ	29	5965	22	4	99	0.172174
30	2	وَإِن كُنتُمۡ فِي	30	5966	23	4	78	0.135652

Figure 3: Verses Table

3.5. Dataset Utilization in the Project

The detailed structure of the dataset allows it to support a personalized Quran memorization plan by providing:

- **Deriving Recitation Patterns for Model Training**:
  - The dataset provides training examples of common recitation patterns, average memorization speeds, and frequent errors.

    - This data allows the model to generalize typical learning behaviors, making it more accurate in guiding both new and experienced students through personalized recitation plans.

- **Collective Cluster-Based Data**
    - In addition to individual data, the dataset enables the calculation of average recitation pace and performance across groups of students at similar levels.
    - These averages provide a foundation for creating initial plans for new students or those without a personal recitation history in certain surahs. By using typical pace data from students with similar progress patterns, the model can set realistic starting goals even before personalized patterns are established.

This refined structure clarifies that initial plans for students without history rely on collective averages from similar students, while historical data and recitation patterns aid in model training and personalization.

CHAPTER 4

ARCHITECTURE AND DESIGN

4.1. Machine Learning Approach

Memorizing the Quran is a process that requires sequential understanding and the ability to adapt to the progress of each student. To meet these needs, we recommend a hybrid approach that combines Long Short-Term Memory (LSTM) networks and Reinforcement Learning (RL). This combination is ideal because LSTM is excellent at recognizing patterns in sequential data, such as the student's historical memorization performance, while RL allows the model to adapt dynamically to real-time progress. Together, these methods provide a balanced approach that supports continuous growth in memorization and adapts to each student's unique pace and retention needs.

In addition, an algorithm will be used to evaluate the overall performance of students in various recitation parts, providing basic metrics. These metrics will help prepare initial plans for students who do not have previous recitation data, allowing a personal starting point based on average performance. This approach ensures a steady growth in memorization while adapting to the learning path of each student.

4.2. What are LSTM and RL?

4.2.1. Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) designed to handle sequential data and retain dependence on previous information for long periods[7]. This is very useful for tasks that require knowledge of past information to understand or predict future results. The LSTM network relies on memory cells to capture patterns over time, making it suitable for analyzing students' historical memorization data. By identifying pace, retention, and accuracy trends, LSTM helps the model set personalized, data-driven goals that align with the student's learning style and past performance.

4.2.2. Reinforcement Learning (RL)

Reinforcement learning is a type of machine learning in which an agent learns to make decisions by interacting with the environment[8]. The agent receives feedback in the form of rewards or penalties based on its actions, directing it towards achieving a specific goal over time. Through trial and error, the agent learns how to optimize cumulative rewards, which allows it to adapt its behavior to achieve better results. In the context of Quran memorization, RL can adjust daily goals in real-time based on each student's performance, promoting a flexible and responsive learning experience.

4.3. Why use LSTM and RL in our project?

4.3.1. Long Short-Term Memory (LSTM)

The reasons to use Long Short-Term Memory (LSTM) are:

- **Capturing Sequential Dependencies**: Quran memorization is a cumulative process where each recitation session builds on previous sessions. LSTM models can capture these dependencies by processing sequences of past performance data, such as the number of verses recited and pace. This enables the model to learn patterns over time, allowing it to generate a basis for each student's daily memorization and revision goal.

- **Predicting Memorization Trajectories**: By analyzing historical data for each student, the LSTM model can recognize patterns in learning speed, consistency, and areas that require more effort[7]. For example, if a student shows steady progress in memorizing about five verses per session with few mistakes, LSTM can predict that the student may be ready for a small increase in the daily goal. Conversely, if previous data shows repeated challenges in certain surahs or verses, the model can adjust the plan to ensure that those areas are reviewed more thoroughly.

- **Foundation for Baseline Plans**: The LSTM generates a personalized baseline plan for each student, setting daily goals for new memorization and revision based on past patterns. This plan is initial and adaptable, as it is aligned with the unique speed of each student and his retention needs. Moreover, the RL component optimizes the plan so that the model can keep up with the performance in real-time.

4.3.2. Reinforcement Learning (RL)

The reasons to use Reinforcement Learning (RL) are:

- **Real-Time Plan Adjustments**: Unlike LSTM, which generates a basic plan based on historical data, RL enables real-time adjustments. After each recitation session, the RL model reviews performance metrics, such as the number of errors, missed sessions, or student accuracy, and updates the plan accordingly. For example, if a student completes a session with high accuracy and few errors, the RL model may slightly increase the goal for new verses for the next session. Conversely, if accuracy decreases or errors increase, the agent may prioritize additional review sessions.
- **Adaptability to Immediate Performance**: RL is characterized by responding to immediate changes in performance. For example, if a student has not reviewed a particular Surah for a long time, the RL agent can add that Surah to the Daily plan, even if it is not included in the basic plan created by LSTM. This ensures that the student does not lose retention of previously memorized parts while working on new material.
- **Reinforcement Through Rewards and Penalties**: Reinforcement through rewards and punishments: the RL model uses a reward system to guide

adjustments in the daily plan. Positive performance metrics, such as high accuracy and consistency, may lead to rewards such as adding new verses for memorization. Conversely, penalties, such as repeating mistakes, may lead to a reduction in the daily target or the introduction of additional review sessions. This system based on rewards and penalties helps to balance progress and retention, enabling the student to progress without feeling overwhelmed.

4.4. Key Model Inputs

- **LSTM Inputs**: LSTM in this project uses existing columns from the historical dataset, such as letters_count, pages_count, and pillar_id, which represent the amount and type of verses recited in previous sessions.
- **RL Agent Inputs**: The RL agent uses the data generated newly in each recitation session, which is stored in a separate database with additional columns added specifically for tracking performance metrics in real-time.

4.5. Data Processing Workflow

1. **Preprocessing**:
    - Grouping daily recitation data into sequences for each student, enabling the LSTM to learn from past performance patterns.
2. **Model Training**:
    - The LSTM is trained on these sequences to estimate an initial daily memorization and revision plan based on historical trends.
    - For new students with no history, an algorithm calculates average performance metrics from other students to generate initial plans based on common recitation patterns.
3. **RL Agent for Real-Time Adjustment**:

- The RL agent dynamically adjusts the plan:
  - **New Surahs with No History**: Combines general pace with cluster-based averages to set a suitable starting target.
  - **Previously Memorized Surahs**: Integrates the student's general and surah-specific pace for revisiting familiar sections.
- The RL agent improves retention by adjusting the quantities according to recent errors, missed sessions, and time since the last review.

4.6. Summary of the Approach

The combined approach of LSTM and RL, supported by an algorithm that uses average performance metrics, creates a flexible and personalized system for memorizing the Quran. This approach adapts to the historical performance of each student, real-time progress, and data from similar students. By customizing the learning plan with targeted adjustments, the model aligns with the pace of each student, focusing on difficult areas and enhancing memory retention. This adaptive approach enables confident progress and provides a balanced and responsive memorization experience designed to meet changing needs.

4.7. Needed Skills

4.7.1. Main Skills Needed

4.7.1.1. Machine Learning and Deep Learning

The backbone of this project relies on machine learning—specifically, using Long Short-Term Memory (LSTM) networks and Reinforcement Learning (RL). LSTM networks are vital for understanding patterns in the sequence of a student's past recitations, while RL enables the system to adapt to each student's real-time performance, adjusting plans to meet their current needs. LSTM helps us understand sequential data, which is perfect for tracking memorization over time. On the other hand, RL allows the system to respond to a student's real-time progress by adjusting daily memorization plans. Together, these methods provide a personalized, data-driven learning experience that can flexibly adapt to each student's needs.

4.7.1.2. Data Processing and Analysis

High-quality data is at the heart of this project, as our model will rely on historical recitation records and real-time performance metrics to make informed recommendations. Data processing skills are needed to clean, organize, and transform this data into suitable for our models. The input data must be clean, well-organized, and contain relevant features for the model to make accurate predictions and adjustments. This ensures the LSTM and RL models can make meaningful decisions based on each student's recitation patterns, pacing, and accuracy levels. Without strong data processing, the model's output may be unreliable or difficult to interpret, hindering the project's success.

4.7.1.3. Database Management and SQL

This project requires effective storage and retrieval of both historical and real-time performance data. Database management skills are needed to create and organize tables for storing recitation history, performance metrics, and dynamically generated recommendations. The project combines existing historical data with continuously updated, session-based performance metrics. Database skills are crucial to efficiently organize, access, and manage this information, ensuring that the system can easily retrieve and store data for real-time adjustments. A well-designed database structure allows the model to access necessary information instantly, facilitating smooth operations.

4.7.1.4. Python Programming

Python is the primary programming language for machine learning and data processing, making it essential for implementing and integrating LSTM and RL models. Python provides the libraries we need (like TensorFlow for LSTM and RLlib for RL), and its versatility allows us to manage data processing, model training, and database interactions in one environment.

4.7.2. Skills to Be Developed

4.7.2.1. Long Short-Term Memory (LSTM)

LSTM networks are central to the project, as they are responsible for understanding sequential data patterns from the historical dataset. Gaining experience in sequential modeling is essential for accurate memorization and revision predictions.

4.7.2.2. Reinforcement Learning (RL)

RL is a specialized area of machine learning essential for this project. The RL model is responsible for dynamically adjusting daily recitation plans based on real-time feedback, making it central to the project's goal of adaptive memorization.

4.7.2.3. Data Processing and Feature Engineering

The accuracy of the LSTM model and RL agent relies heavily on clean, well-structured data. Developing data processing skills is crucial for managing historical data and creating real-time performance tracking metrics.

4.7.2.4. Backend Development

To effectively manage data interactions and integrate models, backend development skills are essential for this project. Backend skills enable the creation of a powerful system for retrieving, updating, and managing data from the database, especially with the addition of new recitation data. The use of backend frameworks such as Flask or Django enables secure and efficient processing of data requests and form outputs while facilitating the development of user interfaces such as APIs for teachers to input or view student progress. Backend development skills help ensure that the database, models, and interface components work together seamlessly, maintaining data integrity and supporting the model's adaptability.

4.8. The Initial Design

4.8.1. Use Case Diagram

The use case diagram illustrates the various functionalities of the Quran Memorization Plan Generator system from the perspective of a user. The primary actor interacting with the system is the user, who can be a teacher, administrator, or supervisor overseeing students' memorization progress. Below is a detailed description of the key use cases:

1. **View Students' Plans:** The user can view personalized memorization plans for individual students. These plans include daily goals for memorization and revision, designed to adapt to each student's progress and retention capacity.

2. **Record Recitation Feedback:** During or after a recitation session, the user can record feedback on the student's performance. This feedback is critical for evaluating progress, such as accuracy, speed, and retention.

3. **Adjust Plans After Recitation in Real-Time:** The system uses the feedback recorded by the user to dynamically adjust the student's plan. Based on the student's performance, the system can increase or decrease daily goals, prioritize revision, or modify the pace of memorization.

4. **Generate Periodic Plans:** The system periodically updates memorization plans to reflect long-term performance trends. These updates ensure the plans remain relevant and aligned with the student's overall progress.

5. **Pause or Modify an Ongoing Plan:** The user can manually intervene to pause or modify a student's memorization plan. This feature is helpful in cases where the student requires a temporary break or specific adjustments that the automated system may not account for.

6. **Add New Student:** When a new student is added, the system generates an initial plan. If historical data is unavailable for the new student, the system retrieves average performance metrics from similar students to provide a personalized starting plan.

7. **Generate Initial Plans:** Initial plans for students are generated by the system based on either historical data (for returning students) or average performance data (for new students). These plans provide a foundation that can be further refined through real-time adjustments.

8. **Generate Plans Manually:** The user has the option to create plans manually for students who may require a custom approach. This feature provides flexibility for unique cases where automated plans are insufficient.

The diagram emphasizes the modular nature of the system, highlighting the dependencies and extensions between various functionalities. It ensures flexibility, adaptability, and a user-centric approach to Quran memorization.

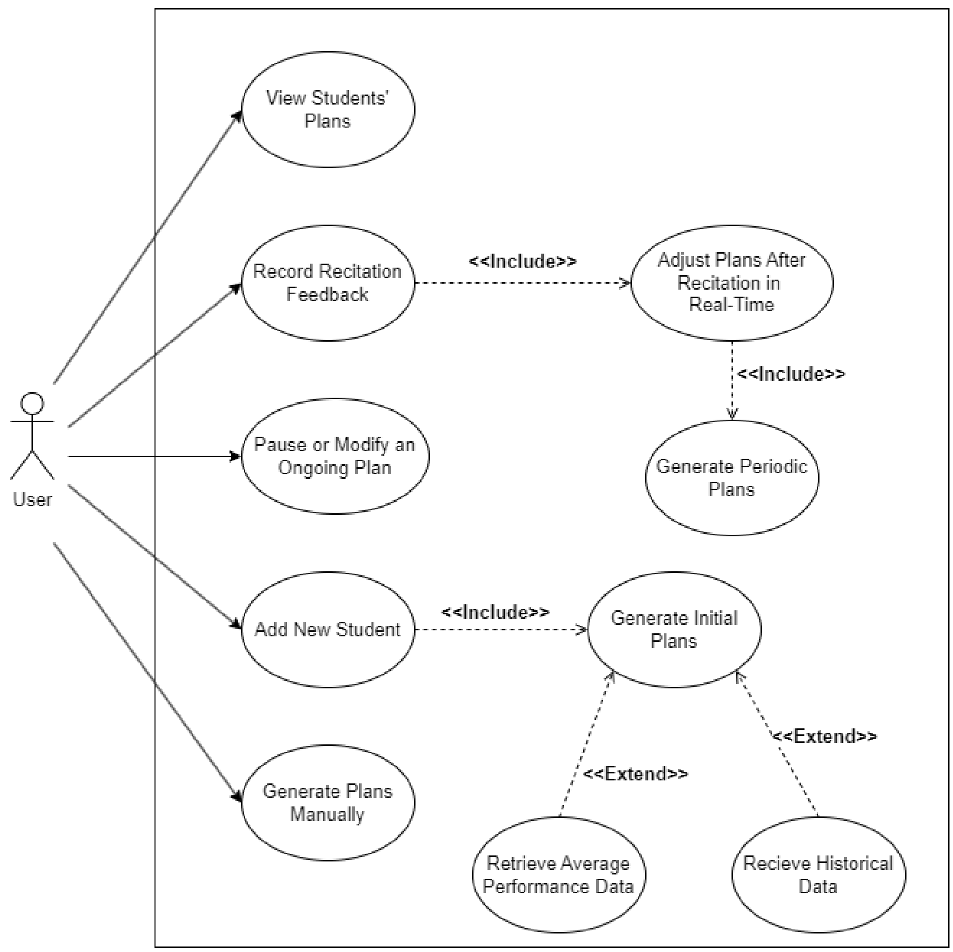

Figure 4: Use Case Diagram

4.8.2. ER Diagram

This Entity-Relationship (ER) Diagram represents the database design for the system. The tables and relationships illustrate how data related to students, recitations, revisions, and performance metrics are organized. The tables shown in the ER diagram are:

- **STUDENT**: Stores general information about each student, including demographics, start date, and memorized portions.
- **SURAH**: Contains details of each Surah (chapter) in the Quran, such as the name, number of verses, and total pages.
- **VERSE**: Tracks details of individual verses within each Surah, including position and letter count.
- **CLUSTERED_STUDENT_PERFORMANCE**: Records cluster-based average memorization and revision pace for different Surahs, used to guide new students without historical data.
- **RECITATION_SESSION**: Logs each recitation session, including the type, count of letters/pages, acceptance status, rating, and RL-based reward signal.
- **REVISION_HISTORY**: Keeps track of each Surah's revision history for each student, including the last revision date, frequency, and time since the last review.
- **DAILY_PLAN**: Holds the student's daily target for new memorization, major revision, and minor revision.
- **RL_PLAN_ADJUSTMENTS_LOG**: Logs adjustments made to the student's plan by the RL model, which helps in tracking changes to the memorization targets and the reason for the adjustment.

- **REVISION_SESSION**: Logs individual revision sessions, specifying the Surah and verses revised along with completion status.

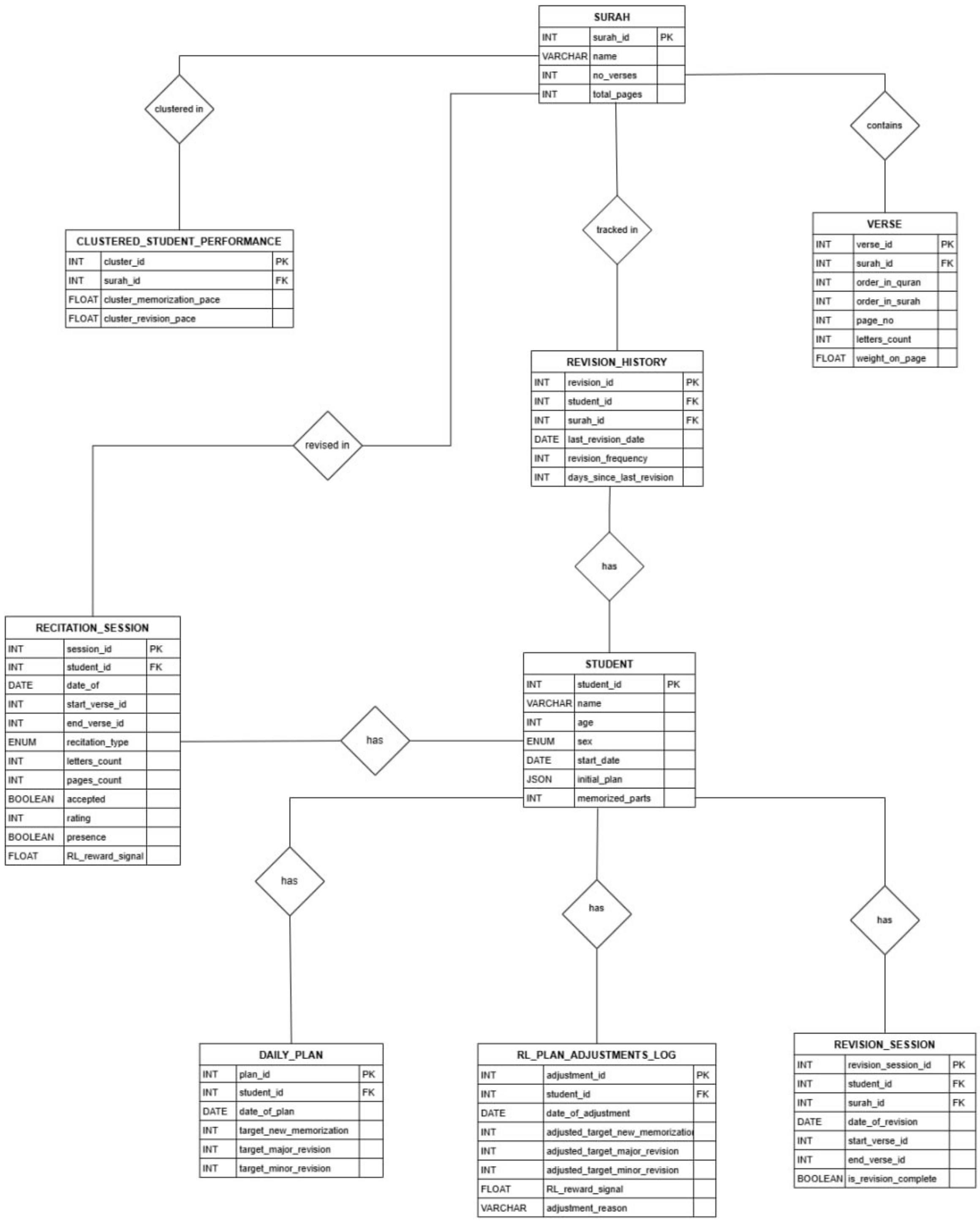

Figure 5: ER Diagram

4.8.3. Data Flow Diagram

A Data Flow Diagram (DFD) is a visual representation that shows how data moves through a system. It illustrates where data originates, how it flows between processes, how it's stored, and how it is used by the system. In this project, the DFD explains how student data is processed and used to create adaptive learning plans.

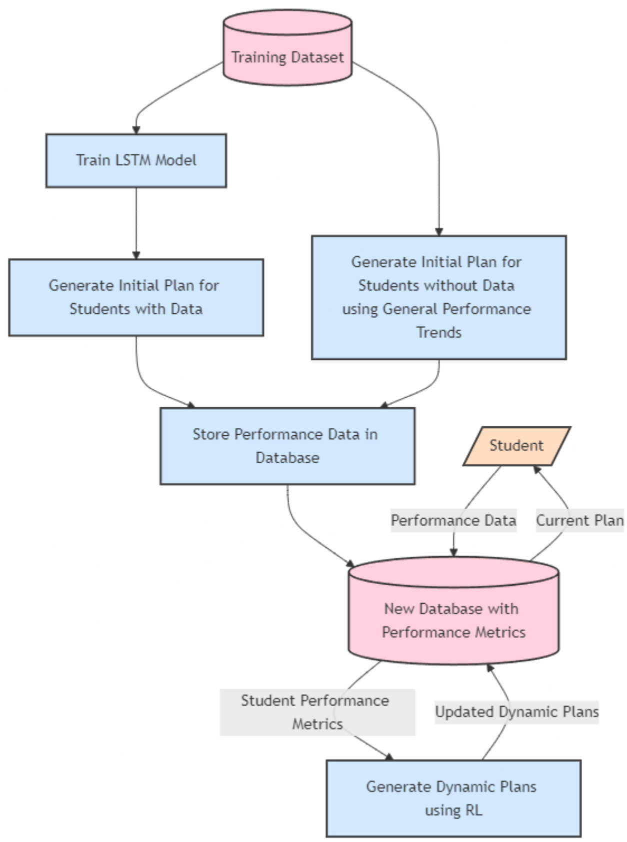

Figure 6: Data Flow Diagram

In the DFD above:

1. **Training Dataset:**

   This initial dataset contains historical data on students' recitation and memorization.

   - The LSTM Model is trained using this dataset to identify memorization patterns and generate initial plans for students with existing data.

- For students without historical data, general performance trends are calculated from this dataset to create an initial plan.

2. **Generate Initial Plans:**
   - For students with data, the LSTM model creates a plan based on their historical performance.
   - For new students, general trends are used to set the starting point.

3. **Store Performance Data in Database:**
   - Both initial plans and ongoing performance data are stored in the database, which collects real-time information on each student’s progress.

4. **Database with Performance Metrics:**
   - The database receives current performance data from the students, including their progress on the memorization plan.
   - It stores metrics such as errors, rating, and recitation pace, which are used to dynamically adjust the student’s learning plan.

5. **Generate Dynamic Plans using RL:**
   - The RL model accesses performance data from the database to create adaptive plans for each student.
   - Based on recent performance, the RL model makes real-time adjustments, providing students with an optimized plan that supports their individual learning pace and retention needs.

4.8.4. Flowchart

A flowchart is a visual representation of a process, showing each step in a sequence. It uses shapes like boxes, diamonds, and arrows to depict the flow of tasks and decisions within a system.

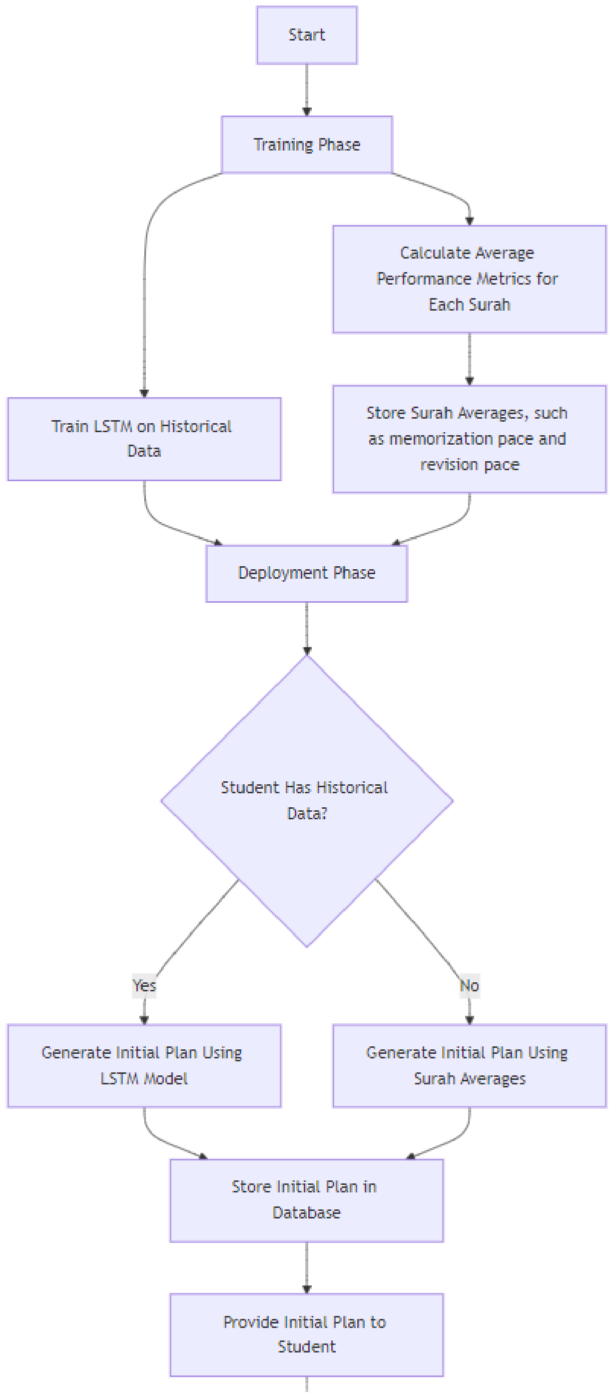

Figure 7: Flowchart (Part 1)

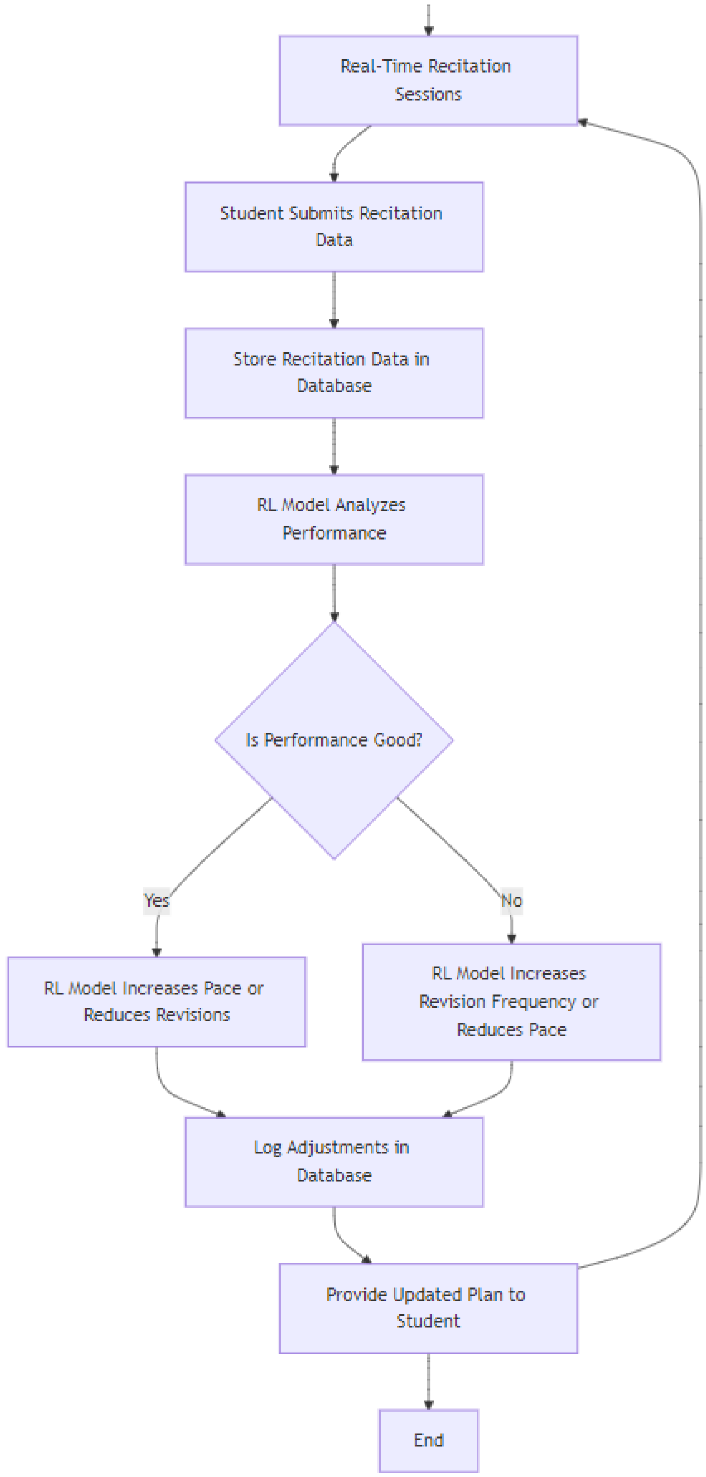

Figure 8: Flowchart (Part 2)

This flowchart illustrates the process for generating and adjusting personalized Quran memorization plans using LSTM and RL models.

1. **Training Phase**:
    - The process starts with the Training Phase, where the system trains an LSTM model on historical data to learn memorization patterns.
    - For students without historical data, average metrics for each Surah are calculated and stored.
2. **Deployment Phase**:
    - When a student needs an initial memorization plan, the system checks if they have historical data.
    - If historical data exists, the LSTM Model generates a plan. If not, the system uses Surah Averages to create the plan.
    - The initial plan is then stored in the database and provided to the student.
3. **Real-Time Recitation Sessions**:
    - During each recitation session, the student submits their performance data, which is stored in the database.
    - The RL Model then analyzes this data to evaluate if the student's performance is progressing well.
4. **Performance Evaluation and Adjustment**:
    - If the student performs well, the RL model may increase the pace or reduce revision frequency.
    - If performance needs improvement, the model may decrease the pace or increase revision sessions.
    - Adjustments are logged into the database, and an updated plan is provided to the student, ensuring it adapts to their needs in real time.

4.8.5. Sequence Diagram

A sequence diagram is a type of interaction diagram that shows how processes operate with each other and in what order. It focuses on the sequence of messages exchanged between components in a system to accomplish a particular function. Sequence diagrams help understand the flow of interactions over time, especially in complex systems.

4.8.5.1. Add New Student

This sequence diagram illustrates the process of adding a new student to the system. When a user submits the details of a new student, the system validates and stores the information in the database. After confirming successful storage, the system initializes a default learning plan for the student and stores it in the database. Finally, the system notifies the user that the new student has been added successfully. This workflow ensures that new students are seamlessly integrated into the system with an initial plan that can be customized further.

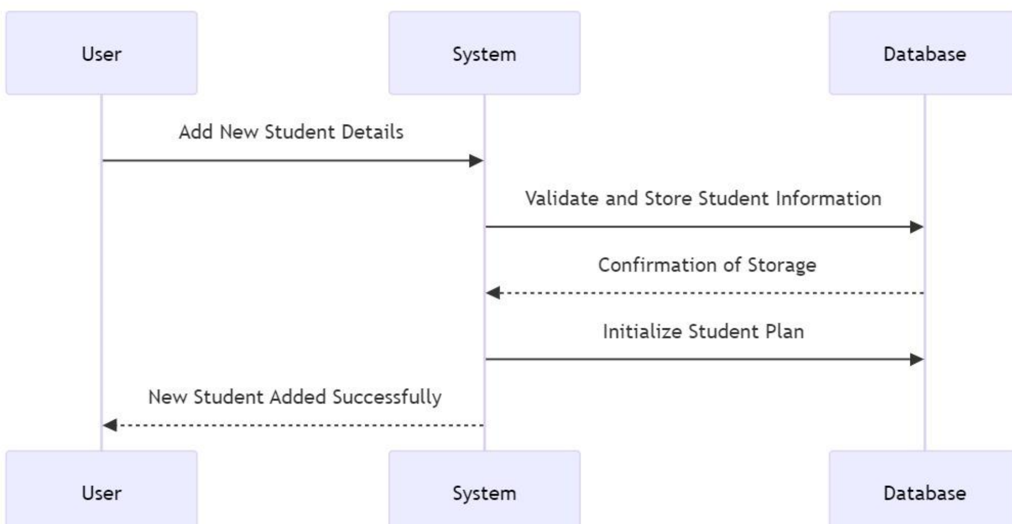

Figure 9: Sequence Diagram - Add New Student

4.8.5.2. Request Initial Plan

This sequence diagram explains how the system generates a personalized plan for a user. When a user requests an initial plan, the system first queries the database for any available historical performance data. If historical data exists, it retrieves this information and uses an advanced Long Short-Term Memory (LSTM) model to generate a personalized plan. If no historical data is available, the system relies on cluster-based averages retrieved from the database to create a plan using a predefined algorithm. Regardless of the method, the generated plan is stored in the database for future use. The user is then presented with their initial plan. This workflow ensures that the system can provide tailored plans, whether the user is new or has a performance history.

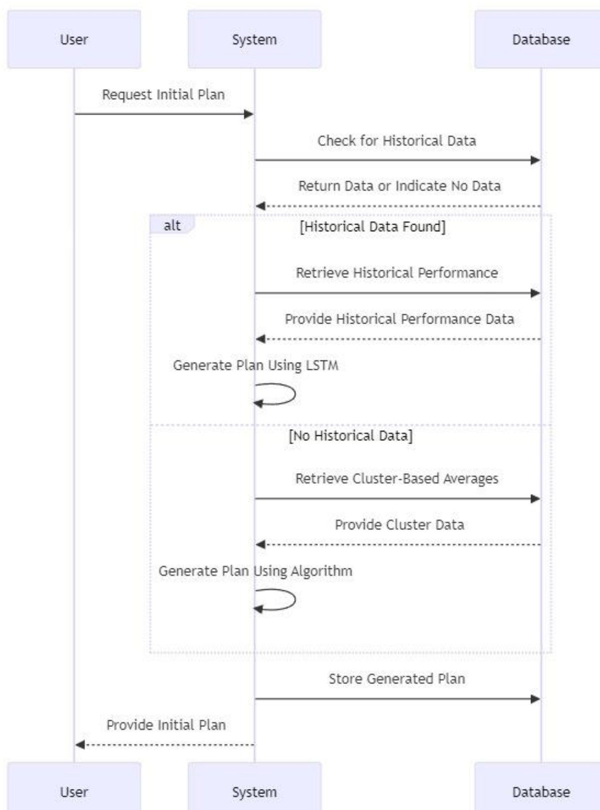

Figure 10: Sequence Diagram - Request Initial Plan

4.8.5.3. Record Recitation Feedback

This sequence diagram focuses on how the system incorporates user feedback to enhance their learning plan. Users submit recitation data, which the system stores in the database. The system then retrieves both historical and current performance data and sends it to a reinforcement learning (RL) model for analysis. The RL model adjusts the memorization plan based on this analysis and returns the updated plan to the system. The system stores the updated plan in the database and notifies the user of the changes. This workflow highlights the adaptive nature of the system, enabling it to refine plans in real time based on user performance.

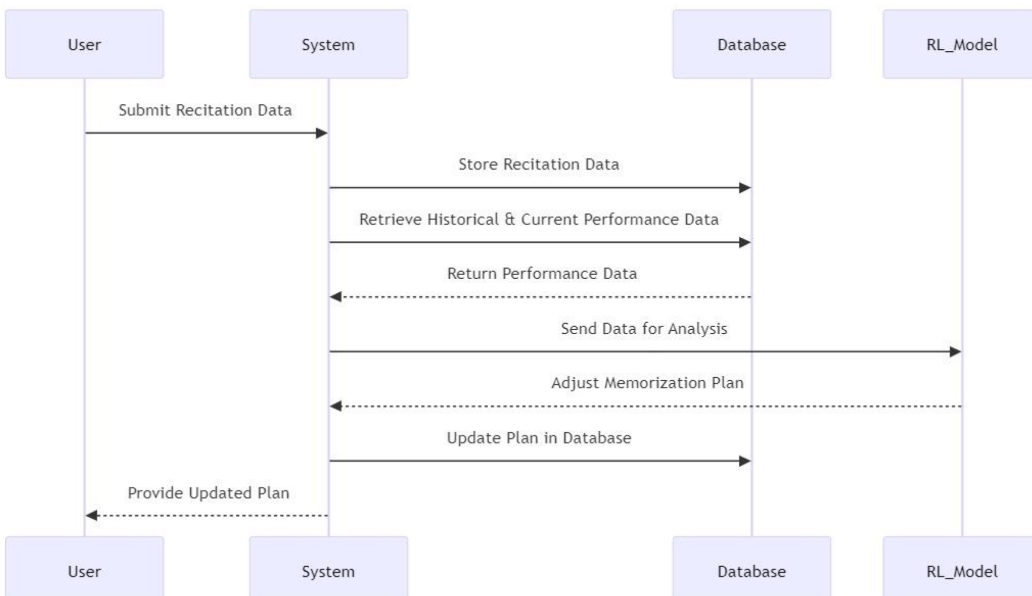

Figure 11: Sequence Diagram - Record Recitation Feedback

4.8.5.4. Pause or Modify an Ongoing Plan

This sequence diagram demonstrates the process by which a user can adjust an ongoing plan. It begins with the user submitting a request to pause or modify the plan. Upon receiving this request, the system retrieves the current plan details from the database. After obtaining the relevant data, the system processes the request to adjust the plan based on the user's input. The updated plan is then stored back in the database to ensure that changes are reflected in future actions. Finally, the system notifies the user that the requested modifications have been successfully applied. This workflow underscores the system's flexibility and its ability to cater to individual user needs dynamically.

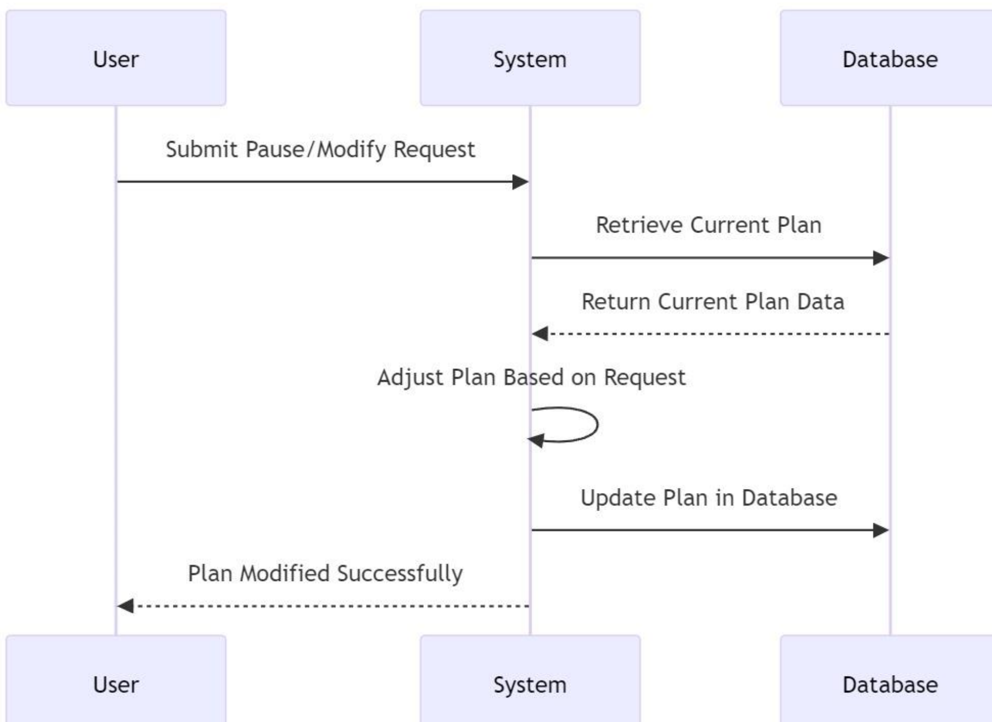

Figure 12: Sequence Diagram - Pause or Modify an Ongoing Plan

4.8.6. Prototype Design

4.8.6.1. Students' Plans

On the Student's Plans page, the teacher can see the list of students assigned to them. They can review a student's previous plan, create a new plan, or add a new student. This is illustrated in Figure 13.

Figure 13: UI - Students' Plans

Figure 14: UI - Add New StudentFigure 15: UI - Students' Plans

Figure 16: UI - Students' Plans

Figure 17: UI - Add New StudentFigure 18: UI - Students' Plans

4.8.6.2. Add New Student

On the Add New Student page, the teacher provides the student's details, such as name and gender, along with the initial period for the plan. If the student has memorized parts of the Quran or has previous plans, the teacher can include that information to help the model create a plan tailored to the student's pace. The teacher can also set optional criteria for the new plan. Finally, they choose to either generate the plan using AI or create it manually. This is shown in Figure 14.

Figure 19: UI - Add New Student

Figure 20: UI - Student's New PlanFigure 21: UI - Add New Student

Figure 22: UI - Add New Student

Figure 23: UI - Student's New PlanFigure 24: UI - Add New Student

4.8.6.3. Student's New Plan

When the teacher chooses to generate a plan using AI, the new plan is displayed with dates and can be modified if needed. This is shown in Figure 15.

Figure 25: UI - Student's New Plan

Figure 26: UI - Create a New PlanFigure 27: UI - Student's New Plan

Figure 28: UI - Student's New Plan

Figure 29: UI - Create a New PlanFigure 30: UI - Student's New Plan

4.8.6.4. Create a New Plan

When creating a new plan for a former student, the teacher selects the initial period for the plan and chooses to either generate it using AI or create it manually. This is shown in Figure 16.

Figure 31: UI - Create a New Plan

Figure 32: UI - View Student's PlansFigure 33: UI - Create a New Plan

Figure 34: UI - Create a New Plan

Figure 35: UI - View Student's PlansFigure 36: UI - Create a New Plan

4.8.6.5. View Student's Plans

When the teacher views the student's plan, they can record the achievement and evaluation for each recitation. This information helps the model create a more tailored plan for the student in the future. This is shown in Figure 17.

Figure 37: UI - View Student's Plans

Figure 38: UI - View Student's Plans

Figure 39: UI - View Student's Plans

Figure 40: UI - View Student's Plans

CHAPTER 5

Conclusion

This project represents a significant step forward in creating an adaptive, personalized system to support students in memorizing the Quran. By integrating Long Short-Term Memory (LSTM) networks with Reinforcement Learning (RL), the system addresses critical challenges in adapting memorization plans to individual learning patterns and retention needs. The LSTM model captures sequential dependencies from historical data, generating personalized memorization and revision plans. Meanwhile, the RL model dynamically adjusts these plans in real-time, ensuring steady progress while maintaining the retention of previously memorized material. This hybrid approach ensures that the system promotes progress without overwhelming students. The recommendation system further benefits from the database, which efficiently stores and retrieves performance metrics, enhancing the decision-making capabilities of the models.

To further enhance the system, we might incorporate user feedback into the real-time adjustment process and develop a more intuitive user interface for seamless interaction. Future improvements should also focus on enhancing the system's scalability to accommodate a larger number of students and optimizing computational efficiency to ensure smooth operation across different environments.

Looking ahead, emphasis will be placed on the full implementation of the system, starting with training the LSTM and RL models. Thorough preprocessing will ensure the dataset is optimized for training, while the database structure will be finalized to support efficient storage and retrieval. Following implementation, testing will assess system accuracy, adaptability, and usability. Feedback from these evaluations will be used to refine the models, enhance database functionality, and improve overall system stability before deployment.

Ultimately, this system combines innovative AI techniques with thoughtful design and user-centric features to create a transformative tool for Quran memorization. By fostering both retention and progress, it addresses a vital educational need while paving the way for further advancements in personalized learning systems.

References

[1] K. Haryono, R. A. Rajagede, and M. Negara, “Quran Memorization Technologies and Methods: Literature Review,” *IJID (International Journal on Informatics for Development)*, vol. 11, pp. 192–201, Jan. 2023, doi: 10.14421/ijid.2022.3746.

[2] S. Maghsudi, A. Lan, J. Xu, and M. van der Schaar, “Personalized Education in the Artificial Intelligence Era: What to Expect Next,” *IEEE Signal Processing Magazine*, vol. 38, no. 3, pp. 37–50, May 2021, doi: 10.1109/MSP.2021.3055032.

[3] Research Scholar from School of Computer and Information Sciences, University of Hyderabad.., G. R. Anil, S. A. Moiz, and Assistant Professor at School of Computer and Information Sciences, University of Hyderabad., “Personalized Dynamic Learning Plan Generator for Smart Learning Environments,” *IJRTE*, vol. 8, no. 2, pp. 6175–6180, Jul. 2019, doi: 10.35940/ijrte.B3806.078219.

[4] M. Somasundaram, K. A. M. Junaid, and S. Mangadu, “Artificial Intelligence (AI) Enabled Intelligent Quality Management System (IQMS) For Personalized Learning Path,” *Procedia Computer Science*, vol. 172, pp. 438–442, Jan. 2020, doi: 10.1016/j.procs.2020.05.096.

[5] H. Oussama, N. El Faddouli, M. H. Alaoui Harouni, and J. Lu, “Artificial Intelligent in Education,” *Sustainability*, vol. Volume 14, p. 5: 2862, Mar. 2022, doi: 10.3390/su14052862.

[6] L. Zhang, J. D. Basham, and S. Yang, “Understanding the implementation of personalized learning: A research synthesis,” *Educational Research Review*, vol. 31, p. 100339, Nov. 2020, doi: 10.1016/j.edurev.2020.100339.

[7] D. Niu, Z. Xia, Y. Liu, T. Cai, T. Liu, and Y. Zhan, “ALSTM: Adaptive LSTM for Durative Sequential Data,” in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov. 2018, pp. 151–157. doi: 10.1109/ICTAI.2018.00032.

[8] R. S. Sutton and A. Barto, *Reinforcement learning: an introduction*, Second edition. in Adaptive computation and machine learning. Cambridge, Massachusetts London, England: The MIT Press, 2020.