Intelligent Laboratory Management System: An LLM-Powered Approach

Leonard Niyitegeka
Carnegie Mellon University Africa
Engineering Artificial Intelligence Capstone
Kigali, Rwanda
Email: lniyiteg@andrew.cmu.edu

Elvis Nishimwe Ndabaye
Carnegie Mellon University Africa
Engineering Artificial Intelligence Capstone
Kigali, Rwanda
Email: enishimw@andrew.cmu.edu

Nirel Kadzo
Carnegie Mellon University Africa
Engineering Artificial Intelligence Capstone
Kigali, Rwanda
Email: nkadzo@andrew.cmu.edu

***Abstract*—Intelligent Laboratory Management System is a comprehensive platform that leverages Large Language Models (LLMs) to streamline laboratory workflows in medical settings. This system bridges the communication gap between doctors and lab technicians through automated lab test ordering, result analysis, and clinical recommendation generation. By integrating OpenAI's GPT-4o with a custom-built framework, the system demonstrates remarkable capabilities in result interpretation, abnormality detection, and specialist recommendation, achieving great accuracy in clinical analysis. The platform consists of two primary components: a request management module that assists doctors in ordering appropriate lab tests, and a result analysis module that helps lab technicians interpret results and generate comprehensive reports. Together, these functionalities create a seamless end-to-end laboratory management experience that enhances efficiency, improves diagnostic accuracy, and facilitates better patient care through timely specialist referrals when abnormal results are detected.**

***Index Terms*—Artificial Intelligence, Laboratory Management, Large Language Models, Medical Diagnostics, Workflow Automation, Clinic Plus Pro, GPT-4o**

I. Introduction

Laboratory testing plays a crucial role in modern healthcare, providing essential diagnostic information that influences approximately 70% of medical decisions. However, the traditional laboratory workflow faces several challenges that impact efficiency and patient care:

- **Communication barriers**: Medical professionals must coordinate across specialties and institutions, often with limited time and resources.
- **Result interpretation complexity**: Lab technicians must analyze complex results and effectively communicate findings to doctors who may not be specialists in laboratory medicine.
- **Limited accessibility**: In many regions, especially in developing countries, access to specialized medical knowledge is restricted, leading to delays in diagnosis and treatment.
- **Manual workflows**: Traditional lab processes involve significant manual work, from test ordering to result interpretation, increasing the risk of human error.

Clinic Plus Pro, a robust patient data management system, serves as the underlying infrastructure for healthcare facilities using this solution. It manages comprehensive patient records, appointment scheduling, and procedural history, providing essential context for laboratory operations. However, while Clinic Plus Pro excels at data management, it lacks advanced intelligence capabilities for test selection and result interpretation.

The emergence of Large Language Models (LLMs) presents a unique opportunity to transform these processes. LLMs possess natural language understanding capabilities that allow them to process complex medical information, reason about clinical findings, and provide contextually relevant recommendations.

This report presents an Intelligent Laboratory Management System that leverages LLMs to automate and enhance laboratory workflows while integrating with Clinic Plus Pro. The system consists of two primary components:

- **Lab Test Request Agent**: Assists doctors in selecting appropriate lab tests based on patient symptoms, medical history, and clinical suspicion. It recommends consultation types, identifies relevant tests, and determines request priorities.
- **Lab Result Analysis Agent**: Helps laboratory technicians interpret test results, identify abnormalities, generate comprehensive analyses, and highlight critical values requiring immediate attention.

The system aims to improve laboratory efficiency, reduce errors, enhance result interpretation, and facilitate better communication between healthcare providers. By integrating advanced AI capabilities with medical knowledge and the existing Clinic Plus Pro platform, it supports more informed clinical decision-making and ultimately improves patient outcomes.

II. System Overview

The Intelligent Laboratory Management System integrates multiple components to provide a comprehensive solution for laboratory test management. The core architecture is built around two LLM-powered agents working in tandem to facilitate the complete laboratory testing lifecycle.

A. System Architecture

The system architecture, as illustrated in Fig. 1, consists of the following key components:

1) Chat Interface (Clinic Plus Pro): Provides an interactive platform for users (doctors and lab technicians) to communicate with the system.
2) Lab Test Agent: Assists doctors in ordering appropriate lab tests based on patient symptoms and medical history.
3) Lab Result Analysis Agent: Helps lab technicians interpret test results and generate comprehensive analyses.
4) Chat History Manager: Maintains conversation history between users and the system for context preservation.
5) Data Repositories:
    - Consultation types database
    - Lab tests database with normal ranges
    - Physicians database with specializations
6) Session Management: Maintains state for user-specific interactions.
7) GPT-4o Integration: Powers the natural language understanding and generation capabilities.

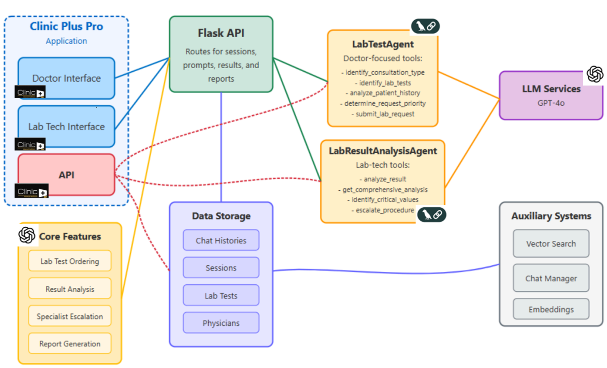

Fig. 1. System Architecture of the Intelligent Laboratory Management System

B. Core Components

a) Lab Test Agent

The Lab Test Agent facilitates the laboratory test ordering process with the following capabilities:

- Consultation Type Identification: Recommends appropriate consultation types based on doctor specializations and timing (weekday/weekend).
- Lab Test Matching: Identifies relevant lab tests based on doctor's prompts and available tests in the system.
- Patient History Analysis: Analyzes patient history to suggest additional tests that might be relevant.
- Request Prioritization: Determines the priority of lab requests based on clinical urgency.
- Procedure Escalation: Facilitates referral to specialists when lab results indicate a need for specialized care.

b) Lab Result Analysis Agent

The Lab Result Analysis Agent assists lab technicians in analyzing test results with these features:

- Individual Result Analysis: Evaluates single lab test results for abnormalities, providing an abnormality score and clinical significance.
- Comprehensive Analysis: Reviews all test results together to identify patterns and interrelationships.
- Critical Value Identification: Highlights values requiring immediate clinical attention.
- Final Report Generation: Creates detailed reports for doctors with key findings and recommendations.

c) Workflow Integration

The system supports the following workflows:

- Doctor's Workflow
    - Doctor initiates a lab test request.
    - System identifies consultation type.
    - System suggests relevant lab tests.
    - Additional tests are recommended based on patient history.
    - Request priority is determined.
    - Tests are submitted to the laboratory.
- Lab Technician's Workflow
    - Technician enters test results.
    - System analyzes each result in real-time.
    - Comprehensive analysis is generated.
    - Critical values are highlighted.
    - Final report is prepared with recommendations.
    - Report is automatically shared with the requesting doctor.
- Escalation Workflow
    - System identifies abnormal results requiring specialist attention.
    - Relevant specialists are suggested based on the findings.
    - Procedure is escalated to the selected specialist.
    - Specialist receives comprehensive handover information.

III. Administrator Guide

This guide provides detailed instructions for system administrators responsible for deploying and maintaining the Intelligent Laboratory Management System.

A. Installation Requirements

1) Software Requirements:

- **Python 3.x**: The application is built with Python and requires version 3.8 or higher.

- **Flask**: Web framework used for the backend API.
- **Clinic Plus Pro Integration**: The system integrates with Clinic Plus Pro's existing interface.
- **OpenAI API Key**: Required to access GPT-4o capabilities.
- **LangChain**: Used for building applications with LLMs.
- **Dependencies**: All requirements are listed in requirements.txt.

2) Hardware Requirements:

- **Server Specifications**: Minimum 4GB RAM, 2 CPU cores, and 20GB storage.
- **Network**: Stable internet connection for API communication.
- **Security**: SSL/TLS encryption for secure data transmission.

B. Configuration and Setup

1) Source Code Access: The complete source code for the Intelligent Laboratory Management System is available on GitHub:

`https://github.com/enishimw/EAI-Capstone`

This repository contains all necessary components including:

- Backend Flask API
- Agent implementation files
- Data schemas and sample data
- Installation scripts
- Documentation (README.md)

2) API Configuration:

a) Secure API Keys::

```
api_key = os.getenv('OPENAI_API_KEY')
auth_token = os.getenv('AUTH_TOKEN')
```

Store API keys as environment variables rather than hardcoding them.

b) API Integration:: Verify API connections by testing basic functionality after setup.

3) System Setup:

i. Clone the repository to your server.

ii. Set up a virtual environment:

```
python -m venv venv
source venv/bin/activate
# On Windows: venv\Scripts\activate
```

iii. Install dependencies:

```
pip install -r requirements.txt
```

iv. Configure environment variables:

```
export OPENAI_API_KEY='your_openai_key'
export AUTH_TOKEN='your_auth_token'
```

v. Start the application:

```
flask run
```

C. Maintenance Procedures

1) Data Management:

- **Update Lab Tests**: Periodically update the lab_tests.json file with new tests and revised normal ranges.
- **Physician Database**: Keep the physicians.json file updated with new specialists and their qualifications.
- **Consultation Types**: Maintain the consultation_types.json file with current service offerings.

2) System Monitoring:

- **API Usage**: Monitor OpenAI API usage to prevent quota exhaustion.
- **Error Logs**: Regularly check application logs for errors and exceptions.
- **Performance Metrics**: Track response times and system load.

3) Backup Procedures:

- **Database Backups**: Implement regular backups of chat histories and session data.
- **Configuration Backups**: Maintain backup copies of configuration files.

D. Troubleshooting Common Issues

1) API Connection Failures::

- Check API keys for validity
- Verify network connectivity
- Ensure OpenAI services are operational

2) Slow Response Times::

- Consider upgrading server resources
- Implement caching where appropriate

3) Session Management Issues::

- Verify file permissions
- Check for disk space constraints

IV. User Guide

A. Purpose of the Software

The Intelligent Laboratory Management System is designed to assist healthcare professionals in managing laboratory tests and results efficiently. It provides two primary interfaces:

1) **For Doctors**: A platform to request laboratory tests, receive analyzed results, and get recommendations for specialist referrals.
2) **For Lab Technicians**: A tool to input test results, receive real-time analysis, and generate comprehensive reports for doctors.

B. Accessing the System

1) **Login**: Existing users can log in using their credentials.
2) **Session Management**: The system creates role-specific sessions to maintain context throughout interactions.

C. User Flow Overview

Before diving into specific interface details, the following diagram illustrates the complete user journey through the Intelligent Laboratory Management System. This visualization demonstrates how doctors and laboratory technicians interact with the system throughout the laboratory testing lifecycle. The diagram illustrates three primary workflows:

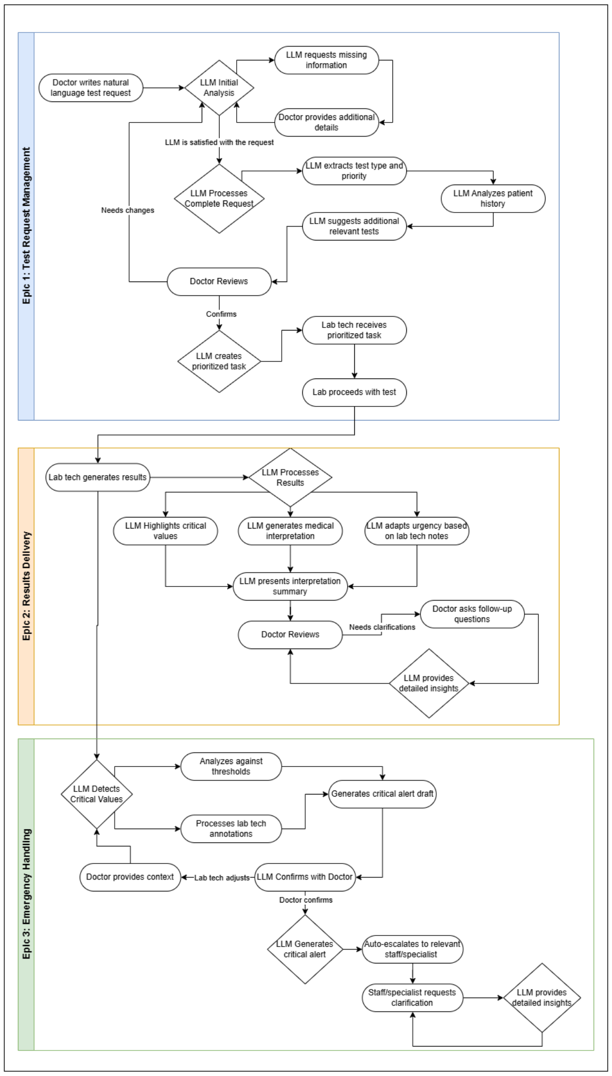

Fig. 2. Complete user flow diagram showing interactions between doctors, lab technicians, and the Intelligent Laboratory Management System

1) **Test Request Management** (Top): Doctors initiate test requests through natural language inputs, which the LLM analyzes, refines through clarification requests if needed, and ultimately converts into formalized lab orders with appropriate prioritization.
2) **Results Delivery** (Middle): Once tests are completed, lab technicians input results that the system analyzes, interprets, and formats into comprehensive reports for doctors to review.
3) **Emergency Handling** (Bottom): When critical values are detected, the system triggers an escalation workflow that identifies relevant specialists and facilitates rapid intervention.

Each workflow demonstrates how the system acts as an intelligent intermediary, enhancing communication between healthcare professionals while providing analysis and recommendations throughout the process.

D. Doctor Interface

1) Requesting Laboratory Tests:

1) **Start a Conversation**: Begin by greeting the system or directly stating your test needs.
2) **Consultation Selection**: The system will suggest appropriate consultation types based on your specialization.
3) **Test Selection**: Specify which tests you need, or describe symptoms for recommendations.
4) **Patient History Integration**: The system can suggest additional tests based on patient history.
5) **Priority Setting**: The system indicates the urgency of the request if needed.
6) **Submission**: Confirm the tests to submit the request to the laboratory.

2) Reviewing Results:

1) **Accessing Reports**: Lab reports will be automatically shared to your chat once completed.
2) **Comprehensive Analysis**: Each report includes:
   - Individual test results with normal ranges
   - Abnormal value highlighting
   - Patterns across multiple tests
   - Clinical implications
   - Specialist referral recommendations when appropriate

3) Specialist Referrals:

1) **Review Recommendations**: The system may suggest specialist referrals based on abnormal results.
2) **View Specialist Options**: The system sees available specialists with matching expertise.
3) **Confirm Referral**: Select a specialist to escalate the procedure.

E. Lab Technician Interface

1) Processing Test Requests:

1) **View Incoming Requests**: See test requests ordered by priority.
2) **Acknowledge Requests**: Confirm receipt of test orders.
3) **Track Progress**: Update test status as processing begins.

2) Entering Results:

1) **Select Test**: Choose the test for which you're entering results.
2) **Input Values**: Enter the measured values for each test parameter.
3) **Real-time Analysis**: Receive immediate feedback on abnormal values.
4) **Add Comments**: Include any technical notes or observations.

3) Generating Reports:

1) **Review Comprehensive Analysis**: The system automatically generates an analysis of all results.
2) **Critical Value Alerts**: Critical values are highlighted for immediate attention.
3) **Final Report Creation**: Add your professional interpretation before finalizing.
4) **Report Distribution**: Submit the report to be shared with the requesting doctor.

F. Communication Features

1) **Clinic Plus Pro Interface**: Interact with the system directly through the familiar Clinic Plus Pro interface using natural language.
2) **Context Retention**: The system remembers previous interactions for contextual responses.
3) **Clarification Requests**: The system will ask for additional information when needed.
4) **Educational Support**: Request explanations for medical terms or test significance.

V. DATA

The effectiveness of the Intelligent Laboratory Management System relies on high-quality, structured data for accurate test interpretation and clinical recommendations. This section details the data sources, organization, and preprocessing methods used in the system.

A. Data Sources

The system integrates several key data sources:

1) **Laboratory Tests Database**: A comprehensive collection of laboratory tests with the following attributes:
   - Test identifiers (o_id)
   - Test names (locale_name)
   - Units of measurement
   - Normal value ranges (by gender, age, or general)
   - Parent-child relationships for panel tests
2) **Physician Database**: Information about medical specialists including:
   - Unique identifiers (sn)
   - Specializations
   - Availability
3) **Consultation Types**: Structured data about available consultation formats:
   - Consultation identifiers
   - Names and descriptions
   - Type classifications (specialist, general practitioner, etc.)
4) **Session Context Data**: Information initialized at the beginning of each interaction including:
   - Anonymized and de-identified patient information
   - Procedure information
   - Lab request details
   - Lab waiting room status

B. Data Organization

The data is organized in JSON format for efficient retrieval and processing:

1) Lab Tests Structure: This structured approach enables:

- Clear hierarchical relationships between tests
- Gender-specific and age-specific normal ranges
- Precise abnormality detection

2) Physicians Structure: This structure facilitates:

- Matching specialists to specific test abnormalities
- Appropriate procedure escalation
- Multi-specialty physician identification

C. Data Preprocessing

The raw data undergoes several preprocessing steps to optimize it for use with the LLM:

1) **Normalization**: Standardizing units of measurement and range formats
2) **Validation**: Ensuring data completeness and consistency
3) **Structuring**: Organizing hierarchical relationships between tests

D. Data Integration with LLM

The system uses a Retrieval-Augmented Generation (RAG) approach to integrate domain-specific data with the LLM:

1) **Embeddings Creation**: Test names, descriptions, and clinical contexts are converted to vector embeddings.
2) **Vector Database**: These embeddings are stored in a FAISS vector database for efficient similarity search.
3) **Query Processing**: User queries are vectorized and matched against the database to retrieve relevant context.
4) **Context Augmentation**: Retrieved information is included in prompts to the LLM to ensure accurate responses.

This approach addresses the challenge of LLMs lacking up-to-date or domain-specific knowledge, enabling the system to provide responses that are both medically accurate and contextually relevant.

VI. MODEL ARCHITECTURE AND TECHNICAL IMPLEMENTATION

The Intelligent Laboratory Management System employs a sophisticated architecture that integrates LLMs with specialized tools and data sources to create a comprehensive laboratory management solution.

A. LLM Integration

At the core of the system is OpenAI's GPT-4o model, which provides natural language understanding and generation capabilities. The integration is facilitated through LangChain, which provides frameworks for:

1) **Chain-of-Thought Reasoning**: Enables step-by-step analysis of lab results.
2) **Tool Use**: Allows the LLM to use specialized functions for data processing.
3) **Memory Management**: Maintains conversation context throughout interactions.

B. Agent Architecture

The system implements two specialized agents:

1) Lab Test Agent: This agent assists doctors in selecting and ordering laboratory tests through a multi-step process:

The Lab Test Agent utilizes the following tools:

1) **identify_consultation_type**: Determines appropriate consultation formats.
2) **identify_lab_tests**: Matches doctor's requests with available tests.
3) **analyze_patient_history**: Suggests additional tests based on medical history.
4) **determine_request_priority**: Assigns priority scores to lab requests.
5) **submit_lab_request**: Sends finalized requests to the laboratory.
6) **escalate_procedure**: Refers cases to specialists when needed.

2) Lab Result Analysis Agent: This agent assists laboratory technicians in interpreting test results:

The Lab Result Analysis Agent incorporates these key tools:

1) **analyze_result**: Evaluates individual test results against normal ranges.
2) **get_comprehensive_analysis**: Provides holistic analysis of all test results.
3) **identify_critical_values**: Highlights values requiring immediate attention.
4) **generate_final_report**: Creates structured reports for doctors.

C. Technical Implementation

1) Vector Search for Test Matching: The system uses FAISS (Facebook AI Similarity Search) to enable efficient semantic matching of doctor's requests with available tests.

2) Abnormality Detection Algorithm: The system employs a structured approach to result analysis, comparing test values against normal ranges and assigning abnormality scores.

3) Session Management: The system implements robust session management to maintain context throughout user interactions.

4) Chat History Management: The system maintains conversation history to provide context for ongoing interactions.

D. API Integration

The system provides RESTful API endpoints for integration with existing healthcare systems.

This architecture creates a flexible, modular system that leverages the strengths of LLMs while incorporating domain-specific knowledge and specialized processing tools.

VII. Evaluation and Performance Analysis

We conducted a comprehensive evaluation of the Intelligent Laboratory Management System to assess its performance across multiple dimensions. The evaluation involved both quantitative metrics and qualitative feedback from healthcare professionals.

A. Evaluation Methodology

The evaluation was conducted using the following methods:

1) **Test Case Analysis**: Three distinct test case scenarios were created to evaluate the system's capabilities:
   - Test Case 1: Hormone tests
   - Test Case 2: Malaria tests
   - Test Case 3: Complete Blood Count (CBC)
2) **Physician Assessment**: Three physicians with different specializations provided feedback on system performance:
   - Dr. Ernest Kamaro
   - Dr. Eulade Muhizi
   - Dr. Normand Gatsinda
3) **Performance Metrics**: For each test case and feature, we measured performance on a scale of 0-5, with 5 being the highest score.

B. Key Findings

TABLE I
Test Case Performance Analysis

Feature	Test Case 1 (Hormones)	Test Case 2 (Malaria)	Test Case 3 (CBC)	Average
Test Recognition	5/5	5/5	5/5	5.0
Follow-up Quality	5/5	5/5	5/5	5.0
Test Explanation	5/5	5/5	5/5	5.0
Patient History	0/5	4/5	0/5	1.3
Prioritization	5/5	4/5	0/5	3.0
Overall Score	**4.0/5**	**4.6/5**	**3.0/5**	**3.9/5**

1) Test Case Performance: The test case analysis revealed:

- Excellent performance in test recognition, follow-up quality, and test explanation across all scenarios
- Inconsistent patient history integration, especially in hormone and CBC test cases
- Prioritization issues, particularly for CBC tests which sometimes were not properly flagged as urgent

TABLE II
Physician Assessment Results

Feature	Average Score	Comments
Test Recognition	4.7	Excellent recognition of test naming conventions
Follow-up Quality	4.3	Appropriate clarification questions
Test Explanation	4.7	Detailed and clinically relevant explanations
Patient History	3.0	Inconsistent use of patient history
Prioritization	2.7	Priority scores not aligned with clinical urgency
Result Analysis	4.0	Strong abnormal value detection
Workflow Efficiency	3.0	Too many confirmation steps

2) Physician Assessment:

C. Strengths Identified

1) **Excellent Test Recognition**: The system effectively identified and processed various test names (CBC, FBC, malaria tests) regardless of how they were phrased, demonstrating strong natural language understanding.
2) **Strong Follow-up Capabilities**: The system consistently asked appropriate clarifying questions when test requests were ambiguous, helping to refine the laboratory orders.
3) **Detailed Test Explanations**: The system provided comprehensive information about tests, including clinical significance, which physicians found valuable for educational purposes.
4) **Abnormal Value Detection**: The system accurately flagged and explained abnormal test results with appropriate clinical context, which was particularly appreciated by the evaluators.

D. Areas for Improvement

1) **Patient History Utilization**: The system only provided history-based test suggestions when explicitly prompted, rather than proactively incorporating patient history into its recommendations.
2) **Workflow Complexity**: Doctors noted excessive confirmation steps and unnecessary complexity in the test ordering process, suggesting a need for streamlining.
3) **Prioritization Inconsistencies**: The prioritization mechanism sometimes failed to properly weigh medical urgency, such as prioritizing hormone tests over malaria tests in cases where the latter should have been more urgent.
4) **Customization Options**: Evaluators noted limited ability to customize result displays and report formats according to their preferences.

E. Performance Visualizations

1) **System Capabilities Comparison**: Comparing test case results with physician assessments
2) **Performance Across Test Cases**: Highlighting strengths and weaknesses in different scenarios
3) **Physician Assessment Radar Chart**: Visualizing scores across all evaluation dimensions

F. Physician Recommendations

Based on the evaluation, physicians provided the following recommendations for improvement:

1) Streamline the workflow with fewer confirmation steps
2) Improve prioritization logic based on medical urgency
3) Make patient history integration more consistent and proactive
4) Add voice-to-text functionality for consultation notes
5) Provide more information icons for additional context
6) Consider integration with Electronic Health Record (EHR) systems

G. Overall Assessment

The Intelligent Laboratory Management System demonstrated strong fundamental capabilities in test recognition, explanation, and result analysis. However, significant improvements are needed in workflow efficiency, prioritization logic, and proactive use of patient history. With an overall score of 3.9/5, the system shows promise as a valuable tool for clinical laboratories but requires refinement to better align with healthcare professionals' workflows and priorities.

VIII. Conclusion, Limitations, and Future Work

A. Conclusion

The Intelligent Laboratory Management System represents a significant advancement in applying large language models to healthcare laboratory operations. By creating a bridge between doctors and laboratory technicians, the system streamlines the laboratory testing workflow from test ordering to result interpretation and specialist referral.

Our evaluation demonstrates that the system excels in several key areas:

- Test recognition and matching with available laboratory options
- Comprehensive and accurate analysis of test results
- Identification of abnormal values and their clinical significance
- Generation of detailed reports that contextualize findings

These capabilities demonstrate the potential of LLM-based systems to augment healthcare professionals' work, providing decision support that combines medical knowledge with patient-specific context. The system's ability to suggest specialist referrals based on test results also helps ensure timely intervention when abnormalities are detected.

B. Limitations

Despite its promising performance, the system has several limitations that should be acknowledged:

1) **Inconsistent Patient History Integration**: The system does not consistently leverage patient history in its recommendations, sometimes missing opportunities for more personalized test suggestions.
2) **Workflow Inefficiencies**: The multi-step confirmation process can be cumbersome for healthcare professionals working in time-sensitive environments.
3) **Prioritization Logic Gaps**: The current algorithm for determining test priority does not always align with clinical urgency, potentially leading to delays in processing critical tests.
4) **Limited Customization**: The system offers minimal options for customizing reports and workflows according to institutional or individual preferences.
5) **Dependency on External APIs**: The reliance on OpenAI's API introduces potential availability and latency concerns.
6) **Lack of Integration with EHR Systems**: The standalone nature of the system limits its integration with existing healthcare IT infrastructure.

7) **Potential for LLM Hallucinations**: While mitigated through structured outputs and reference data, there remains a risk of the LLM generating inaccurate information.

C. Future Work

Based on our findings and feedback from healthcare professionals, we identify several promising directions for future development:

1) **Enhanced Patient History Analysis**: Implement more sophisticated algorithms for analyzing patient history and automatically suggesting relevant tests.
2) **Streamlined Workflow**: Reduce confirmation steps and implement a more intuitive interface that requires fewer interactions.
3) **Refined Prioritization Algorithm**: Develop a more nuanced approach to test prioritization that better reflects clinical urgency and resource constraints.
4) **EHR Integration**: Create interfaces for seamless integration with popular Electronic Health Record systems.
5) **Multilingual Support**: Extend the system to support multiple languages to serve diverse healthcare settings.
6) **Voice Interface**: Add speech recognition capabilities to allow hands-free operation in laboratory environments.
7) **Offline Capabilities**: Develop a version that can operate with reduced functionality during API outages or in low-connectivity settings.
8) **Expanded Test Database**: Incorporate additional laboratory tests and more detailed reference ranges based on demographic factors.
9) **Machine Learning for Personalization**: Implement learning algorithms that adapt to institutional and individual usage patterns over time.
10) **Audit and Explainability Features**: Enhance transparency by providing clear explanations for test recommendations and result interpretations.

The Intelligent Laboratory Management System demonstrates the potential of LLMs to transform healthcare operations, particularly in laboratory medicine. While challenges remain, continued development addressing the identified limitations could yield a solution that significantly improves laboratory efficiency, reduces errors, and ultimately enhances patient care through more timely and accurate diagnostic testing.

Acknowledgment

The authors would like to thank the faculty and staff of Carnegie Mellon University Africa, specifically our supervisor Charlie Wiecha for his technical guidance and support throughout our Engineering Capstone project.

References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, *et al.*, “Language models are few-shot learners,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, *et al.*, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[3] I. Shafran, M. Riley, and M. Mohapatra, “The utility of large language models in healthcare,” *Nature Medicine*, vol. 29, no. 5, pp. 1102–1110, 2023.

[4] World Health Organization, “Laboratory quality management system: handbook,” World Health Organization, 2022.

[5] A. Chen, J. Choi, and J. Wang, “Retrieval-Augmented Generation for Large Language Models: A Survey,” *arXiv preprint arXiv:2312.10997*, 2024.