**AI-Enhanced Database Management System**

Software & Thesis submitted to Swansea University in Partial Fulfilment for the Degree of Master of Science in Computer Science from the Department of Computer Science.

**Submission date: September 2024**

**Mutasim El-Khidir (2254626)**

**Abstract**

While databases and their contents are crucial to any business and its operations, users often need training, make mistakes, and may feel uncomfortable using a particular database, as databases are not all created equal nor are intuitively navigable and/or well-designed.

Arguably, the internet is a collection of databases, and the browser is a user interface to access these databases; while internet browsing has become intuitive, other databases and their interfaces are not. Designing a Database Management System (DBMS) that is intuitive to use and user-friendly may not be easily achieved in specialised fields, i.e. healthcare.

Integrating an Artificial Intelligence aid into the user experience may limit and/or eliminate the user’s interaction with massive, complex, newly created or poorly designed databases or systems in the first place. A predefined set of instructions triggered by a written or dictated command from the user can help load, navigate, validate, visualise, and manipulate the data.

The solution is envisioned as a search field interface, allowing users to type or dictate queries in their natural language. These queries are then translated into instructions to be applied to multiple databases and their content. As a result, the user does not have to interact directly with poorly designed or newly created databases.

Advanced solutions harnessing the power of machine learning are explored to automate the process while respecting the research's available time frame.

**Declarations & Statements**

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

This dissertation is the result of my own independent work, except where otherwise stated. Other sources are acknowledged by giving explicit references.

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

**Mutasim El-Khidir**

XX/09/2024

**Table of Contents**

Page:

1. Abstract 2
2. Declarations & Statements 3
3. Table of Contents 4
4. Figures, Tables, Abbreviations & Symbols 5
5. Introduction
6. Background
7. Literature review
8. Software: AI Assistant (Version 2.2)
9. Evaluation
10. Discussion
11. Conclusion
12. References

**Figures, Tables, Abbreviations & Symbols**

**Introduction**

Evidence shows that employees waste considerable time operating database systems for administrative and operational purposes. This avoidable inefficiency has a negative rippling effect on all industries, with very few exceptions. (Admin Matters: The Impact of NHS Administration on Patient Care, n.d.)

In a labour—and capital-intensive industry like healthcare, operational efficiency saves lives before money. Reducing administrative costs (time) means serving more patients without hiring more staff, increasing staff satisfaction, and saving money.

For example, doctors are not trained to use database systems as part of their educational curriculum. Even if software usage is included in the academic material, a new release or update may require new learning material and training. A newly hired doctor might want to extract patient information during a meeting in a hypothetical scenario. However, if the doctor cannot effortlessly/intuitively use the available database systems, the doctor and patient will suffer.

Well-designed database systems are intuitive to use. However, creating an intuitively navigable database and information system that meets its inception purpose requires ingenuity, time, money, and all sorts of resources that are not readily available, especially in an industry like healthcare (NHS). This is evidenced by the data losses reported during the COVID-19 pandemic. (Kelion, 2020)

Designing an intuitive user interface requires a deep understanding of the end user. This is not easily achieved when factoring in user perception to visual representation, pattern recognition, preferences, and other elements affecting the user experience while using a database.

Siri, Google Assistant, and Alexa create an experience that feels intuitive, natural, and effective, reducing the cognitive load on users. A similar setup may serve as a fallback solution for an uneducated user. Capitalising of the widespread knowledge in interacting with these assistants, including Chatbots.

**Background**

AI solutions in information systems and database management have been utilised on various levels for decades. Before the invention of Tone Dialling, human operators connected the phone lines manually after asking the user whom they wanted to speak with. Arguably, tone-dial technology was the foundation of sound recognition, and a telephone switchboard is a database management system, with the operator acting as a Database Administrator (DBA).

1952, Bell Labs created the first speech-recognition device (Audrey) to recognise spoken digits. Currently, one example of advancements in Natural Language Processing (NLP) is the commercialisation of AI voice assistants, which can understand many accented languages and perform complex tasks on behalf of the user.

Contemporary commercial application examples include chatbots, telephone directories (voice-activated), and other AI-collected and processed information, i.e., social media listeners. These applications have eliminated the need for a human operator to perform a highly predictable and automatable task, saving money and raising efficiency by reducing administrative workloads.

The same NLP technology used to generate profit for technology giants like Apple, Google, and Amazon can eliminate tedious and automatable tasks in most workplaces. By integrating NLP to translate automatically and action the user's query/task in natural language, workloads are reduced and eliminated.

Emerging AI solutions for database management harnessing powerful AI tools are commercialised to address the needs of a growing market with massive amounts of data that may benefit from a novel approach. Examples of commercialised products that perform a similar task are Amazon Aurora, Oracle Digital Assistant, Microsoft Azure SQL Database, and others.

However, the administrative time waste issue remains prevalent, whilst the available technology does not address the needs of a population that may seem unaware of the potential AI and NLP can bring to the workplace with Human Computer Interaction (HCI) in mind.

Professor Yannis Ioannidis said, “The database and HCI communities live on their separate islands with essentially no ferry line between them.” (Ioannidis, 1996) Academic material addressing this specific angle took much work to find. Nevertheless, searching by keyword has surfaced interesting material dating back to when the technology had not matured or reached today's levels by comparison. With much of the research being completed and Intellectually Protected by the tech giants for profitability, the released technologies will have addressed many aspects of this research. However, the scientific material and software remain proprietary.

The primary search tool used to find the literature is Google Scholar, which generated results from reputable academic sources. However, the research also considered tech companies and potential users. The keywords used in the literature search process have multiple variations and combinations, but the core terms were NLP, ASR, AI, DBMS, and HCI.

Finding scientific literature on the subject was challenging because companies hold proprietary rights over scientific research and the tools needed to achieve the project goals. While the proposal may be reinventing the wheel compared to the number of products advertised performing virtually similar functions. This research should provide a simplified version of the “mechanics” and underlying technology behind these products.

However, the literature found was enlightening. The possible applications and areas of AI that can enhance databases seem infinite. Meanwhile, the DBMS HCI element is widely ignored outside of commercial space. An example would be harnessing the power of Google Voice Assistant, Siri, or Alexa in a working environment. A hypothetical example would be a doctor pulling patient information through a voice command in natural language, then having all the information the doctor needs at that time presented to the doctor. The impact of an efficient system saves lives in an industry like healthcare. (Healthcare UX: When UX Hurts And Even Kills)

The research papers examined demonstrate the progress made in the relevant fields of the ongoing research, while highlighting the need and potential for utilising these technologies in various scenarios to raise efficiency in most industries. While healthcare has been repeatedly mentioned and researched in this paper, it is the most appropriate example, while actual use cases extend beyond healthcare.

**Literature review**

**Natural language processing: an introduction** (Nadkarni et al., 2011)

As a critical technology to be used in this research, a good starting point would be to understand the mechanics of NLP, its limitations, and set expectations.

The paper provides background information on how natural language processing evolved in the previous century, highlighting historical milestones and explaining the issues and solutions used. Different approaches and technologies work together to enable a computer to understand human natural language as input and react to it.

While the research dates to 2011, it has pointed out solutions and the logic behind them as an introduction to NLP, arguably contributing to the newer methods in use today (Neural Networks). While this is outdated by modern standards, it highlights the rapid progress in computer science within two decades. Nevertheless, the research is informative from reputable institutions with quality references.

As NLP matures, with open-source solutions available for use in this research (LangChain), understanding its capacity and abilities highlights an opportunity to research further audio-triggered commands actioned by nonhuman subjects.

Overall, the research is of high quality, mentioning technologies still in use or have arguably paved the way for more efficient solutions, like Neural networks, which were available and researched when this paper was published.

**Literature Review on Automatic Speech Recognition** (Ghai & Singh, 2012)

As a subfield of NLP, automatic speech recognition (ASR) is arguably the technology that enables the fastest way to communicate ideas with a computer until telekinesis is discovered.

The paper explains the multiple approaches and technologies used to correctly classify and convert human speech (voice) into text, detailing the differences between the applied solutions and speech processing stages and mentioning various tools.

The paper highlighted the limitations of developing an ASR system, considering factors like training materials, speed, accuracy, and linguistics, some of which are still present a decade after the research was published. It also mentioned many use cases of ASR related to the research being conducted.

The paper often mentions Neural Networks without properly explaining how and why they contribute to ASR. Thus, the functions performed by the neural network remain ambiguous, rendering it a “black box” solution to the problem.

**The Integration of Artificial Intelligence into Database Systems (AI-DB Integration Review)** (Austine et al., 2023)

This publication considers a broader aspect of AI's potential contribution to Databases, focusing on harnessing AI abilities to design and manage databases. AI can bring a new perspective to the designer, as AI may have an advantage in pattern recognition and through shared knowledge generated by a neural network.

According to the authors, “AI-powered DB” should be automatically scalable and optimised, which should improve database performance in terms of running queries and future-proofing a database from scalability difficulties.

The authors have addressed the HCI element of databases under the “Intelligent Database Interface (IDI)”, inexplicitly reaching a similar conclusion as per this research, where NLP is a possible solution to the HCI problems users face with Databases.

The paper considers many aspects AI can bring without thoroughly examining each avenue, which is predictable due to AI's seemingly endless potential. This renders the paper more of a review of AI potential in databases.

**AI Meets Database: AI4DB and DB4AI** (Li et al., 2021a)

The research examined the potential AI brings to Databases and vice versa. The research recognises that database management systems have evolved separately from AI, and some techniques can be exchanged to improve the performance of AI and Databases together and separately.

The research dives into the details of database management operations and how they can be improved by automating tasks and optimising queries through various machine-learning techniques. Furthermore, the paper highlighted the difficulties AI developers face when building AI, as the training process requires data to be governed with techniques found in database management systems.

The paper expands further in another version, “Database Meets Artificial Intelligence: A Survey,” which has even more details on how AI and databases can exchange benefits. While the research is well-detailed, it ignores the HCI element of databases and AI, focusing only on software performance-related challenges and forgoing the human element.

However, the improvements AI will bring to query processing will ultimately affect the user experience, assuming AI will be able to construct efficient queries from natural language input and will tune databases without human interference, thus reducing administrative time, as per this research’s goals.

The research is of the highest quality, as would be expected from computer science professors at highly prestigious universities. However, it is not easily read due to the jargon associated with advanced sciences that are very new and unheard of outside the industry.

**XuanYuan: An AI-Native Database** (Li et al., n.d.)

The research paper describes various tasks, challenges, and future expectations for DBMS and the DBA. It suggests various AI integration levels to solve the present and predicted issues for databases and database management systems. The author created five levels of AI integration within databases, each addressing functions and problems that are progressively more complicated—reaching a point of automation resembling machine awareness or science fiction.

The author examined current research and solutions relevant to the mentioned challenges and suggested a new DBMS, “XuanYuan”. An AI-native DBMS functions at "level 5" by the author's standards. No adequate explanation was provided on how it could be achieved.

The paper explained in detail many of the processes in a DBMS, which created expectations, some of which are being assessed. NLP integration discussed in this research would be level 1, per the authors' standards. The authors' suggestions all pointed to AI in general without mentioning the mechanics of how AI can perform a specific function, like designing, which would require a form of consciousness and purpose in AI unless prompted by a human.

**The Architecture Of An Active Data Base Management System** (McCarthy, n.d.)

The research paper investigates how an active DBMS should perform in its environment, focusing on the mechanics of how an active DBMS would react if certain conditions were met. Various complications were examined, and the most interesting was the nested transactions.

The paper describes the work of an ongoing project (HiPAC) designed to handle DBMS operations under a set of rules, “Event-Condition-Action” (ECA). Thus, decisions are made without a user prompt, following a preset logic in handling certain events.

The paper provides viable concepts for the ongoing NLP in DBMS research, as it demonstrates a system of processing predefined actions with predictable complications that will appear in this research at later stages. More specifically, accessing multiple databases and the complex queries dictated to the search bar would require a governing system that resembles the one described as an active DBMS.

Furthermore, a search function connected to multiple databases that take natural language as input is an active DBMS, taking advantage of what has been released following this research paper.

Considering the publishing date, the work that the author describes is now an integral part of many intelligent systems, including voice assistants like Alexa and Siri. These assistants can make many decisions in active time without user interference, raising privacy concerns regarding data collection and processing.

While the research may seem outdated, considering the simplicity of the working and publishing time, it explains how a DBMS can make decisions autonomously in active time. Considering the possible complications, data types, and decisions that could be made, it is no surprise that the HiPAC project is still ongoing.

**The rise of artificial intelligence in healthcare applications** (Bohr & Memarzadeh, 2020)

Following the initial example of healthcare, this research paper examines the applications and possible applications in the healthcare field specifically. With a broader overview of AI's effects on healthcare in general, the research examines the application of AI's contribution to genetics all the way to the subject's environment.

The paper lists possible applications for NLP in healthcare without expanding on the details. However, the applications mentioned can add value to this research as they underline other issues where NLP may assist decision-making besides reducing administrative workloads. An example would be NLP lie detection, which is ongoing research that should impact situations like drug-seeking behaviour in healthcare and general interactions with civil service.

The paper is a chapter in the book titled (Artificial Intelligence in Healthcare by Adam Bohr and Kaveh Memarzadeh), which serves as an introduction to the following chapters that are more detailed in AI applications. The authors are arguably healthcare experts with limited computer science knowledge. Thus, the paper should serve as a market study of AI-enhanced DBMS research.

**Machine Learning for Databases** (Li et al., 2021b)

The research paper describes the different types and levels of machine learning and their use case scenarios for databases. While machine learning is considered by many to be an important part of AI, machine learning is a broad term with sub-categories, each with a different use case.

As the authors describe, machine learning applications may seem distant from HCI but will ultimately cascade down to the user experience and interaction with the database. This is

confirmed by the previously reviewed research by the same authors regarding AI in databases (AI4DB and DB4AI).

While the authors are experts in the field, the language used can be difficult to decipher, which is expected considering the complexity of the subject. They have provided advanced use cases that emerged recently in the market by tech giants like Google, with the newly released database systems that utilise generative AI as part of their functions, i.e. (Google's AlloyDB AI).

**Literature findings and conclusion:**

Finding scientific literature on the subject was challenging because companies hold proprietary rights over scientific research and the tools needed to achieve the project goals. While the proposal may be reinventing the wheel compared to the number of products performing virtually similar functions, going into detail should prove otherwise on many levels or show the unique selling point of the project in hand.

However, the literature found was enlightening. The possible applications and areas of AI that can enhance databases seem infinite. Meanwhile, the DBMS HCI element is widely ignored outside of commercial space. An example would be harnessing the power of Google Voice Assistant, Siri, or Alexa in a working environment. A hypothetical example would be a doctor making the rounds and speaking with patients, does not have to go back to log the interactions nor the observations as they are processed in the background in real-time, with less room for human error.

The research papers examined demonstrate the progress made in the relevant fields of the ongoing research while highlighting the need and potential for utilising these technologies in various scenarios to raise efficiency in most industries. While healthcare has been repeatedly mentioned and researched in this paper, it is the most appropriate example, while actual use cases extend beyond healthcare.

**Software: AI Tool for Databases (Version 1.0)**

Initially imagined as a search bar with space to type or dictate a query in natural language, the following Graphic User Interface (GUI) was the initial idea and product:

A rectangular black and white rectangular object

Description automatically generated

The above interface would be insufficient for generating text responses or to have to option to read the answer instead of hearing it. Unless it opens other windows and has more interactive GUI functions that are time-consuming to develop and test, considering the research time frame. Thus, the below currently used GUI was selected:

A screenshot of a computer

Description automatically generated

While visually less attractive, the core functionalities are easier to include and test in the above format, considering the ongoing testing and development of the functionalities in the initial design and the time needed to utilise sophisticated GUI tools.

The below diagram acted as a simplistic process/data flow of the search bar:



Many visualisations were created in the development of the program, with various levels of abstraction and detail. Some were created with Software Engineering principles in mind, as in the Unified Modelling Language (UML) and Class-Responsibility-Cards (CRC) diagrams and visualisations, while other visualisations serve as a guide and provide general explanations.

The graph below should express the general logic/functions and procedures of the current program iteration in a simplistic format:

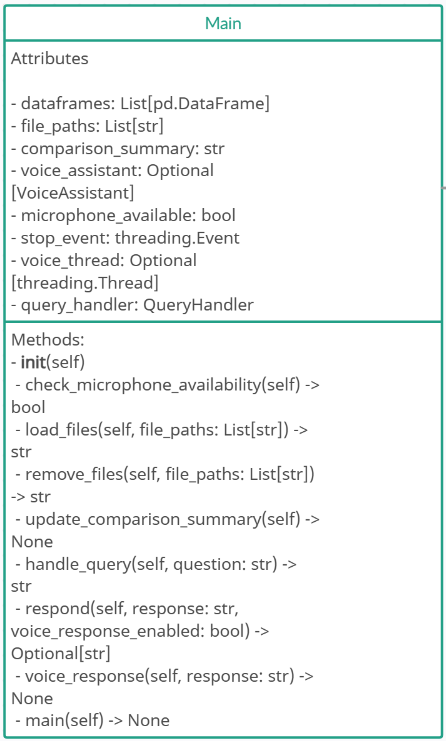
A diagram of application software

Description automatically generated

The current program iteration is a graphical user interface (GUI) based application developed using the Tkinter framework, designed to facilitate file handling, data comparison, query processing, and voice-assisted interaction. At its core, the application allows users to load multiple data files, compare their contents, query the data, and receive responses through text or voice. The modular structure of the application ensures ease of use, flexibility, and expandability, making it a versatile tool for data-centric tasks.

Core Architecture and Functionality

The application's primary class is the Main class, which encapsulates file management, data comparison, voice assistance, and query processing functionality. The application is designed to handle multiple data files simultaneously, utilising the (pandas) library to read and manage these datasets. Each file is loaded into a DataFrame and serves as the foundational structure for performing comparisons and answering queries



.

A screenshot of a computer program

Description automatically generated

File Handling and Data Processing

The application’s file management capabilities are enabled through a FileHandler class, which reads and parses files into DataFrame objects. This structure allows for efficient manipulation and comparison of data. Users can load multiple files and remove them as needed, with the internal data structure dynamically updating to reflect the changes. This flexibility is crucial when the user may need to analyse large datasets from disparate sources.

Query Processing

A key feature of the application is its ability to process user queries related to the loaded datasets. The QueryHandler class is responsible for parsing and responding to these queries. If only a single dataset is loaded, the query is directed toward that file. In cases where multiple files are loaded, the system bases its responses on the comparison summary. This query functionality enables users to interact with the data more intuitively, asking direct questions and receiving relevant answers in real-time.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Data Comparison

When multiple files are loaded, the application leverages a DataComparer class to compare the datasets. This component generates a comparison summary, providing insights into similarities and differences across the data files. This feature is particularly useful in research scenarios where multiple datasets must be analysed concurrently, allowing users to draw comparisons and make informed decisions based on the compiled data.

Voice Assistance and Multithreading

Incorporating voice-based interaction, the application checks for microphone availability upon startup. If a microphone is detected, a VoiceAssistant object is initialized, allowing users to ask queries verbally and receive responses through text- to-speech. This feature enhances the accessibility of the tool, especially in environments where voice interaction is preferable. The application employs multithreading to handle voice responses, ensuring that the GUI remains responsive while voice interactions occur asynchronously. This design prevents blocking of the main application thread, allowing for a smooth user experience.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

User Interface

The graphical interface is managed by the GUIHandler class, which handles the user’s interaction with the system. The interface is intuitive and user-friendly, enabling users to load files, ask questions, and receive responses with minimal effort. The design focuses on accessibility, allowing both text-based and voice-based interactions.

Program Flow and Modularity

Upon execution, the program follows a clearly defined flow, beginning with the initialisation of the App class. Once instantiated, the main() method is called launching the GUI and initialising the core functionality. The modular design of the program, with distinct components for file handling, data comparison, query processing, and voice interaction, ensures that the application can be easily extended and adapted to different use cases. The reliance on external classes, such as FileHandler, DataComparer, QueryHandler, and VoiceAssistant, promotes a separation of concerns, making the program maintainable and scalable.

In summary, the application is a comprehensive data processing tool capable of handling multiple datasets, performing complex comparisons, and enabling intuitive interaction through both text and voice. Its modularity, ease of use, and adaptability make it an asset in research environments where efficient data analysis and interaction are paramount.

**Evaluation**

**Discussion**

**Conclusion**

**References**

Admin Matters: The Impact Of NHS Administration On Patient Care. (n.d.). The King’s Fund. https://www.kingsfund.org.uk/insight-and-analysis/long-reads/admin-matters-nhs-patient-care

Austine, O., Olalekan, M., Babatunde, Y., & Lateef, O. (2023). The Integration Of Artificial Intelligence Into Database Systems (Ai-Db Integration Review).

Bohr, A., & Memarzadeh, K. (2020). The rise of artificial intelligence in healthcare applications. In Artificial Intelligence in Healthcare (pp. 25–60). Elsevier. https://doi.org/10.1016/B978-0-12-818438-7.00002-2

Ghai, W., & Singh, N. (2012). Literature Review on Automatic Speech Recognition. International Journal of Computer Applications, 41(8), 42–50. https://doi.org/10.5120/5565-7646

Ioannidis, Y. E. (1996). Visual user interfaces for database systems. ACM Computing Surveys, 28(4es), 137-es. https://doi.org/10.1145/242224.242399

Kelion, L. (2020, October 5). Excel: Why using Microsoft’s tool caused Covid-19 results to be lost. BBC News. https://www.bbc.co.uk/news/technology-54423988

Li, G., Zhou, X., & Cao, L. (2021a). AI Meets Database: AI4DB and DB4AI. Proceedings of the 2021 International Conference on Management of Data, 2859–2866. https://doi.org/10.1145/3448016.3457542

Li, G., Zhou, X., & Cao, L. (2021b). Machine Learning for Databases. The First International Conference on AI-ML-Systems, 1–2. https://doi.org/10.1145/3486001.3486248

Li, G., Zhou, X., & Li, S. (n.d.). XuanYuan: An AI-Native Database.

McCarthy, D. R. (n.d.). The architecture of an active database management system.

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: An introduction. Journal of the American Medical Informatics Association, 18(5), 544–551. https://doi.org/10.1136/amiajnl-2011-000464