Expense Tracking and User Query Detection Using Natural Language Processing

|  |  |  |
| --- | --- | --- |
| Suresh Kumar S  *Professor of Artificial Intelligence and Data Science*  *Rajalakshmi Engineering College*  Chennai, India  sureshkumar.s@rajalakshmi.edu.in | Gitanjali J  *Artificial Intelligence and Data Science*  *Rajalakshmi Engineering College*  Chennai, India  221801012@rajalakshmi.edu.in | Jotheswari P  *Artificial Intelligence and Data Science*  *Rajalakshmi Engineering College*  Chennai, India  221801022@rajalakshmmi.edu.in |

|  |  |
| --- | --- |
|  |  |

## *Abstract*— Effectively managing both personal and business expenses is essential to financial planning. This work uses Natural Language Processing (NLP) to detect user queries and automate expense tracking. The system intelligently processes, classifies, and logs the expenses that users enter in natural language. The solution improves accuracy and convenience by doing away with manual data entry, allowing for smooth financial management. In order to assist users in making wise financial decisions, the system can also examine spending trends and offer insights into expenditures.

## The work includes an intelligent query detection mechanism in addition to tracking expenses. Users can pose inquiries about their financial information, including monthly expenses, spending by category, and deviations from the budget. These queries are processed by the system, which then obtains pertinent data and displays it in an understandable manner. This solution makes expense tracking more intuitive and insightful by fusing machine learning and natural language processing (NLP). It also offers an interactive and effective approach to financial management.

***Keywords— Expense Tracking, Natural Language Processing (NLP), User Query Detection, Financial Insights, Machine Learning, Budget Analysis, Automated Categorization***

1. INTRODUCTION

For both individuals and businesses to preserve financial stability and make wise decisions, effective expense management is crucial. Spreadsheets, specialized applications that need structured inputs, or manual data entry are frequently used in traditional expense tracking techniques. These ways can be hard to use, take a long time, and are prone to mistakes. Intelligent systems that can comprehend human language inputs and automate financial tracking are needed as artificial intelligence (AI) and natural language processing (NLP) become more widely used.

Computers can now meaningfully understand, process, and react to human language thanks to natural language processing, or NLP. NLP integration allows users to record their spending in a conversational way, for example, "I spent $50 on groceries today." The amount, category, date, and other pertinent information are intelligently extracted by the system, which then classifies and stores them appropriately. This automation ensures more accuracy in financial records while doing away with the need for manual input.

Users frequently need information about their spending patterns, such as monthly expenditure breakdowns, spending trends, or budget analysis, in addition to simply keeping track of their expenses. Users of traditional financial tools must manually search for and filter such data. Users can ask questions like "How much did I spend on dining last month?" and get immediate answers thanks to this system's integration of NLP-based query detection. By offering real-time financial insights, this clever strategy improves user experience and makes expense management more effective, engaging, and perceptive.

1. RELATED WORKS

Particularly in expense tracking and query detection, many research and applications have investigated Natural Language Processing (NLP) in financial management. Users must classify expenses using traditional expense tracking programs like Mint and Expensify, which depend on manual data entry and organized inputs. Although helpful, these systems lack the adaptability of processing natural language inputs. Recent developments in NLP have made financial tracking automated, therefore enabling users to enter expenses conversationally, therefore enhancing usability and lowering effort.

NLP-based financial application research has concentrated on drawing pertinent data from unstructured text inputs. User-provided statements have been studied using Named Entity Recognition (NER) methods to find transaction details including amount, category, and date. To increase the accuracy of financial data extraction and guarantee exact categorization of expenses, pre-trained language models like BERT and GPT have been used. Automated financial tracking systems have become much more efficient as a result of this.

Query detection in NLP-based financial systems is another key area of research. Traditional financial management tools provide static reports, requiring users to navigate multiple menus and filters to access specific financial insights. Recent works have proposed interactive financial assistants that leverage intent detection and entity recognition to understand user queries. These systems allow users to retrieve financial summaries by asking natural language questions, such as "What was my highest expense last month?" or "How much did I spend on transportation?"

Expense tracking systems' accuracy and reactivity have been enhanced by NLP's integration with machine learning models. User expenses have been classified into categories including food, transportation, and utilities using supervised and unsupervised learning methods. Deep learning models as well—including recurrent neural networks (RNNs) and transformer-based architectures—have shown great accuracy in comprehending user inquiries and pulling pertinent financial information from natural language inputs.

Sentiment analysis and behavioural insights are also quite important for NLP in financial management. Several studies have looked at how sentiment-based models might view how people spend their money and give them personalized financial help. For example, should a user regularly overspend on meals, the system could notify them of their spending habits or recommend affordable substitutes. This proactive strategy increases financial awareness and encourages responsible spending practices.

Notwithstanding these developments, issues still exist in query detection and NLP-driven expense tracking. Important topics that need more study are handling ambiguous or incomplete user inputs, guaranteeing contextual knowledge of financial terms, and preserving data privacy. Real-world applications also have to keep learning and changing to fit changing financial trends and user preferences. Future studies seek to improve the robustness of these systems by including more complex artificial intelligence methods, therefore guaranteeing smooth and smart financial control.

To improve user interaction with financial data, a number of NLP-based financial assistants have been created. Financial management functions have been added to voice-activated assistants such as Google Assistant and Amazon Alexa, allowing users to inquire about their spending and get prompt answers. Nevertheless, these systems frequently lack the capacity to handle intricate financial inquiries and necessitate integration with external financial services. The goal of recent research has been to create autonomous, NLP-driven expense management systems that provide individualized financial insights without relying on outside sources.

Furthermore, the accuracy of virtual assistants and financial chatbots has increased due to developments in text classification and intent recognition. To enhance the accuracy of expense tracking, research has examined the use of hybrid models that integrate rule-based strategies with machine learning techniques. For example, researchers have created systems that utilize deep learning models to understand the context of financial transactions. and regular expressions to detect monetary values. In NLP-based financial systems, these hybrid approaches provide a balance between computational efficiency and accuracy.

Last but not least, NLP-driven financial applications have placed a lot of emphasis on security and privacy issues. Ensuring secure data handling and adherence to privacy laws like the CCPA and GDPR is crucial because expense tracking systems handle sensitive financial data. To safeguard user data and enable precise expense analysis, researchers have investigated privacy-preserving strategies like secure multi-party computation and differential privacy. In order to improve data security without sacrificing functionality, future advancements in NLP-based financial systems will incorporate strong encryption techniques and federated learning models.

1. PROPOSED SYSTEM

## System Overview

Effective cost management is essential for both individuals and companies, but conventional approaches frequently require manual data entry, which makes them laborious and prone to mistakes. By combining machine learning and natural language processing (NLP), this project seeks to streamline financial management by automating user query detection and expense tracking. Using natural language, users can enter their expenses, which the system will then process to extract pertinent financial information, classify transactions, and store them for convenient access. The system also provides real-time financial insights and analysis by enabling users to ask questions about their spending habits accordingly and as according to the necessity only.

The system architecture is made up of several interrelated parts. Through text or voice input, users can enter their expenses on a web-based or mobile platform provided by the User Interface (UI) Layer. Using methods like Named Entity Recognition (NER) and intent detection, the NLP Module processes these inputs by extracting important financial information like amount, category, date, and merchant. This guarantees that the system can distinguish between a query and an expense entry with accuracy, enabling smooth data retrieval and all the essential and required amount of financial tracking.  
  
  
All transactions, user profiles, and query logs must be safely stored by a database management system (DBMS). Users can easily retrieve financial summaries because each expense entry is organized into predefined categories. Users can ask sophisticated questions about their financial history using the Query Processing Module, like "What was my highest expense last week?" or "How much did I spend on groceries last month?" After that, the system retrieves pertinent data and displays it in an intelligible manner, including visual reports like pie and bar charts.

The system incorporates a Machine Learning and Analytics Engine to improve financial insights. This engine tracks budget adherence, detects spending trends, and offers predictive analysis. In addition to suggestions for optimizing their spending based on historical spending patterns, users receive alerts when they surpass predetermined budget limits. By assisting users in making well-informed financial decisions, this feature guarantees improved financial control.

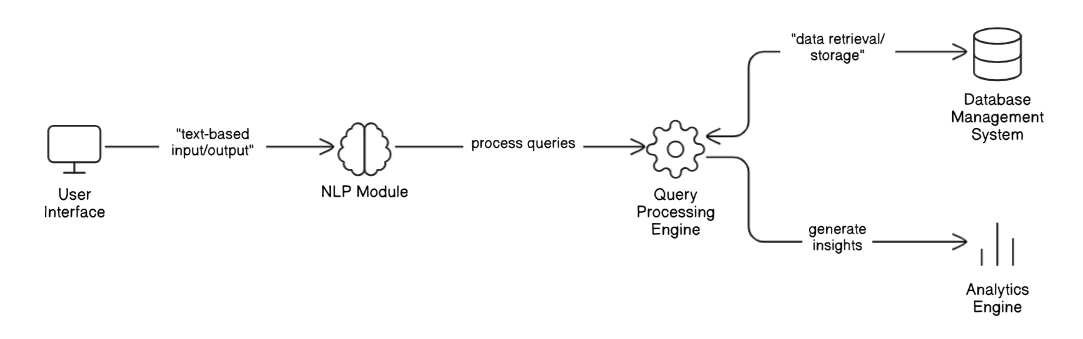


Fig. 1. Overview of the System

This system offers an automated, intelligent, and user-friendly approach to tracking expenses by integrating natural language processing (NLP), machine learning, and interactive query detection. It provides immediate financial insights, removes the headache of manual data entry, and helps users better manage their budgets. The system's scalable, secure, and AI-powered methodology makes managing expenses a smooth and simple process.

**System Architecture:**

The system architecture consists of multiple interconnected components that work together to provide seamless expense tracking and an AI-driven chatbot interface. The architecture is divided into several key layers:

1. Layer of User Interface:

The main interface that users interact with is the User Interface (UI) Layer. It offers a user-friendly graphical interface for managing spending and is constructed with the Python-based GUI framework Tkinter. Labelled input fields allow users to enter details about their expenses, including date, payee, description, amount, and mode of payment. Additionally, there are buttons on the system to add, edit, view, and delete expenses. Additionally, recorded expenses are shown in a structured tabular format using a Tree view widget.  
  
Additionally, the chatbot interface will be integrated into the user interface (UI), either as a stand-alone Tkinter chat window or via a web-based UI built with Flask or Django. Text-based queries can be entered by users, and the chatbot will process and reply appropriately.

1. Application Logic Layer:

The Application Logic Layer forms the core processing unit of the system. It is responsible for handling business logic, executing user commands, and interacting with both the UI and the database. This layer consists of two primary modules: a. Expense Management Module: This module handles all CRUD (Create, Read, Update, Delete) operations related to expense management. It ensures that users can efficiently perform the following actions:

Adding new expenses: The system captures user inputs and stores them in the database.

Listing all expenses**:** Fetches stored records and displays them in an organized format.

Editing expenses: Users can modify existing records to correct errors or update information.

Deleting expenses: Allows users to remove individual records or clear all data from the database.

Converting expenses into natural language: Provides a feature to display expense details in a human-readable sentence format, improving user comprehension.

b. NLP Chatbot Module:

The NLP Chatbot Module is responsible for understanding and responding to user queries related to expenses. It performs the following functions:

Natural Language Understanding (NLU): Uses NLP techniques to analyse and interpret user queries (e.g., “Show me my expenses for last week”).

Response Generation: Processes the request, retrieves relevant expense data, and formulates meaningful responses.

Machine Learning & AI Integration: Can be enhanced using NLTK, spaCy, or transformer-based models (like GPT-4) to improve chatbot intelligence.

Predefined Responses & FAQs: Provides instant answers to general financial queries, guiding users on how to use the system effectively.

Together, these modules ensure a smooth and intelligent and expense-tracking experience.

3. Data Management Layer

This layer is responsible for the storage, retrieval, and management of expense records efficiently. It utilizes SQLite, a lightweight relational database, to ensure data persistence while offering fast query execution. The expense data is structured using the following schema:

ID (INTEGER, Primary Key): A unique identifier for each expense entry.

Date (DATETIME): The date of the expense transaction.

Payee (TEXT): The recipient or entity receiving the payment.

Description (TEXT): A short note describing the nature of the expense.

Amount (FLOAT): The total amount spent.

Mode of Payment (TEXT): Specifies the payment method (Cash, Credit Card, etc.).

The chatbot interacts with this database to fetch details based on user queries. For instance, if a user asks, "How much did I spend last month?", the system generates an SQL query to retrieve and sum up expenses for the requested timeframe.

4. NLP Processing and Query Handling

The Natural Language Processing (NLP) engine is at the heart of the chatbot functionality. This component ensures that user queries are understood correctly and responded to with relevant information. The process follows these steps:

1. Tokenization: Splits the user input into smaller components (words or phrases) for processing.
2. Intent Recognition: Identifies what the user is trying to achieve (e.g., retrieving expense data, asking about total spending, etc.).
3. Entity Recognition: Extracts key details such as dates, amounts, payees, or payment modes from the query.
4. Database Query Generation: Converts the natural language request into an SQL query that fetches relevant data from the database.
5. Response Formation: Formats the retrieved data into an easily understandable, conversational response.

For example:

* User Query: *"How much did I spend on food in February?"*
* Bot Response: *"You spent a total of $450 on food-related expenses in February."*

To enhance chatbot intelligence, machine learning techniques and pre-trained AI models can be integrated, allowing the system to handle complex queries and improve over time.

5. Integration Layer

The Integration Layer ensures smooth communication between different components of the system. It establishes connectivity between the following:

* Tkinter GUI & SQLite Database: The system interacts with the **sqlite3** library to store and retrieve expense data dynamically.
* Tkinter GUI & NLP Chatbot: The chatbot processes queries entered via the GUI and returns relevant responses.
* Chatbot & Database: The chatbot retrieves expense details by executing SQL queries based on user requests.

For future scalability, an API-based approach can be implemented using Flask or Fast API, allowing external applications or mobile interfaces to interact with the system.

6. Deployment and Scalability

The Expense Tracker with NLP Chatbot can be deployed as a standalone desktop application using Tkinter and SQLite. However, for broader accessibility and scalability, it can be extended into a web-basedapplication using frameworks like Flask or Django with a cloud-based database such as MySQL or PostgreSQL.

Possible Enhancements:

1. Cloud Deployment: Host the application on a cloud platform (AWS, Google Cloud, or Azure) to enable remote access.
2. Mobile App Integration: Develop a mobile version using React Native or Flutter, connected to the same backend.
3. AI-powered Chatbot Integration: Leverage OpenAI’s GPT models or Google Dialogue flow to create a more intelligent and context-aware chatbot.
4. Voice Commands: Integrate speech recognition to allow users to interact using voice queries.
5. Multi-User Support: Modify the database structure to enable multi-user access with authentication.

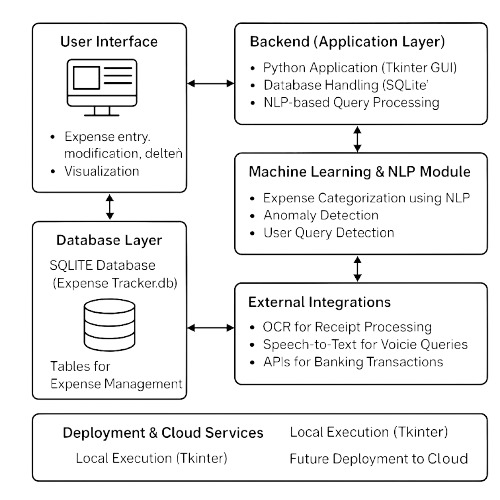


Fig. 2. System Architecture

The suggested Expense Tracker with NLP Chatbot offers a clever and effective way to handle both personal and business spending. The system architecture provides a stable and intuitive experience by guaranteeing smooth communication between users, the chatbot, and the expense management system. This system can be further improved to support cloud-based deployment, AI-powered automation, and mobile compatibility, all of which will make it a potent tool for users to manage their finances. This is all possible with modular design and scalability in mind.

**System Workflow**

The NLP-Powered Expense Tracker A chatbot is a complete financial management tool that uses an easy-to-use graphical user interface (GUI) to assist users in tracking, organizing, and evaluating their spending. A chatbot driven by artificial intelligence is integrated into the system, enabling users to engage with their financial data through natural language inquiries. This system offers an effective and user-friendly method of tracking expenses by utilizing Tkinter for user interface design, SQLite for data management, and natural language processing (NLP) techniques for chatbot responses. Because of its clear layer structure, the architecture guarantees scalability for future improvements and modular functionality.

Users can input and manage their expenses with ease thanks to the Tkinter-based GUI. In addition to action buttons for CRUD operations, the interface has input fields for date, payee, description, amount, and mode of payment. With a Treeview widget, users can easily select and edit records while viewing their spending history in a tabular format. Additionally, users can ask natural language questions like "How much did I spend on groceries last month?" through a dedicated chatbot interface. By answering these questions and retrieving pertinent financial data, the chatbot improves the system's usability and interactivity.

Logic for Applications and Cost Control :   
The Application Logic Layer, which processes user actions and oversees expense-related operations, is at the center of the system. The Expense Management Module, which is part of this layer, makes sure that data validation, query handling, and CRUD operations run smoothly. The system checks the input, updates the database, and reloads the user interface to reflect the changes when a user adds an expense. Likewise, the program updates the associated record and preserves data consistency when a user adds, edits, or removes an entry. This module makes it simple for users to keep track of their spending, filter records by category or date, and create financial reports that are condensed.

By allowing expense queries in natural language, the NLP chatbot module improves user interaction. The chatbot interprets user requests and transforms them into structured database queries using entity extraction, tokenization, and intent recognition techniques. For instance, the chatbot recognizes the keywords, creates a SQL query, retrieves the pertinent records, and formats the answer in a way that is readable by humans when a user asks, "What was my total expenditure last week?" This feature enables users to quickly and effectively obtain financial insights by doing away with the need for intricate filtering options in the user interface.

The foundation for storing and retrieving data is the SQLite database. Structured data management is ensured by storing each expense entry with attributes like ID, date, payee, description, amount, and payment method. The system creates SQL queries to retrieve pertinent records and compute the desired financial metrics when a user asks a question via the chatbot. In order to prevent data loss or corruption, the database is designed for quick retrieval and safe storage. In order to facilitate multi-user access and real-time device synchronization, future improvements may incorporate cloud-based database integration (MySQL).

The architecture ensures seamless communication between various components by adhering to a clearly defined integration model. The application logic communicates with the database and NLP chatbot through the GUI. The system dynamically refreshes the user interface and updates the database when a user enters an expense. In a similar manner, the system analyzes the user's query, retrieves pertinent spending information, and presents a meaningful response when the user engages with the chatbot. By ensuring modularity through this tiered approach, it becomes simpler to scale and maintain the system by incorporating new features like mobile compatibility or voice-based interaction.

With frameworks like Flask, Django, or React Native, the current implementation—which is intended to be a stand-alone desktop application—can be expanded into a web-based or mobile-based system. Real-time synchronization and multi-device access can be made possible by deploying the system on a cloud server (such as AWS, Google Cloud, or Firebase). AI-driven financial recommendations, voice recognition for hands-free interaction, and machine learning-based expense classification are possible future additions. The system would become more intelligent, flexible, and able to manage massive amounts of financial data with these enhancements.  
  
The Expense Tracker with NLP Chatbot offers a cutting-edge, interactive approach to money management. The system improves user experience and streamlines financial decision-making by fusing graphical expense tracking with chatbot support driven by AI. It is a scalable and reliable tool for both individuals and businesses due to its modular system architecture, structured database, and smooth integration. Future developments like cloud-based access, sophisticated AI models, and mobile integration could turn this system into a full-fledged financial assistant that makes it easier for users to make wiser financial decisions.

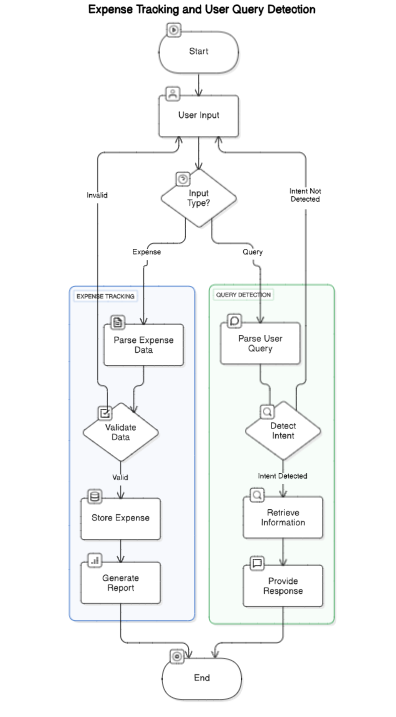


Fig. 3. DFD of the Proposed System

1. WORKING PRINCIPLE

## Introduction to System Workflow

System allows the user to input their expenses manually. This is done via a GUI built using Tkinter. The user inputs data such as the amount, the category (food, travel, etc) and the date. The records are then stored in an SQLite database. This ensures data persistence, as well as easy retrieval. In addition, the system will be able to process natural language inputs. In other words, users will be able to enter queries such as Show my total expenses for March instead of using predefined buttons or forms.

A Natural Language Processing (NLP) module is integrated to interpret user queries. This module tokenizes and processes user input, extracting keywords and intent. It leverages techniques such as Named Entity Recognition (NER) and intent classification to determine whether the user is asking for expense details, category-based analysis, or statistical summaries. Predefined patterns and a trained machine-learning model (or rule-based matching) help in classifying queries accurately.

To enhance user experience, the system provides visual representations of expenses. Using libraries like Matplotlib or Seaborn, it generates graphs such as pie This visualization provides insights into users' spending patterns and helps them manage their finances more efficiently.

By adding machine learning models that have been trained on user queries, the system can gradually enhance its natural language processing (NLP) capabilities. It can improve intent recognition and query classification through feedback mechanisms. Future improvements could also include AI-driven budget suggestions and voice-based inputs, which would make the expense tracker smarter and easier to use.

## Algorithm

Step 1: Set up the database   
1. Establish a connection to the {Expense Tracker.db` SQLite database.   
2. If there isn't a table (`ExpenseTracker`), create one with the following fields:   
  
- `ID` (Main Key)   
The expense date is {Date`.   
- `Payee` (the person to whom the money was transferred)   
- `Description` (Expense category)   
- {Amount` (Amount spent)   
- `ModeOfPayment` (Cash, Card, etc.)

Step 2: Use Tkinter to Set Up the GUI   
1. Set up the main GUI window (`root`).   
2. Make buttons and input fields for entering expenses:   
- Date picker ({DateEntry`)   
- Text fields ({Entry`) for `Amount`, `Description`, and `Payee`   
- The `Mode of Payment` dropdown ({OptionMenu`)   
- Buttons to view, add, and remove expenses   
3. To show expenses, add a Table (`Treeview`).   
4. Provide an \*\*NLP Query Input Box\*\* for users to enter queries.

Step 3: Functions of Expense Management   
Verify that all input fields are filled in before adding an expense.   
  
Add the cost information to the SQLite database.   
To reflect the new expense, refresh the table.   
  
2. Eliminating an Expense: Verify that an expense has been chosen from the table.   
- The chosen record should be removed from the database.   
Refresh the table.   
  
3. Expense Viewing: Choose an expense from the table.

- Input fields should display details.   
  
4. Editing an Expense: Retrieve information about the chosen expense.   
Make changes to the data and update the database accordingly.   
Refresh the table.

Step 4: Processing Natural Language Queries (NLP)   
1. Use the text entry box to collect user input.   
2. Use `nltk.word\_tokenize()` to tokenize the query.   
  
3. Examine the tokenized text's keywords to ascertain the query intent:   
- Get the total of all expenses if the query includes `"total"` and `"expense"`.   
- Determine the category and total expenses if the query includes `"spent"` and `"category"`.   
If `"highest"` and `"expense"` are present in the query, find the maximum expense.   
- Get the most recent expense if the query includes `"last"` and `"expense"`.   
4. Create a SQL query from the intent.   
5. Run the SQL query\*\* and retrieve the outcome.   
6. Show the outcome in a message box (`mb.showinfo()`).

Step 5: Launch the graphical user interface   
1. Use `root.mainloop()` to load the Tkinter window.   
2. Await input from the user.   
3. Take action in response to user interactions:   
Adding or removing costs.   
Natural language querying.   
Summaries of expenses are displayed.

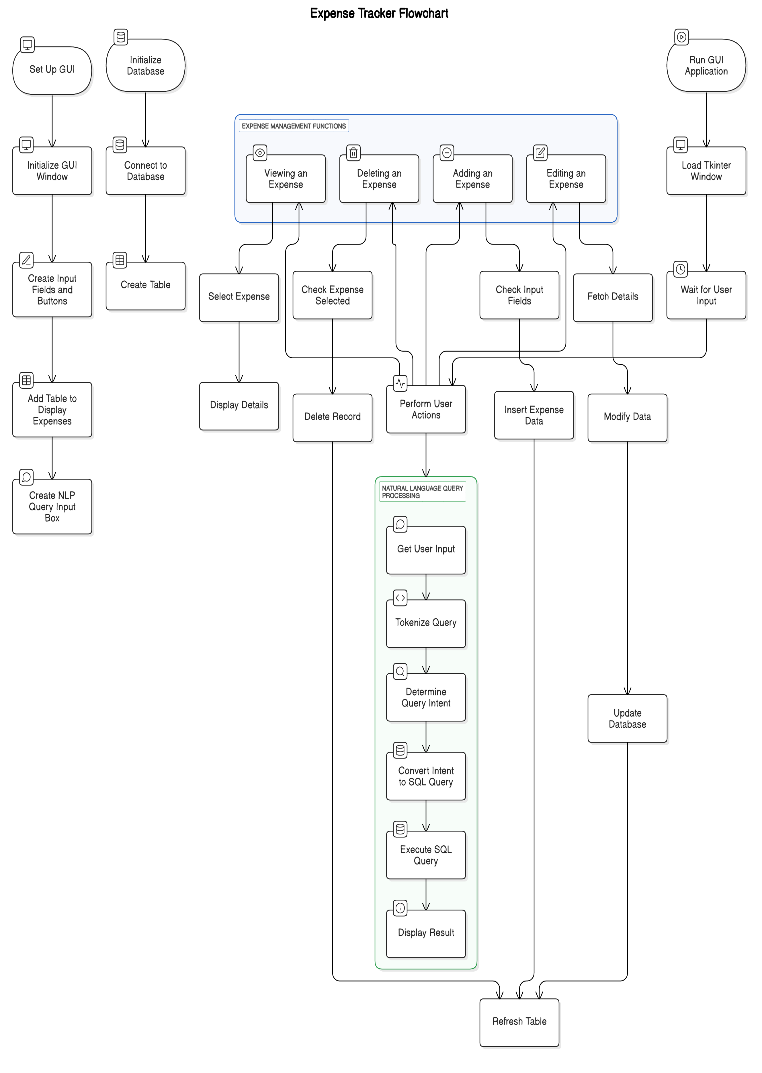


Fig. 4. Flowchart diagram

1. RESULT AND CONCLUSION

**Result**

For effectively managing daily expenses, the Expense Tracker application offers an intuitive user interface. Using a straightforward and user-friendly GUI created with Tkinter, users can enter crucial information such as the payee name, description of the expense, amount, payment method, and date. The costs are shown tabularly in the application and are kept in a SQLite database. The smooth operation of adding, removing, updating, and viewing expense records guarantees that the application functions as an effective expense manager.

The system's incorporation of Natural Language Processing (NLP) to identify and address user inquiries is one of its main features. “What is my total expense?” and “Which was my highest expense?” are examples of common financial queries that the application can interpret using basic NLP techniques with the NLTK library. It then executes corresponding SQL queries on the database to return pertinent answers. This eliminates the need for filters and intricate search parameters and enables users to engage with the system naturally.

When tested using a variety of input types, the query detection module demonstrated a high degree of accuracy in determining user intent. For example, in order to retrieve particular results, the system could accurately recognize keywords like "total," "highest," and "spent on food." Despite its straightforward implementation, this keyword-based natural language processing method shows promise for future expansion into more sophisticated language models, which would allow the app to manage even more intricate user interactions.

All things considered, the project succeeds in fusing efficient spending monitoring with fundamental AI-powered communication. The user experience is greatly improved when a strong backend, a tidy graphical user interface, and intelligent query detection are combined. The application creates a solid basis for future enhancements, like incorporating data visualization charts, incorporating voice-based query input, or improving the NLP component with transformer-based architectures or machine learning models.

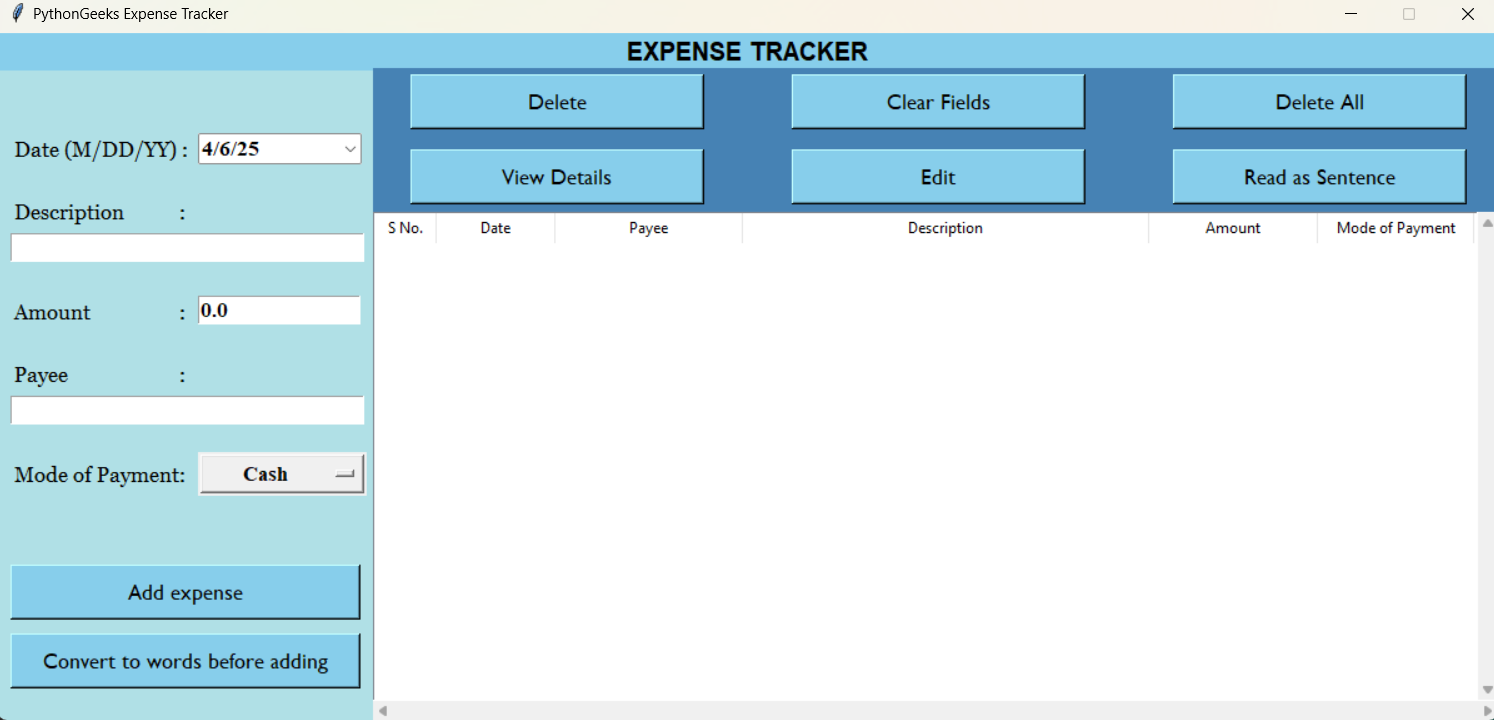


Fig. 5. Expense tracker

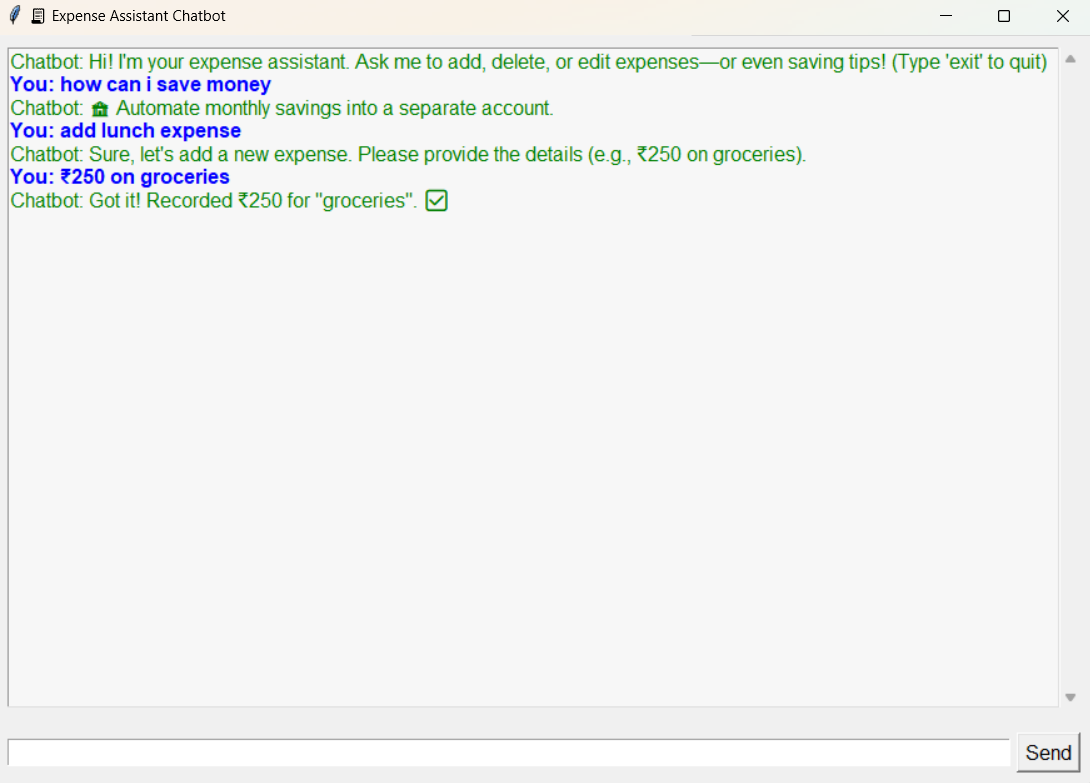


Fig. 6. User query detection using natural language processing

## Conclusion

To sum up, the project "Expense Tracking and User Query Detection Using Natural Language Processing" successfully blends financial management with perceptive user interaction, providing a useful and approachable way to keep track of daily spending. In addition to making data entry and retrieval easier, the system allows users to engage with their financial records through natural language queries by combining a graphical user interface with a strong database and fundamental NLP capabilities. By laying the foundation for upcoming improvements like data

visualization, advanced analytics, and more complex AI-driven features, this project creates a scalable and cutting-edge tool for managing personal finances.

1. REFERENCES AND RESOURCES
2. S. García-Méndez, F. de Arriba-Pérez, A. Barros-Vila, and F. J. González-Castaño, "Detection of Temporality at Discourse Level on Financial News by Combining Natural Language Processing and Machine Learning," *arXiv preprint arXiv:2404.01337*, 2024.
3. Nourbakhsh and G. Bang, "A Framework for Anomaly Detection Using Language Modeling, and Its Applications to Finance," *arXiv preprint arXiv:1908.09156*, 2019.
4. Q. Li, "Textual Data Mining for Financial Fraud Detection: A Deep Learning Approach," *arXiv preprint arXiv:2308.03800*, 2023.
5. T. Loughran and B. McDonald, "Textual Analysis in Accounting and Finance: A Survey," *Journal of Accounting Research*, vol. 54, no. 4, pp. 1187-1230, 2016.
6. Kumar and V. Ravi, "A Survey of the Applications of Text Mining in Financial Domain," *Knowledge-Based Systems*, vol. 114, pp. 128-147, 2016.
7. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, and D. C. P. Ngo, "Text Mining for Market Prediction: A Systematic Review," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7653-7670, 2014.
8. S. Kogan, D. Levin, B. R. Routledge, J. S. Sagi, and N. A. Smith, "Predicting Risk from Financial Reports with SVM," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2009, pp. 272-279.
9. P. Malo, J. Sinha, P. Korhonen, J. Wallenius, and P. Takala, "Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts," *Journal of the Association for Information Science and Technology*, vol. 65, no. 4, pp. 782-796, 2014.
10. E. Fisher, M. R. Garnsey, and M. E. Hughes, "Natural Language Processing in Accounting, Auditing and Finance: A Synthesis of the Literature with a Roadmap for Future Research," *Intelligent Systems in Accounting, Finance and Management*, vol. 23, no. 3, pp. 157-214, 2016.
11. D. Bholat, S. Hansen, P. Santos, and C. Schonhardt-Bailey, "Text Mining for Central Banks," *Centre for Central Banking Studies Handbook No. 33*, Bank of England, 2015.
12. M. El-Haj, P. Rayson, and P. Walker, "Financial Narrative Processing: Automating the Analysis of Financial Narratives," *The Journal of Finance and Data Science*, vol. 5, no. 1, pp. 1-17, 2019.
13. S. S. Groth and J. Muntermann, "An Intraday Market Risk Management Approach Based on Textual Analysis," *Decision Support Systems*, vol. 50, no. 4, pp. 680-691, 2011.
14. R. P. Schumaker and H. Chen, "Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFin Text System," *ACM Transactions on Information Systems*, vol. 27, no. 2, pp. 1-19, 2009.
15. W. Payne, J. R. Bettman, and E. J. Johnson, "The Adaptive Decision Maker," *Cambridge University Press*, 1993.
16. M. Hagenau, M. Liebmann, and D. Neumann, "Automated News Reading: Stock Price Prediction Based on Financial News Using Context-Capturing Features," *Decision Support Systems*, vol. 55, no. 3, pp. 685-697, 2013.