BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 1, December 2025 | Hal 434-444
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i1.776

Development of Chatbot Integration in Personal Finance Management Applications Using Generative Pre-Trained Transformer (GPT) 4o

Sawali Wahyu*, Tomflynn Beltsazar

Faculty of Computer Science, Department of Informatic Engineering, Esa Unggul University, Jakarta, Indonesia
Email: [1,*] sawaliwahyu@esaunggul.ac.id, [2] tomflynnbeltsazar@student.esaunggul.ac.id
Correspondence Author Email: sawaliwahyu@esaunggul.ac.id

Abstract−Financial management is considered a crucial aspect considering the importance of financial matters in life. However, based on the survey of 13 Employees and 2 Parallel Students of Esa Unggul University revealed that most don't regularly record their finances. The reasons cited includes time constraints and lack of understanding. Additionally, none of the respondents used a financial recording application due to perceived complexity and inefficiency. To overcome this, a chatbot-based personal finance management application was built. The chatbot feature in this application is supported by the GPT-4o model and implemented by performing rapid engineering, which aims to simplify and streamline financial records. The app will be developed using the Mobile-D method, which includes five stages: explore, initialize, productize, stabilize, and system test & fix. The purpose of this study is to integrate chatbot features to simplify and accelerate the financial recording process. To evaluate its effectiveness, the app will undergo System Usability Scale (SUS) and User Acceptance Testing (UAT). The SUS resulted in a score of 77.16, categorized as "Good," while the UAT showed positive user feedback. These results show that the integration of chatbots with financial applications has successfully helped users manage their finances more easily and efficiently. This study proves the potential of chatbot technology integration in personal finance management, offering user-friendly solutions to overcome common obstacles in financial record keeping.

Keywords: Chatbot; Finance Management; Generative Pre-Trained; Mobile Application; Mobile-D

1. INTRODUCTION

For some people, financial management is considered a crucial aspect given the importance of financial matters in today's life. In an era where consumptive lifestyles have become a common culture, especially in urban areas, managing finances has become increasingly sensitive and important. However, some research shows that there are still many people who do not realize how important financial management is, especially in the context of personal financial management [1].

Based on the survey of 13 Employees and 2 Parallel Students of Esa Unggul University revealed that most don't regularly record their finances. The reasons cited includes time constraints and lack of understanding. Additionally, none of the respondents used a financial recording application due to perceived complexity and inefficiency. To address this, a personal finance management app aimed at simplifying and streamlining financial recording is proposed. In addition to collecting data from respondents regarding financial record-keeping in their daily lives, further analysis was conducted through Several applications with similar functions already exist and are used by many people, such as Wallet, Money Manager, iMoney, Sprouts, and Ivy Wallet. These applications provide features and functions to help users manage their finances. However, none of these applications offer integration with chatbots to simplify the financial recording process.

In a study entitled Mobile Web-based Personal Finance Recording Application [1] states that Personal Finance Management Application made for mobile devices can help users record their income or expenses and view their monthly financial statements whenever and wherever they are. In a study titled Designing Android-Based Personal Finance Management Applications [2], a mobile application system was produced that will display financial data both income and expenses. The advantage of this system is that users can find out information on their financial condition so that it can be a reference in managing finances. In a study entitled Development of Money Tracker Application to Improve Financial Literacy and Personal Finance Management [3], The Money Tracker application improves financial literacy among its users by simplifying personal financial management and enabling users to analyze their spending and income patterns. This helps users make better financial plans and address various financial issues effectively. Based on previous research, the success of financial application development lies in its ease of use from the user's perspective. The purpose of this study is to integrate chatbots to facilitate the process of efficient financial recording. By using the chatbot feature, users can do various things such as recording, adding categories, creating budgets, and so on, simply by giving commands through the chatbot.

Ever since the emergence of ChatGPT in late 2022, there has been a huge surge in the integration of artificial intelligence in various applications. The use of AI as the basis for application building has significantly expanded the scope of artificial intelligence use [4]. From personal assistants capable of understanding natural language to analytics platforms that can provide deep insights from data, AI's ability to understand and respond to human content has opened the door for developers to think more creatively about how to incorporate AI into their products, resulting in innovations that benefit users in various aspects of users' lives [5].

Generative Pre-Trained Transformer (commonly known as ChatGPT) is a large language model developed by OpenAI. The GPT model uses a transformer architecture, which allows the model to process and understand the relationship between each word in a given text [6]. GPT is trained using pre-training and fine-tuning techniques. In the pre-training stage, the model is given text from various sources available on the internet, which allows the model to learn complex language structures and patterns. Fine-tuning is then performed using data sets that have been specifically labeled for specific tasks, such as translating languages, interacting in conversations or being a virtual assistant that can perform

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 1, December 2025 | Hal 434-444
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i1.776

various commands [7]. OpenAI provides an API for application developers that can be used to integrate the GPT model and perform fine-tuning to adapt the model to the needs of the application [8].

This research will be conducted on a mobile-based financial management application with chatbot integration to facilitate the financial recording process efficiently. Chatbot integration can help users manage their finance like recording a transaction, creating a category and budget, etc. just by sending messages to the AI.

2. RESEARCH METHODOLOGY

Figure 1 below illustrates a comprehensive framework that has been created to outline the necessary steps for achieving the research target. This framework provides a clear and structured approach, ensuring that each phase of the research is methodically addressed. By following this framework, the research process becomes more organized and goal oriented.

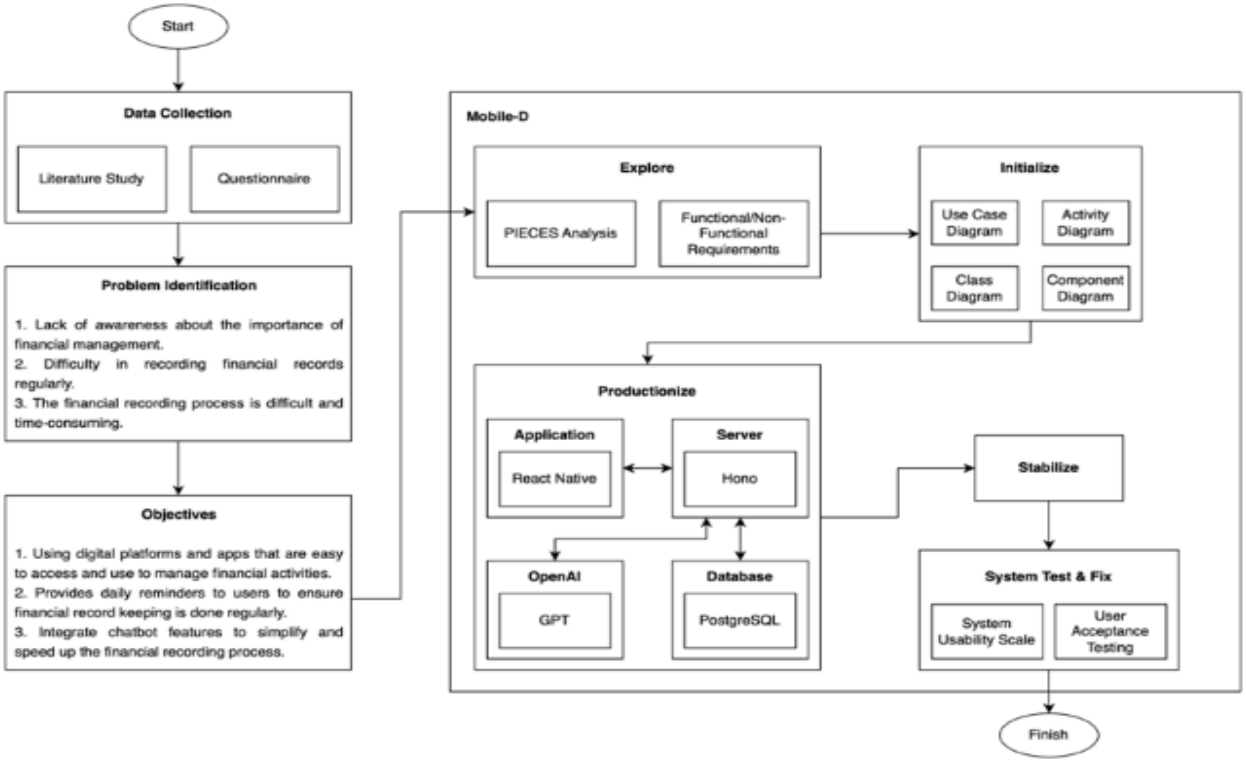

Figure 1. Research Method

2.1 Data Collection

The research begins with the data collection stage, which involves conducting a literature review of relevant journals published within the last five years. In addition to the literature study, questionnaires are distributed to gather primary data from 13 employees and 2 parallel students of Esa Unggul University. This approach ensures a comprehensive understanding of the topic by combining both existing academic insights and firsthand responses from the participants.

2.2 Development Method

After data collection, the application development stage is carried out. The development method used is the Mobile-D method. Mobile-D is an agile Software Development Life-Cycle (SDLC) method, making it suitable for small team sizes and short processing times. The advantage of using this development method is that it is specialized for creating mobile applications so that development from planning to maintenance can be fully focused on producing applications that are in accordance with the wishes of the user [9]. The following are the stages of the Mobile-D method, as illustrated in Figure 2 below:

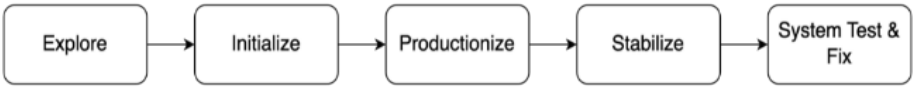

Figure 2. Mobile-D Method

Explanation of the stages of figure 2 :

a. Explore : Perform PIECES analysis to find out the advantages and disadvantages of the existing system and make functional and non-functional requirements for the application to be made.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 1, December 2025 | Hal 434-444
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i1.776

b. Initialize : Design software using the Unified Modeling Language (UML). The diagrams used are use case diagram, activity diagram, class diagram, component diagram, and deployment diagram.
c. Productionize : Implement the UML design and application architecture that has been created into the programming process to become an application. The application created is a mobile application that uses the React Native framework with the TypeScript programming language and chatbot API integration using GPT 4o.
d. Stabilize : Stabilize the application to ensure there are no bugs or features that do not function properly.
e. System Test and Fix : Testing the application that has been made using the System Usability Scale (SUS) and User Acceptance Testing (UAT) methods.

3. RESULT AND DISCUSSION

3.1 Explore

At this stage, the research focuses on problem identification to determine the specific needs that must be addressed in the application development process. This step is crucial as it helps to clarify the requirements and challenges that the application aims to solve. By thoroughly identifying these problems, the development team can ensure that the application is designed to meet the users' needs effectively and efficiently.

3.1.1 PIECES Analysis

This analysis evaluates key aspects such as Performance, Information, Economics, Control, Efficiency, and Services to identify areas where improvements can be made. By thoroughly examining these factors, the comparison highlights the strengths and weaknesses of both applications, guiding the development of a more efficient and effective financial management solution. **Kesalahan! Sumber referensi tidak ditemukan.** below provides a detailed PIECES analysis comparing the existing financial management application with the proposed new application.

Table 1. PIECES Analysis

Domain	Current Application	Proposed Application
Performance	Current finance management app has fairly good performance and responsive enough to have a good user experience.	Proposed finance management app should have a good performance and responsiveness to ensure a positive user experience.
Information	Current finance management app shows the user's financial information in an easy-to-understand format like table and graphs.	Proposed finance management app will show the user's financial information in an easy-to-understand format while also giving an option to ask the chatbot about the user's current finance information.
Economy	Many of the current finance management app limit basic features like category management, csv export, etc. behind a paywall, which can be an obstacle for many users on a tight budget.	Proposed finance management app will not limit basic features like category management, csv export, etc. behind a paywall.
Control	Current finance management app offers some personalization options like theme switcher, category management, and daily reminder notification.	Proposed finance management app will offer personalization options like theme switcher, category management, and a customizable daily reminder notification.
Efficiency	Current finance management app requires the user to input each transaction details manually.	Proposed finance management app will give the option for users to record their transactions via chatbot, where users only need to provide brief information, and the transactions will be saved automatically.
Service	Current finance management app has a cloud save service to prevent user's data loss.	Proposed finance management app will offer a cloud save service to prevent user's data loss and a chatbot service to help users manage their finance in the app.

Based on Table 1, a method of analysis used to identify and evaluate problems in a system (both existing and planned systems).PIECES helps to view the system from various aspects so that improvements can be formulated in a more focused manner.

3.1.2 Functional Requirement

These requirements are essential for defining the core functionalities that the application must provide to meet the needs of its users. By clearly specifying these functional requirements, the development process is guided to ensure that the final application delivers the expected features and performance. **Kesalahan! Sumber referensi tidak ditemukan.** below outlines the functional requirements for the financial management application that will be created.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 1, December 2025 | Hal 434-444
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i1.776

Table 2. Functional Requirements

No.	Requirement	Function
1	Authentication	User can register an account.
		User can log in their registered account using valid credentials.
2	Manage Income and Expense	User can record income transaction.
		User can record expense transaction.
3	Manage Budget	User can create a new budget.
		User can edit a budget.
		User can set which categories are included in the budget calculation
4	Manage Category	User can create a new category.
		User can edit a category.
		User can delete a category.
5	Manage Account	User can create a new account.
		User can edit an account.
		User can delete an account.
		User can modify account balance.
6	Chatbot	User can interact with chatbot to record a transaction.
		User can interact with chatbot to create a new category.
		User can interact with chatbot to create a new budget.
		User can interact with chatbot to ask about user's current financial information.
7	View Financial Report	User can view monthly financial report.
8	Export CSV	User can export financial data into csv file format.
9	Daily Reminder Notification	User can enable daily reminder notification.
10	Cloud Save	User data is saved in the cloud to prevent data loss.

Based on **Kesalahan! Sumber referensi tidak ditemukan.**, explaining about Functional requirements describe what the system must do. They are directly related to the features, processes, and workflows within the system.

3.1.3 Non-functional Requirements

These requirements focus on the system's overall performance, scalability, and ease of use, ensuring that the application operates smoothly under various conditions. By detailing these non-functional requirements, the development process can address critical aspects such as security, scalability, and usability, which are vital for the application's success and user satisfaction. **Kesalahan! Sumber referensi tidak ditemukan.** below outlines the non-functional requirements for the financial management application that will be created.

Table 3. Non-Functional Requirements

No.	Requirement	Function
1	Performance	Application should be performant and capable of handling lots of requests simultaneously.
2	Scalability	Application should easily scale to handle large number of users and data.
3	Ease of Use	User interface should be intuitive and easy to use, even for users who are not very skilled in terms of technology

Based on **Kesalahan! Sumber referensi tidak ditemukan.**, explaining about Non-functional requirements describe how the system works, rather than its main functions. This section describes the performance, scalability, and ease of use of the financial record-keeping mobile application.

3.2 Initialize

In this stage, system design begins by using the identified problems and defined application requirements as a foundation. This process creates a comprehensive blueprint that outlines the structure and functionality of the system to be developed. By doing so, it provides a clear overview of how the final system will operate, ensuring that all necessary components are considered and integrated effectively.

3.2.1 Use Case Diagram

Use case diagrams are visual representations for the expected behavior of the software being developed. Each case describes the system behavior required by actors to achieve their goals [10]. In the use case diagram below, the User is the main actor who interacts with the system. Figure 3 below shows the use case diagram of chatbot-based financial management application :

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 1, December 2025 | Hal 434-444
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i1.776

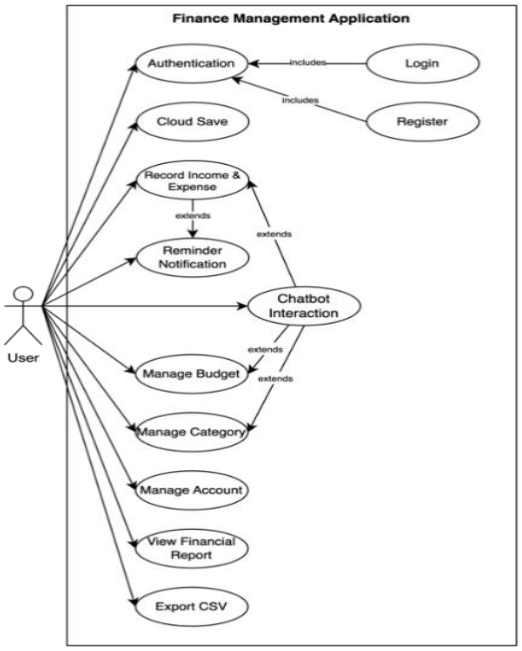

Figure 3. Use Case Diagram

Based on Figure 3. The following are the details of the functional requirements described:

a. Authentication: User verification process, users can login to an existing account, or create a new account by registering.
b. Cloud Save: Storing user data in the cloud to prevent data loss.
c. Manage Income & Expenses: Record, modify, and delete income and expenses.
d. Reminder Notification: Provides daily reminder notifications to users to record transactions.
e. Manage Budget: Create, modify, and delete budgets.
f. Manage Categories: Create, modify, and delete transaction categories.
g. Manage Accounts: Create, modify, and delete.
h. View Financial Report: View details of expenses and income.
i. Export CSV: Export expense and income data to a csv file.
j. Chatbot Interaction: Record transactions, manage budgets, manage categories, through chatbot interaction.

3.2.2 Activity Diagram

Activity diagrams are visual representations of the processes that occur in a system. Activity diagrams do not highlight the behavior of actors and describe the workflow or activity of the system [11]. Activity diagrams are used to illustrate the workflow or sequence of activities of a system. These diagrams are commonly used in software design and analysis to help in understanding how tasks are performed in a process [12]. Figure 4 below shows the activity diagram of a chatbot-based finance management application.

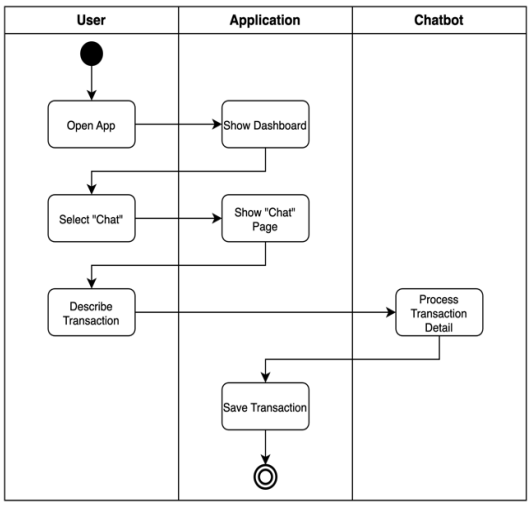

Figure 4. Activity Diagram

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 1, December 2025 | Hal 434-444
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i1.776

In the activity diagram shown in Figure 5, Explaining the workflow of a mobile financial record-keeping application integrated with a chatbot, to make it easier for users to use the application.

3.2.3 Class Diagram

Class diagrams are used to describe the object structure of the system. This diagram shows the object classes that make up the system and the relationships between these object classes. Class diagrams also show the properties and methods of a class, and the constraints contained in these relationships [13]. Class diagrams are static, which means they don't explain how classes behave during interactions, but rather how the relationship between each class looks like [11]. The following is outlined in figure 5 below :

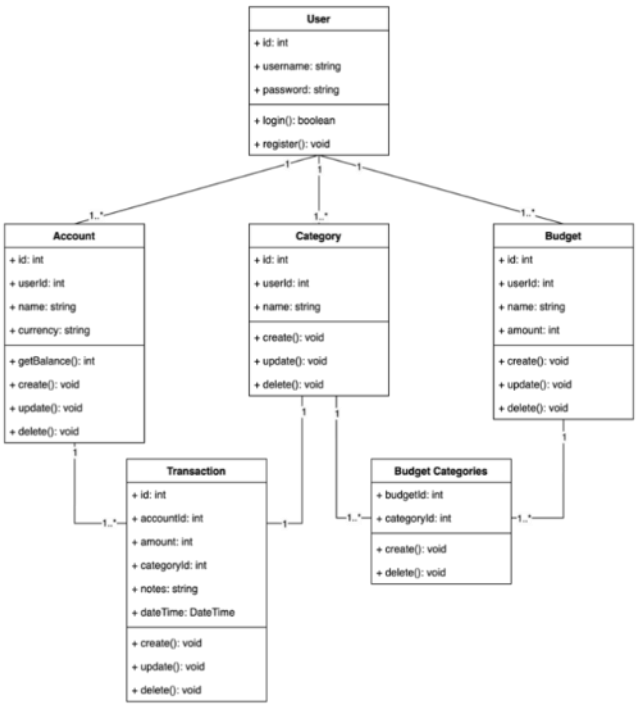

Figure 6. Class Diagram

In the class diagram shown in Figure 6, there is a User class that represents application users, each User can have many Accounts, Budgets, and Categories. Account represents the User's account, and each Account can have many Transactions. The amount of account balance is not stored directly in the Account class, but uses the `getBalance()` method to add up every transaction ever made by the User to get the final balance value. Budget represents the budget created by the user; the user can assign more than 1 Category to the Budget to limit the calculation of transactions to a certain category.

3.2.4 Component Diagram

Component diagrams are used to illustrate the structure of relationships and conditions among various software components [14]. Component diagram is a visual representation that connects each component in the system to provide an overview of how the system works visually [15]. The following is outlined in figure 6 below:

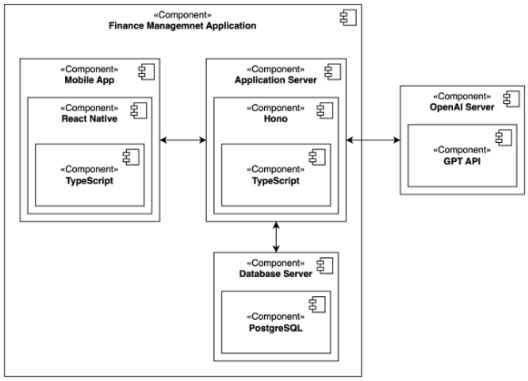

Figure 7. Component Diagram

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 1, December 2025 | Hal 434-444
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i1.776

The personal finance management application consists of three main components as shown in Figure 7, namely the Mobile App made using React Native and TypeScript is a component that will interact directly with the user, the Application Server made using Hono and TypeScript is a component that will process requests from the Mobile App by interacting with the OpenAI Server and Database Server to provide an appropriate response.

3.2.5 Application Architecture

Application architecture is an important foundation in software development. Application architecture defines the overall structure and design of a system, it includes identifying the main components of the application, how they are interconnected, and the principles that govern their design and evolution. Choosing the right architecture is critical as it will affect various aspects of the application, such as performance, scalability, and maintainability [16]. The following is outlined in figure 7 below:

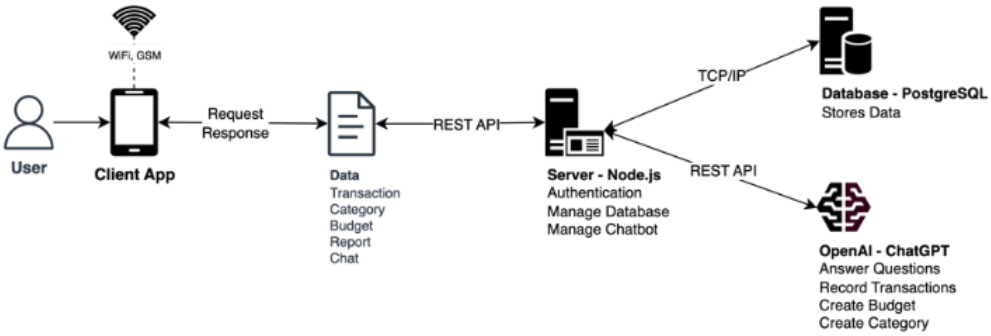

Figure 8. Application Architecture

The financial management application, as illustrated in **Kesalahan! Sumber referensi tidak ditemukan.**, follows a client-server architecture. Client-server architecture is a system design model that divides tasks and responsibilities between the service provider (server) and the service requestor (client). The server is responsible for authentication, database management, and chatbot. The data managed includes transactions, categories, budgets, and chats, stored in a PostgreSQL database. REST API is used for communication between client and server, as well as for integration with OpenAI ChatGPT for chatbot functions. The chatbot functions include answering questions, recording transactions, creating budgets, and creating categories. This architecture allows the application to manage data, perform transactions, and provide AI-powered chat features, while maintaining security and efficiency in data management and processes.

3.3 Productionize

At this stage, the system design is translated into code, marking the beginning of the implementation phase. The design blueprint is meticulously followed, gradually transforming the conceptual design into a functional application. This process continues until the application is fully developed, ready for testing and deployment.

3.3.1 Folder Structure

The client folder, as shown in

Figure **9** above, contains the application's source code that runs on user devices for both Android and iOS platforms.

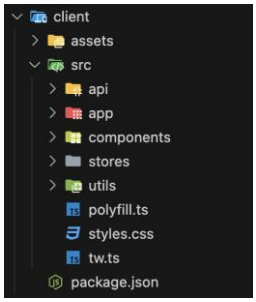

Figure 9. Folder Structure

Below is an explanation of Figure 8. About client folder structure:

a. Assets folder : This folder stores various static resources such as images that will be used in the application.
b. Api folder : This folder stores API function calls that are used to communicate with the server.
c. App folder : This folder contains the views and routes of each page displayed in the application.
d. Components folder : This folder contains reusable UI components that are used in various places of the application.
e. Stores folder : This folder contains the global state that exists in the application.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 1, December 2025 | Hal 434-444
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i1.776

f. The utils folder : This folder contains functions to help solve common small problems.

3.3.2 Chatbot Implementation

The chatbot feature in the app is powered by the GPT-4o model and implemented by performing prompt engineering. First, the chatbots are given instructions that explain their tasks as well as some basic information about their users, as shown in Figure 10 below:

```
{
  role: "system",
  content: `You are an assistant to help users by answering their questions and record their transactions for them.
            User ID is ${userId}.
            Account ID is ${accountId}.
            Current date is ${new Date().toJSON()}.
            These are the available user-defined transaction categories: ${JSON.stringify(userCategories)}`,
},
```

Figure 10. Chatbot Instruction

Explain Figure 9. the chatbot is given several tools so that it can perform actions other than answering questions. Tools are tools that can be used by LLM to perform certain tasks.

3.3.3 User Interface Applications

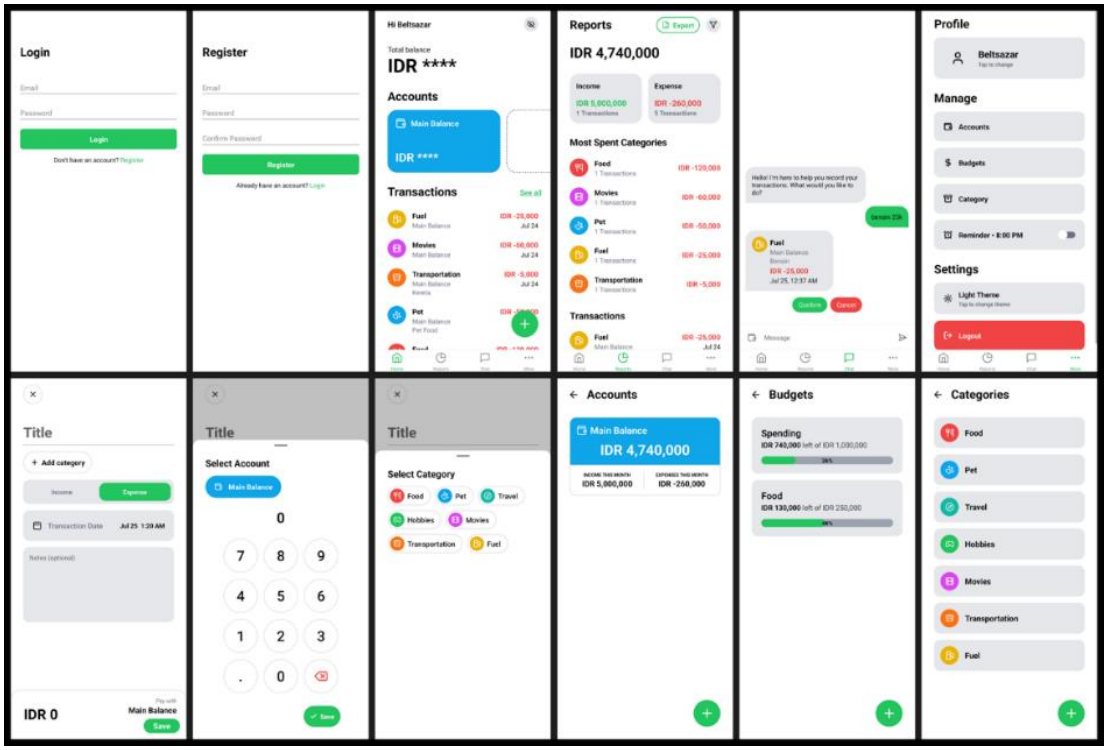

Figure 11. Application UI

Figure 11 above shows the results of the implementation of mobile applications created using the React Native framework with the TypeScript programming language. The following describes the user interface of the finance management application :

a. Login & Register : On the login page, users can log in to their account by entering the right combination of email and password, and if users do not have an account, they can create one on the register page by entering an email and password.
b. Home : The home page is the page that is displayed first after the user logs into the application. On this page there is information such as total balance, account list, and transaction history. Users can add new transactions by pressing the + button in the bottom right corner of the screen.
c. Add Transaction : On this page users can add transactions by entering the transaction title, transaction type (income or expense), transaction date, transaction amount, and transaction category.
d. Report : On this page, users can view their financial reports every month. There are various information such as total expenses and income, a list of the largest category expenses, and all transactions in the selected month. On this page, users can also export their financial data in CSV form.
e. More : On this page users can change their name and profile picture, manage accounts, manage categories, manage budgets, set daily reminders, change themes, and log out.
f. Manage Account : On this page users can create, modify, and delete accounts that have been created. Each account requires a name, icon, color, and balance amount.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 1, December 2025 | Hal 434-444
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i1.776

g. Manage Budget : On this page users can create, modify, and delete budgets that have been created. Each budget requires a name and the amount of the budget created. Users can also add a list of categories for the budget, which aims to limit budget calculations to certain categories only.
h. Manage Category : On this page users can create, change, and delete transaction categories that have been created. Each category requires a name, icon, and color.

3.4 Stabilize

In this stage, stabilization is carried out by unit testing the application that has been made. Unit tests test each function or method in the application to ensure there are no bugs or features that do not work as they should. Figure 12 below shows the results of unit testing.

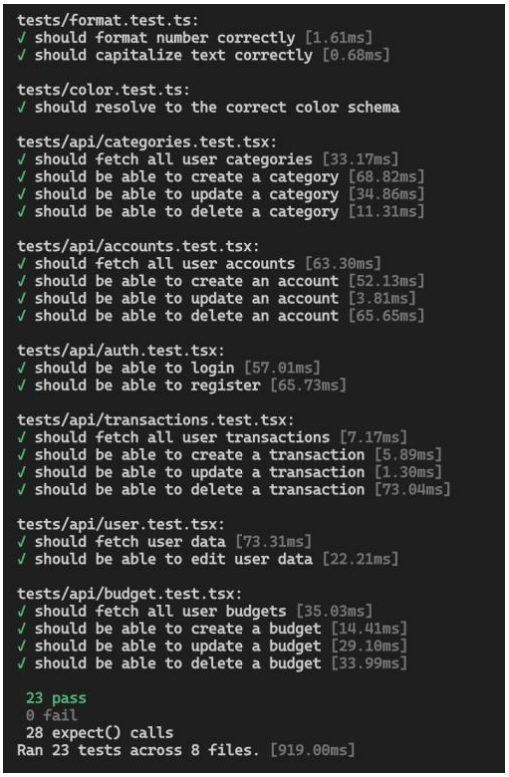

Figure 12. Unit Test Result

Based on Figure 11, Unit test results, which are reports or outputs that show whether each unit (small part of the code/program) in the system is functioning correctly. The image shows unit test results that have no bugs, and the program runs well.

3.5 System Test & Fix

In this stage, testing is conducted on the developed applications using System Usability Scale (SUS) and User Acceptance Testing (UAT). The SUS method evaluates the application's overall usability from the user's perspective, providing insights into how intuitive and effective the application is. Concurrently, UAT is performed to ensure that the application meets the end-users' requirements and expectations, confirming that it functions as intended in real-world scenarios.

3.5.1 System Usabillity Scale (SUS)

System Usability Scale is a Usability testing system developed by John Brooke in 1986. SUS was developed to measure the usability of a system or application. SUS consists of 10 simple statements to understand the problems experienced by users when using a system or application [17]. SUS uses a simple Likert scale, where respondents are asked to give their level of agreement within a 5-point scale range. The SUS testing method is reliable and economical for testing the usability of systems globally [18]. **Kesalahan! Sumber referensi tidak ditemukan.** below shows the results of the System Usability Scale survey conducted on 13 Employees and 2 Parallel Students at Esa Unggul University.

Table 4. System Usability Scale Result

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Total	Total x 2.5
1	1	5	5	5	1	5	5	1	5	20	50
5	1	1	1	1	1	1	1	1	1	24	60
5	1	5	5	5	5	5	1	5	1	32	80
5	1	5	2	5	1	5	2	5	2	37	92.5

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Total	Total x 2.5
4	3	4	3	4	2	4	2	4	3	27	67.5
5	4	5	3	4	4	4	1	5	4	27	67.5
3	2	3	3	2	3	3	2	2	2	21	52.5
5	1	5	1	5	1	5	1	5	1	40	100
2	2	5	1	5	2	5	1	5	1	35	87.5
5	1	5	1	5	1	5	1	5	2	39	97.5
5	1	5	5	5	1	5	1	5	5	32	80
3	1	5	1	5	1	5	1	3	3	34	85
5	1	5	5	5	1	5	1	5	3	34	85
3	3	5	3	3	1	5	1	5	3	30	75
4	2	5	2	4	2	5	1	4	4	31	77.5
										Average	77.16

Based on table 4. the results of System Usability Scale testing, the chatbot-based personal finance management application produces an average score of 77.16. This value indicates that the usability of the application created is included in the Good category, which means that the application can be well received by users in terms of features and usability.

3.5.2 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is a test of the interaction between users and the system directly which aims to ensure that features are running according to user needs [19]. This process is different from system testing such as making sure the system runs well and according to specifications, but rather ensuring that the solution in the system will work for users by giving a direct survey to users [20]. Based on the results of UAT testing, it can be concluded that chatbot-based personal finance management applications can help users in the financial management process. The advantages of this application include user convenience, has an attractive appearance, and has a positive impact on users. Respondents also provided several criticisms and suggestions for the development of the application, such as notification features if they exceed certain spending limits, a feature to scan images of shopping receipts, and synchronization with banks.

4. CONCLUSION

This research successfully Development of Chatbot Integration with mobile-d as the lifecycle of mobile application development, and the System Usabillity Scale resulted in a score of 77.16, categorized as "Good," while the UAT showed positive user feedback. These results suggest that the chatbot-based finance app successfully helps users manage their finances more easily and efficiently. The integration of the GPT-4o API, grounded in the large language model (LLM) paradigm, demonstrates a significant advancement in the development of chatbot-based personal financial management mobile applications. By leveraging the natural language understanding and generative capabilities of GPT-4o, the system is able to provide context-aware financial guidance, personalized recommendations, and interactive user support. This integration not only enhances user experience and engagement but also contributes to more effective financial decision-making processes. Therefore, the application of GPT-4o within this research validates the feasibility and practicality of employing generative AI models for improving the accessibility, efficiency, and reliability of personal financial management tools. This research can be further extended by enhancing the chatbot feature through the integration of speech recognition technology, enabling the system to process and understand voice inputs. Such an improvement would allow the AI to handle more complex interactions by combining natural language processing with voice-based communication, thereby increasing accessibility, improving user experience, and broadening the applicability of the chatbot in personal financial management.

REFERENCES

[1] A. Rosidi and A. Afriyudi, "Aplikasi Pencatatan Keuangan Pribadi Berbasis Web Mobile," *J. Teknol. Inform. dan Komput.*, vol. 9, no. 1, pp. 100–113, 2023. https://doi.org/10.37012/jtik.v9i1.1447.

[2] E. Trivaika and M. A. Senubekti, "Perancangan aplikasi pengelola keuangan pribadi berbasis android," *Nuansa Inform.*, vol. 16, no. 1, pp. 33–40, 2022. https://doi.org/10.25134/nuansa.v16i1.4670.

[3] A. D. Saputra, A. H. Oktavia, D. P. Putra, R. Wijaya, and E. Hikmawati, "Pembangunan Aplikasi Money Tracker untuk Meningkatkan Literasi Keuangan dan Pengelolaan Keuangan Pribadi," *J. Ilm. FIFO*, vol. 15, no. 2, p. 186, 2024, doi: 10.22441/fifo.2023.v15i2.010.

[4] A. Setiawan and U. K. Luthfiyani, "Penggunaan ChatGPT untuk pendidikan di era Education 4.0: Usulan inovasi meningkatkan keterampilan menulis," *J. PETISI (Pendidikan Teknol. Informasi)*, vol. 4, no. 1, pp. 49–58, 2023. Retrieved from https://e-journal.unimudasorong.ac.id/index.php/jurnalpetisi/article/view/784.

[5] M. Ibrahim, J. Nasir, A. Komarudin, A. Maulana, and M. H. Akbar, "Integrasi Kecerdasan Buatan dalam Desain Aplikasi Seluler: Peningkatkan Pengalaman Pengguna di Era Ekonomi Digital," *Nusant. Comput. Des. Rev.*, vol. 1, no. 1, pp. 31–39, 2023. https://doi.org/10.55732/ncdr.v1i1.1091.

[6] F. Sufi, "Generative pre-trained transformer (GPT) in research: A systematic review on data augmentation," *Information*, vol. 15, no. 2, p. 99, 2024. https://doi.org/10.3390/info15020099.

[7] D. Setiawan, E. A. D. Karuniawati, and S. I. Janty, "Peran Chat Gpt (Generative Pre-Training Transformer) Dalam Implementasi

Ditinjau Dari Dataset," *Innov. J. Soc. Sci. Res.*, vol. 3, no. 3, pp. 9527–9539, 2023. Retrieved from https://j-innovative.org/index.php/Innovative/article/view/3286.

[8] A. R. Aryabimo, D. Bernady, N. N. K. Sari, and V. H. Pranatawijaya, "Implementasi Api Chat Gpt Pada Aplikasi Restoran Berbasis Website," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024. https://doi.org/10.23960/jitet.v12i3.4408.

[9] N. A. O. Saputri, E. Amorita, and P. H. Saksono, "E-Lelang Barang Antik Berbasis Mobile Pada Komunitas Pecinta Antik Kreatif Sriwijaya Palembang Menggunakan Metode Mobile-D," *J. Pengemb. Sist. Inf. dan Inform.*, vol. 1, no. 2, pp. 128–137, 2020. Retrieved from https://pdfs.semanticscholar.org/fbd3/825837b4e17d719ffedc8ec7fdf80ed1941f.pdf.

[10] R. Rohmanto and T. Setiawan, "Perbandingan Efektivitas Sistem Pembelajaran Luring dan Daring Menggunakan Metode Use case dan Sequence Diagram," *Intern. (Information Syst. Journal)*, vol. 5, no. 1, pp. 53–62, 2022. https://doi.org/10.32627/internal.v5i1.506.

[11] N. N. Syarif and A. Voutama, "Sistem Informasi Pengelolaan Rumah Kos Berbasis Website Menggunakan Penerapan UML," *Informal Informatics J.*, vol. 9, no. 1, pp. 1–11, 2024. Retrieved from https://pdfs.semanticscholar.org/7c17/b37ed8f1e00a718386f4fff250ec1cf46ebd.pdf.

[12] S. Al-Fedaghi, "Validation: Conceptual versus activity diagram approaches," *arXiv Prepr. arXiv2106.16160*, 2021. https://doi.org/10.48550/arXiv.2106.16160.

[13] F. Chen, L. Zhang, X. Lian, and N. Niu, "Automatically recognizing the semantic elements from UML class diagram images," *J. Syst. Softw.*, vol. 193, p. 111431, 2022. https://doi.org/10.1016/j.jss.2022.111431.

[14] A. Abdilah, B. Lailiah, and R. Sa'adah, "Rancang Bangun Sistem Informasi Rent Car Dengan Pengunaan Aplikasi Online," *Comput. Sci.*, vol. 1, no. 2, pp. 105–112, 2021. https://doi.org/10.31294/coscience.v1i2.474.

[15] B. Wildan, A. P. Sari, and R. Nasution, "Sistem Informasi Manejemen Surat Berbasis Web Pada Pt. Clipan Finance Indonesia, Tbk," *Hexagon*, vol. 2, no. 1, pp. 85–90, 2021. https://doi.org/10.36761/hexagon.v2i1.882.

[16] N. Siagian, T. E. Tamba, H. H. O. Situmorang, and H. Samosir, "Aplikasi Apotek Berbasis Web Menggunakan Arsitektur Microservices (Studi Kasus Apotek Glen, Kab. Toba)," *J. Appl. Technol. Informatics Indones.*, vol. 1, no. 2, pp. 22–28, 2021. Retrieved from : https://pdfs.semanticscholar.org/e7b9/c111afacc9a39940bda86d451d64aca5bc4a.pdf

[17] A. Kaya, R. Ozturk, and C. Altin Gumussoy, "Usability measurement of mobile applications with system usability scale (SUS)," in *Industrial Engineering in the Big Data Era: Selected Papers from the Global Joint Conference on Industrial Engineering and Its Application Areas, GJCIE 2018, June 21–22, 2018, Nevsehir, Turkey*, Springer, 2019, pp. 389–400. https://doi.org/10.1007/978-3-030-03317-0_32

[18] I. Salamah, "Evaluasi usability website polsri dengan menggunakan system usability scale," *J. Nas. Pendidik. Tek. Inform. JANAPATI*, vol. 8, no. 3, pp. 176–183, 2019. https://doi.org/10.23887/janapati.v8i3.17311.

[19] M. A. Chamida, A. Susanto, and A. Latubessy, "Analisa User Acceptance Testing Terhadap Sistem Informasi Pengelolaan Bedah Rumah Di Dinas Perumahan Rakyat Dan Kawasan Permukiman Kabupaten Jepara," *Indonesia. J. Technol. Informatics Sci.*, vol. 3, no. 1, pp. 36–41, 2021. doi: 10.24176/ijtis.v3i1.7531.

[20] E. Suprapto, "User Acceptance Testing (UAT) Refreshment PBX Outlet Site BNI Kanwil Padang," *J. Civronlit Unbari*, vol. 6, no. 2, pp. 54–58, 2021. http://dx.doi.org/10.33087/civronlit.v6i2.85.