

Design And Implementation of a Scalable Real-Time Expense-Tracking Backend System

Anjali Bobade

Department of Science and Technology G.H. Raisoni Skill Tech University, Nagpur, India

Abstract

In today's world, digital payments and online financial activities have become a central part of everyday life. As a result, many people now depend on digital tools to keep track of how much they spend. But often, most expense trackers focus on how they look or how users enter data, and their backends simply can't keep up. In this study, we share a backend system built to grow with users over time. We break it down into layers like a user dashboard, business logic, alert engine, and a secure data layer so that no matter how many users come on board, the system stays quick and dependable. One of the most important features is that real-time budget alerts, which immediately let users know when they are about to exceed their planned spending.

We didn't just stop at being scalable; we also put security first. By using strong authentication and encrypted storage, we make sure everyone's financial data stays safe. After we ran tests with hundreds of users simultaneously, we noticed a huge improvement in both how quickly and how steadily the system performed, especially when compared to the old approach. As a result, we developed a simple, dependable tool that truly helps people make sense of their spending and gives them the reassurance to make better financial choices every day.

Keywords: Expense Tracking, Backend System, Real-Time Processing, REST API, Alerts, Layered Design, Scalability



This is an open-access article under the [CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/) license

1.Introduction

These days it feels normal to spend money without thinking about it. We use our phones to make every small payment and after a while it becomes hard to remember how many times we actually spent money during the day or week. When we are doing our daily things we spend money little by little without paying much attention. We might pay for travel buy something from a store or make a quick online payment and then move on without thinking about it again. Since these payments happen at times they do not stay in our memory for long. By the time a few weeks pass it becomes difficult to connect all those expenses together or understand how they formed a large amount. Later when we check our account balance we often feel confused because we cannot clearly remember where all the money was used. This kind of situation is becoming very common among students and often make users think again before spending money. Even a short message or alert can make a person check their remaining balance. Decide more carefully how to use it during the rest of the month.

Many people form their opinion about an application based on the experience of using it. While these things are important there is another part of the system that plays an important role but usually remains hidden from the user. The main work of the expense tracking application happens behind the screen,

where all the information is handled. When an expense is entered the system has to save it so that the same information is available later without any problem. When users open the expense tracking application again to look at what they entered the information should be shown in a simple and readable way. The expense tracking application should not make the user wait or search much to find their own records. If the system fails to manage the information people may start doubting whether their entries are really safe. For this reason the background process of handling data has to remain consistent and dependable.

For an expense tracking application to continue working the backend has to be arranged in a clear and practical way. Each action performed by the user should be handled without delay. The saved records should stay intact so that users do not face problems finding their entries later. When a user edits or corrects an expense the new information should show up correctly. This way the old entry and the new one will. There will be no confusion. The system should work properly when a lot of people are using it at the same time. If the expense tracking application is slow or does not work as it should users will start to doubt it. The expense tracking application should always work well so users can trust the expense tracking application. Keep using the expense tracking young users who make frequent digital payments.

One thing I noticed is that people usually think they can remember their expenses without writing them down but in practice this does not work well. After some time it becomes difficult to recall which expense happened on which day. Even if someone tries to remember small details are usually forgotten. Some expenses are regular while others happen once in a while and without proper records everything gets mixed up. A few people try to keep notes or write expenses on paper. This habit usually does not continue for long.. The notes get lost or people stop writing regularly. Because of this it becomes necessary to have a method where expenses can be saved safely and checked later whenever required. When expenses are kept in a record it becomes much easier to understand how money is being used. Seeing the expenses written in one place gives an idea than just trying to remember them. It becomes possible to compare spending from one week to another or from one month to another. Over a period of time the records begin to show details that are usually missed in daily life. We may notice repeated spending that did not seem important earlier. Sometimes the pattern only becomes visible after looking at the list of expenses. This kind of view makes it easier to think about spending and plan in a more sensible way. Of depending on rough guesses we can look at the records and decide what changes might be needed.

Expense tracking applications provide a solution to this problem. These applications allow users to enter their expenses regularly and store the information in one place. Users are able to look at their past expenses and understand how their money has been used over time. The information can be arranged in groups so that it becomes easier to read and understand. This makes it simple to notice where most of the money is going. Such small reminders application without any problems. A arranged backend helps keep the system consistent and dependable. In this research work the main focus is on designing and developing a backend system for an expense tracking application. The system is planned in such a way that different parts handle tasks. The backend is arranged, so that requests can be handled in a sequence and information can be processed without confusion. The expense information is stored in a manner so that users can look it up later without difficulty. Maintaining the records in this form also helps prevent mistakes and supports updates or improvements in the future.

The backend in this project is planned in a way that supports day-to-day use without creating complications for users. Whenever users add an expense the entry should be stored immediately and should be visible when they check their records again. The system also checks spending values from time, to time so that users can stay aware of their limits. Notifications are provided in a way so that users understand their spending situation without difficulty. These features help the expense tracking application remain useful in daily situations and not just in testing conditions. A dependable system

can help users trust the expense tracking application and use it regularly as part of their financial management.

2. Technologies Used

The expense tracking system was made with HTML and CSS and JavaScript to create the part that users see. The backend was built with Java and the Spring Framework to do what the expense tracking system is supposed to do and to work with the data. The expense tracking system uses a SQL database to keep the user information and the expense records nice and organized. The expense tracking system uses REST APIs and authentication tokens to make sure the different parts of the expense tracking system talk to each other in a way. The expense tracking system was built to be secure. The expense tracking system uses these technologies to do that.

3. Related Work

Lots of people have made apps to help others keep track of what they spend. These apps usually let users write down what they spend each day and look at what they have been spending. Some systems even give users reports so they can see how their money is being spent over time. Some researchers have tried to make expense management systems by adding things like automatic categorization and guessing what people will spend. Some new apps store data on cloud servers so users can get to their records from any device. Even though cloud storage makes it easy to get to data the inside of the system is still important for it to work smoothly.

Many apps are made with all the parts connected together which's easy to do at first but gets hard to fix and update when the system gets bigger. That is why some researchers think it is better to use microservices architecture, where each part works on its own. Not every expense tracking system needs microservices architecture. If the system is designed well with layers it can still work well. Be easy to maintain. In systems the people who made them do not explain how the backend works very clearly. They also do not say much about getting notifications in time and testing how well the system performs.

This project is, about making a backend system that is structured and supports storing data in a way and sending expense alerts in real time while making sure the system works well. The expense tracking system is the focus of this project. The goal is to make the expense tracking system work smoothly.

4. Research Gap

I looked into existing expense management apps and research studies. I found some problems with them. Most apps let you record expenses and make reports.. They don't explain how they work inside. You can't see how they handle users and lots of data.

Some apps send notifications in time.. You don't know how they do it. Some apps don't warn you away when you go over your spending limit. That doesn't help. Apps also don't tell you much about how they test performance. You don't know how fast they respond or how users they can handle. Security is another problem. Some apps don't say how they protect your data.

Because of these problems we need a backend system. It should be simple organized, secure and work well for users. This project tries to fix these problems by designing a backend system for expense tracking.

5. Research Methodology

This research is, about designing a backend system for an expense tracking app. I used an approach to build the system. Each part does one thing. I designed, built and tested the system. It stores expense data safely. Handles user requests fast. The system lets users record expenses, store data and get alerts when they go over their limit. I tested it with users.

Here is how I did it:

I looked at what the system needs to do.

The system needs to do things and I had to figure out what those things are.

I made a plan for the backend.

The backend is very important. I had to make sure it is good.

I used Java and Spring Framework to build the system.

Java and Spring Framework are useful tools for building the system.

I added a database to the system.

This system now has a database where all the information, for the system is stored.

I made a system that actually works for the system.

6. Problem Statement

In our life many students and working people buy things like food, travel tickets and mobile recharge and they also have other routine needs. These things do not cost a lot of money at the time so people usually do not write down how much they spend. After some time it becomes hard to remember where the money went and how it added up to such an amount. Many people try to keep track of their money by writing it down on paper or just remembering it. This works for a while but it is not a good way to do it for a long time. If you do not have a system you might forget what you spent your money on and you will not really know how you spend your money.

An expense tracking application needs to be able to store what you spend your money on and let you look at it whenever you want. The system needs to work even when a lot of people are using it at the same time. If the application is slow or does not store your money information correctly you will not trust it.

Another important thing is that your financial information needs to be safe and organized. You should be able to update what you spend your money on whenever you need to. The new information should be easy to see without being confusing.

Because of these things we need a backend system that can store expense information safely handle what users ask for quickly and work well all the time. This project is about making such a backend system for an expense tracking application.

7. Proposed System

The system we are proposing is an expense tracking backend that helps users store and manage their expenses in a simple way. The expense tracking application allows users to add expenses look at old records and check their spending limits. The system is designed so that expense records are stored safely and you can look at them whenever you need to. You can make changes to your records. The new information will be easy to see so you do not get confused between old and new entries.

The backend system handles what users ask for step by step and stores the information in a database. This makes the system easier to manage and reduces mistakes. To keep the system organized we used an architecture. This means different parts of the system do things. One part handles how users interact with the system another part handles what users ask for another part checks spending limits. Another part stores the information.

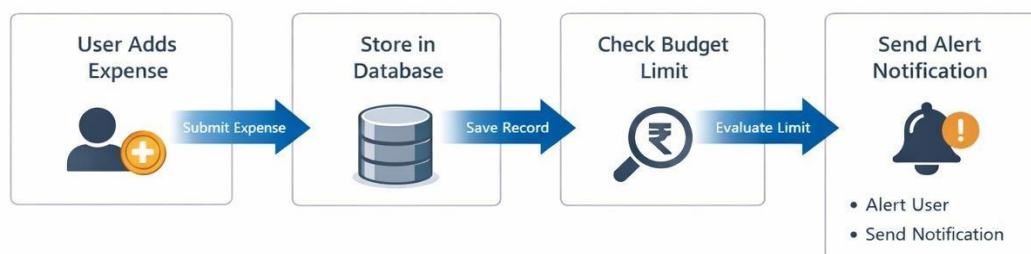


Figure 2: Expense Processing and Alert Flow

This diagram illustrates how expense data moves through the system, from user input to database storage and alert generation.

Fig 1:Expense processing and alert flow

Figure 1 illustrates the flow of expense data in the backend system. When a user adds an expense, the data is stored in the database and the system immediately evaluates the predefined budget limit. If the spending exceeds the limit, an alert notification is generated and sent to the user in real time.

This makes the system really easy to figure out and take care of. The system is simple so we can make changes on without having to change the whole system. We can just make changes, to the system. It will still work fine. The system will still be easy to understand.

The system also has a feature that checks expenses after they are stored. If you spend more, than your limit the system sends you a notification so you can stay aware of your spending.

The proposed system is designed to work with multiple users and keep working smoothly. The goal is to provide a backend structure that can be used in a real expense tracking application, like the expense tracking application we are talking about

8.Layer Architecture

The system has four layers.

- I. Presentation
- II. Application
- III. Alert Engine
- IV. Data Layer

Each one does a job.

- I. The Presentation layer handles user interaction.
- II. The Application layer processes requests.

- III. The Alert Engine layer generates alerts.
- IV. The Data Layer stores data.

This structure makes the system simple and easy to manage

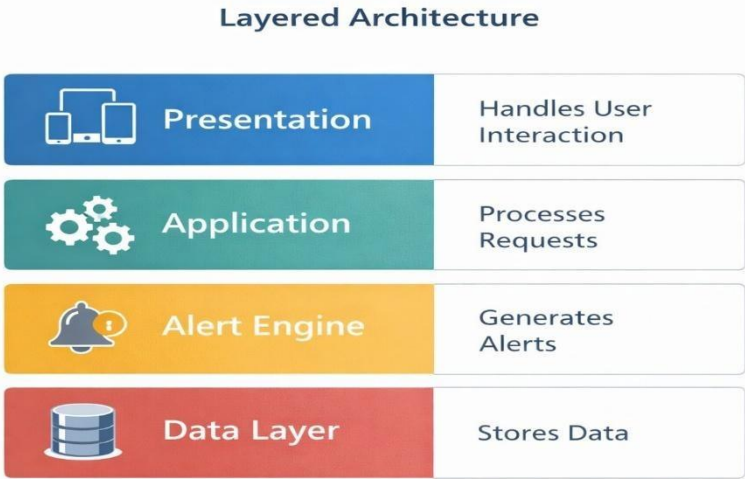


Fig 2: Layered Architecture

Figure 2 shows the four-layer architecture of the proposed backend system. The Presentation layer is where the backend system manages how users interact with it. The Application layer is where the backend system does all the work, for the business. The Alert Engine is what keeps an eye on how much money's being spent and sends out warnings when it needs to. The Data Layer is where the backend system keeps all the records of what people have spent. This way of organizing the backend system makes it easier to understand and work with. The backend system is also better able to grow and change when it needs to. The backend system is also easier to take care of.

9. Performance Testing

We did performance testing to see how the backend system works when a lot of people are using it. We pretended to be users to test the system. We looked at how it takes for the system to respond how much work it can handle, if there are any delays in alerts and if the system is stable. The backend system worked well with up to 1000 users. Did not slow down much or have big problems

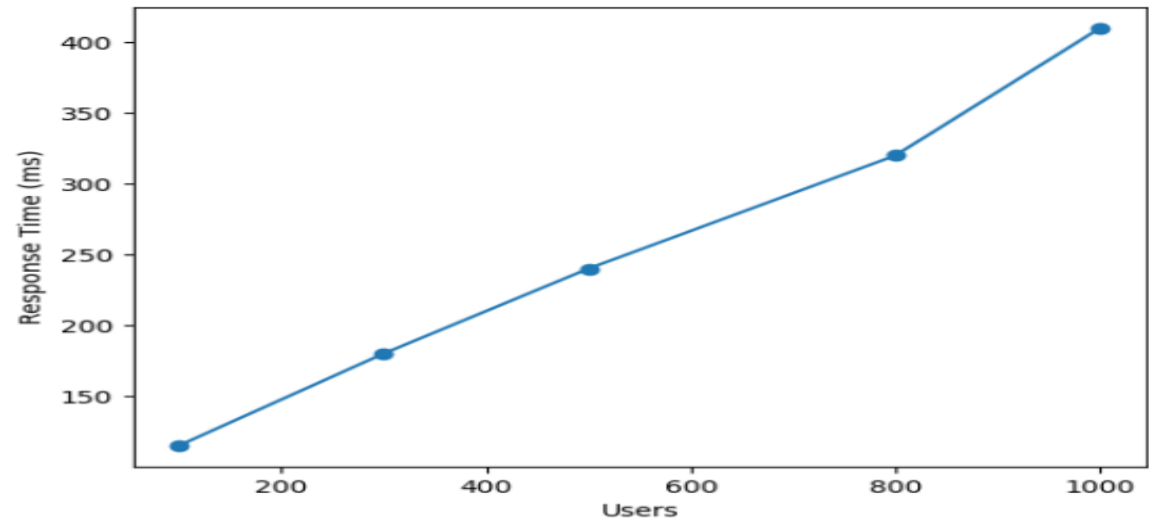


Fig 3: Response Time Graph

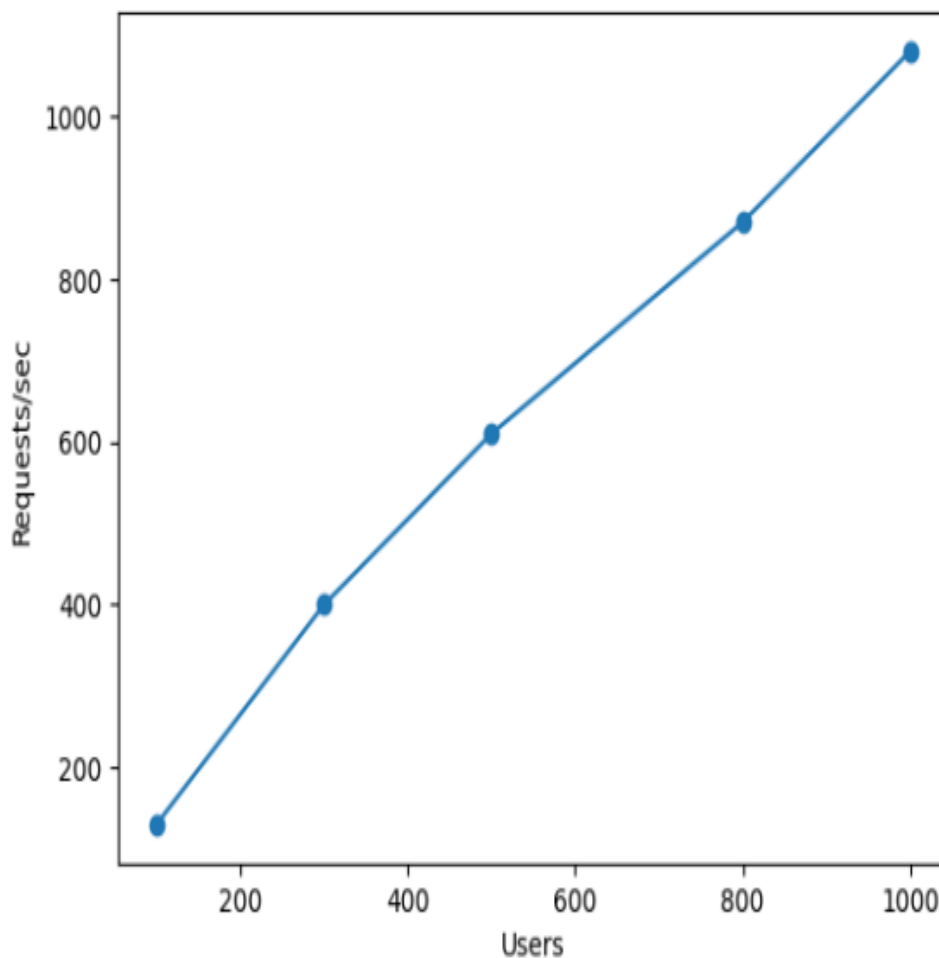


Fig 4:Throughput Graph

This graph shows how response time changes when more users join.

When 100 people used the system it responded in 115 milliseconds.

As more people used the system the response time got longer.

When one thousand people used the system the response time of the system went, up to four hundred and ten milliseconds.

The response time got longer. The system still worked fine.

People could still use it without issues.

The backend system was still okay, with many users.

The response time was still good. Users could still access the system quickly.

The system handled 1000 users well.

10. Results

We tested our expense tracking system by simulating users to see how it would behave under different loads. The main goal was to see how fast the system responded and how data the system could handle and how quickly the system sent alerts and how stable the system was.

The system handled one thousand users at the time without any big problems. As more users joined the response time got a bit slower. It was still stable and usable. Whenever users went over their spending limits the system sent alerts away and there wasn't any noticeable delay.

The performance results that we measured are shown below:

Number of Users	Response Time (ms)	Throughput (Requests/sec)
100	115 ms	130
300	180 ms	400
500	240 ms	610
800	320 ms	870
1000	410 ms	1080

The results show that our system performed consistently well even when lots of users were using it. This means our system design is good at handling users.

Overall our system worked reliably responded quickly. Sent accurate alerts. We did not have any problems when we were testing.

11 .Discussion

Our systems design was made up of lots of layers. This made it really easy to build and to test and to maintain the systems design. We could just focus on one layer at a time with the systems design.

Each part of the system had one job. That made it simpler to fix bugs and make changes without messing up the whole system.

The alert system worked well

It checked spending limits right after each expense was added to the database.

This allowed users to get notifications quickly when they went over their limits.

The alert system helps users keep track of their finances.

It also helps them manage their budgets better with our alert system.

Our alert system is helpful, for users.

Compared to designs our layered structure made the system more organized and flexible. We followed practices for building the backend like secure communication, data validation and structured database management. These decisions helped make the system stable and reliable.

The performance results show that our system is suitable for medium-sized financial applications where stable performance and real-time updates are important

12. Future Work

There are ways we can improve our system in the future.

One idea is to use machine learning to see how people spend their money and figure out what they will spend in the future. Machine learning can look at what people're buying and how much they are spending. This can help us understand what people will buy and how money they will spend on things in the future. We can use machine learning to make guesses about what people will do with their money. Machine learning is very good at looking at things that have happened and using that information to make predictions about what will happen. So we can use machine learning to look at how people spend their money and make predictions about what they will spend their money on in the future. We can use machine learning to look at the way people spend their money and then use machine learning to make guesses about what people will buy with their money. Machine learning is really good, at helping us understand what people will do with their money. This would help the system tell users how to manage their money and make a budget plan. Another thing we can do is make an app for the system. This will make it easier for people to use the system.

We can also make the system work better by using a microservices architecture. This means that different parts of the system can work on their own. The machine learning system will still be a part of this. Other things we might do in the future are test the system with a lot of users, on the cloud. We can use the machine learning system to make the tests better.

We can also add security features, like advanced encryption and stronger authentication to protect financial data. Finally integrating with banking APIs would allow users to automatically import their transactions and reduce manual data entry.

13. Conclusion

This research is about the design and implementation of a real-time expense tracking system that can handle a lot of users. The system has a structure with different layers like presentation and data layers. This structure makes it easy to work on the system and fix problems.

The system uses connections to talk to its different parts and it has special codes to keep the data safe. The alert system sends notifications when someone spends much money. This helps people keep track of their money.

The alert system is really helpful, for people who want to know where their money is going.

We tried out the system to see how well it works.

We found out that the alert system can handle a lot of users at the time. The alert system does not slow down when many people are using it. The expense tracking system worked fine when 1000 people were using it at the same time.

The results show that a good backend system is very important for financial applications like expense tracking systems. The system we made is a base, for building expense tracking applications that can handle a lot of users and keep their data safe and it can be made even better in the future. The expense tracking system is an example of how a well-designed system can help people manage their expenses more effectively.

References

1. Y. B. Kalpana, S. Krishnamoorthy, Varsha, S. Hari, Supriyan and S. Surya "Expense Tracker," Iconic Research and Engineering Journals, vol. 9 No. 5 Pp. 531–540, 2025. This Expense Tracker by Kalpana et al. Is a read.
2. A. Rahangdale, M. Balwani, S. Tichkule, V. Tiwari and M. Tahir "Digital Expense Tracker " International Journal of Progressive Research in Engineering Management and Science vol. 5 No. 10 Pp. 750–753, 2025. The Digital Expense Tracker by Rahangdale et al. Caught my attention.
3. P. Rajesh, M. D. Sai, Y. J. R. Narasimha, K. M. S. Prasad, A. Naveen, B. J. S. S. Kumar and G. TejVarma "Expense Manager: A Comprehensive Web-Based Solution for Financial Tracking " International Journal for Research in Applied Science and Engineering Technology, 2025. This Expense Manager solution seems comprehensive.
4. I. D. Oladipo, T. K. Odunayo U. A. Gusau, Z. A. Abdullahi A. M. Muhammad and A. Abdulkhaliq "Web-Based Student Expense Tracking System " Journal of Institutional Research Big Data Analytics and Innovation 2025. The Web-Based Student Expense Tracking System by Oladipo et al. Is worth reading.
5. S. Gehlot, P. Gupta, K. Chavan, Bhavish and P. Sutar, "An Expense Tracker " International Journal for Research in Applied Science and Engineering Technology, 2024. This Expense Tracker by Gehlot et al. Provides some insights.

6. D. Rahinj, D. Pawar, V. Ghegade, S. A. Shelke and S. N. Divekar "Money Management System " International Journal of Scientific Research in Science, Engineering and Technology 2024. The Money Management System, by Rahinj et al. Seems useful.
7. Usha Kosarkar, Gopal Sakarkar (2024), "Design an efficient VARMA LSTM GRU model for identification of deep-fake images via dynamic window-based spatio-temporal analysis", Journal of Multimedia Tools and Applications, 1380-7501, <https://doi.org/10.1007/s11042-024-19220-w>
8. Anupam Chaube, Usha Kosarkar "Enhanced Deep Learning-Based Feature Analysis for Copy-Move Forgery Detection", 2024 Journal of Information Systems Engineering and Management,9(4s) e-ISSN:2468-4376, <https://www.jisem-journal.com/>