Information Systems Engineering

Mateo Safla and Diego Iza

Faculty of Engineering and Applied Sciences

Eng. Juan Pablo Guevara Gordillo

June 23, 2024

Github Repository link:

- https://github.com/mateo1011s/YourWallet

Description:

The task involves creating a wallet program that allows users to purchase and sell items using a Java-based interface. The program should enable users to input an amount and then click either a 'Purchase' or 'Sale' button to execute the transaction. Upon completing a transaction, a receipt is generated containing the product ID, transaction date, transaction type (purchase or sale), and the amount. The initial balance of the wallet is $100, which is updated accordingly after each transaction.

Functional Requirements

1. **Transaction Logging**: The program must allow the user to record a transaction by entering an amount and selecting the appropriate button (purchase or sale).
2. **Receipt Generation**: After completing a transaction, the program must generate a receipt that includes a unique product ID, the transaction date, the type of transaction (purchase or sale), and the amount.
3. **Balance Update**: The program must automatically update the wallet balance after each transaction, adding the amount in case of a sale and subtracting the amount in case of a purchase, starting with an initial balance of $100.
4. **User Interface**: The interface must include a field for entering the transaction amount and two buttons, one for purchasing and one for selling.

5. **Transaction History**: The program must maintain a history of all transactions, allowing the user to view previous transactions, including all receipt details.

Non-Functional Requirements

1. **Performance**: The program must process transactions and generate receipts in less than 2 seconds to ensure a smooth user experience.
2. **Usability**: The user interface must be intuitive and easy to use, allowing users to perform transactions without prior training.
3. **Security**: The program must protect user data and transactions using encryption methods to ensure the integrity and confidentiality of the information.
4. **Scalability**: The program must be designed to allow the inclusion of new features in the future, such as integration with other payment methods or management of multiple currencies.
5. **Compatibility**: The program must be compatible with different operating systems and Java versions to ensure it can be used in various environments without compatibility issues.

Conclusions about the task:

The development of this wallet program demonstrates the application of basic Java programming skills, including GUI creation, event handling, and data management. The implementation of transaction logging and receipt generation ensures that users have a clear

record of their activities. The task highlights the importance of user-friendly interfaces and the need for efficient data processing to maintain a smooth user experience.

Recommendations about the task:

1. **Enhance Security**: Implement security measures such as encryption for sensitive data to ensure the integrity and confidentiality of user information.
2. **Expand Features**: Consider adding more features like transaction history export, multi-currency support, and integration with other payment systems to enhance the program's utility.
3. **Improve Usability**: Ensure the user interface is intuitive and accessible, possibly including tooltips or help sections to guide users.
4. **Optimize Performance**: Focus on optimizing the code to ensure quick response times for all operations, particularly for transaction processing and receipt generation.
5. **Conduct Testing**: Perform thorough testing to identify and fix any bugs or issues, ensuring the program runs smoothly under various conditions and **inputs.**

References

- Bloch, J. (2018). Effective Java (3rd ed.). Addison-Wesley.
- Gaba, D. (2023). Best Practices for Developing Secure Java Applications. Journal of Computer Science and Technology. Retrieved from https://www.jcst.com/articles/best-practices-secure-java

- Java Documentation. (2024). Transactions in Java. Retrieved from https://docs.oracle.com/javase/tutorial/essential/threads/transactions.html
- Johnson, A. (2021). Implementing Financial Transaction Systems in Java. International Journal of Software Development. Retrieved from https://www.ijsd.com/financial-transaction-systems-java