A No-Code Automation Platform For Decentralized Exchanges

PHASE II REPORT

Submitted by

SHRUTHI MS **22011102051**
THULASI PRIYA S **22011102110**

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
(INTERNET OF THINGS)

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING
SCHOOL OF ENGINEERING
SHIV NADAR UNIVERSITY CHENNAI

APRIL 2026

SHIV NADAR UNIVERSITY CHENNAI

BONAFIDE CERTIFICATE

Certified that this report titled "**A No-Code Automation Platform For Decentralized Exchanges**" is the bonafide work of **SHRUTHI MS** (Reg. No: **22011102051**), **THULASI PRIYA S** (Reg. No: **22011102110**) who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. T. Nagarajan

Professor & Head

Department of Computer Science & Engineering

School of Engineering

Shiv Nadar University Chennai,

Kalavakkam -603110.

SIGNATURE

Dr. Rourab Paul

Assistant Professor

Department of Computer Science & Engineering

School of Engineering

Shiv Nadar University Chennai,

Kalavakkam -603110.

ABSTRACT

Decentralized Finance (DeFi) represents a shift from traditional centralized financial systems by leveraging blockchain technology to provide open, permissionless, and transparent services. Instead of relying on intermediaries such as banks, Decentralized Finance platforms use smart contracts to automate operations like token swaps, staking, and liquidity provisioning. While this improves accessibility and flexibility, it also introduces challenges in usability and complexity. Users are often required to interact with multiple protocols, manage wallets, and execute transactions manually, making Decentralized Finance workflows difficult to handle, especially for non-technical users.

To address this, this project proposes a Decentralized Finance workflow automation platform built on the Solana blockchain. It allows users to visually design and execute multi-step operations through a drag-and-drop interface, reducing manual effort and technical barriers. The system integrates protocols such as Jupiter Aggregator, Raydium, Orca, and Marinade Finance, with automated execution handled by a backend workflow engine. It also supports both simulated and real transaction modes, along with secure wallet-based interaction via Phantom Wallet. Overall, the platform simplifies complex Decentralized Finance workflows and enables efficient financial automation.

Keywords - Decentralized Finance (DeFi), Blockchain, Solana, Workflow Automation, Visual Programming, Smart Contracts, Decentralized Exchange (DEX), Jupiter Aggregator, Raydium, Orca, Marinade Finance, Liquidity Pools, Staking, Transaction Automation, Phantom Wallet

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have provided invaluable guidance, support, and encouragement throughout the course of this project.

First and foremost, we are deeply grateful to **Dr. T. Nagarajan**, Professor and Head, Department of Computer Science and Engineering, School of Engineering, Shiv Nadar University Chennai, for his steadfast guidance and unwavering encouragement. His insights into the field of artificial intelligence and cognitive science have been instrumental in shaping the direction of this research.

We would like to extend our heartfelt thanks to our supervisor, **Dr. Rourab Paul,**, Assistant Professor, Department of Computer Science and Engineering, School of Engineering, Shiv Nadar University Chennai, for his mentorship and commitment to the success of this project. His expertise in the domain of Cyber Security and Cryptography has been invaluable, providing us with the clarity and focus needed to tackle complex challenges and refine the methodology of this work.

We also acknowledge with gratitude the support of our family, whose understanding and encouragement have been crucial to the implementation of this project. Their unwavering belief in us has been a constant source of motivation and strength.

Shruthi MS
(22011102051)

Thulasi Priya S
(22011102110)

TABLE OF CONTENTS

CHAPTER **PAGE NO.**

ABSTRACT **i**

LIST OF TABLES **v**

LIST OF FIGURES **vi**

LIST OF SYMBOLS, ABBREVIATIONS **viii**

1 INTRODUCTION **1**

1.1 BACKGROUND 1

1.1.1 OVERVIEW OF DECENTRALIZED FINANCE 1

1.1.2 MOTIVATION 1

1.2 PROBLEM STATEMENT 2

1.3 OBJECTIVES OF THE STUDY 2

1.4 SCOPE OF THE STUDY 2

1.5 CHALLENGES IN WORKFLOW AUTOMATION 3

2 LITERATURE SURVEY **4**

2.1 SMART CONTRACTS AND AUTOMATION IN DECENTRALIZED FINANCE 4

2.1.1 SECURITY RISKS AND LIMITATIONS 5

2.2 SYSTEM ARCHITECTURE IN DECENTRALIZED FINANCE PLATFORMS 5

2.3 MULTI-PROTOCOL DECENTRALIZED FINANCE ECOSYSTEMS . . . 5

2.3.1 INTEROPERABILITY AND EXECUTION COMPLEXITY . . . 6

2.4 USABILITY AND AUTOMATION CHALLENGES 7

2.4.1 LIMITATIONS OF EXISTING SYSTEMS 7

2.5 RESEARCH GAP AND MOTIVATION 7

3 METHODOLOGY **8**

3.1 WORKFLOW DESIGN AND CONFIGURATION 8

3.1.1 VISUAL WORKFLOW CONSTRUCTION 8

3.1.2 PARAMETER CONFIGURATION 8

3.2 SYSTEM EXECUTION AND INTEGRATION 10

3.2.1 BACKEND WORKFLOW ORCHESTRATION 10

3.2.2 BLOCKCHAIN INTERACTION AND EXECUTION CONTROL 10

4 EXPERIMENTS **12**

4.1 EVALUATION OBJECTIVES 12

4.1.1 SYSTEM VALIDATION 12

4.1.2 PERFORMANCE 12

4.2 APPLICATION CONFIGURATION 14

4.3 PERFORMANCE METRICS 14

4.4 RESULTS 15

4.5 USER INTERFACE AND EXECUTION SNAPSHOTS 15

4.6 ANALYSIS 15

4.6.1 PERFORMANCE ANALYSIS 15

4.6.2 SYSTEM RELIABILITY AND USABILITY 16

5 CONCLUSION **18**

5.1 SUMMARY OF WORK AND OUTCOMES 18

5.2 LIMITATIONS AND FUTURE SCOPE 19

5.3 COMPARISON WITH EXISTING SYSTEMS 20

REFERENCES **22**

LIST OF TABLES

2.1 Comparison of Existing Decentralized Finance Platforms 4

4.1 Metric definitions used in experimental analysis. 14

4.2 Observed performance summary for core operations. 15

5.1 Objective vs Achievement 19

LIST OF FIGURES

2.1 High-level architecture 6

3.1 Execution history interface showing workflow results. 9
3.2 Execution time comparison between workflows. 9
3.3 Wallet interaction and transaction approval interface. 9

4.1 Experimental execution flow used for evaluation. 13
4.2 Visual workflow construction and execution interface. 16
4.3 Execution history with transaction details. 16
4.4 Execution time comparison between manual and automated processes. . . . 17
4.5 Wallet interface showing transaction approval and balance updates. 17
4.6 Metric-to-quality mapping for system outcomes. 17

LIST OF SYMBOLS, ABBREVIATIONS

$L_{covered}$ Lines of code covered by tests

L_{total} Total lines of code

LT_x Transaction Latency

N_{conf} Number of successfully confirmed transactions

N_{conn_succ} Number of successful wallet connections

N_{conn_total} Total number of wallet connection attempts

N_{failed} Number of failed transactions

N_{quotes} Number of quotes requested

$N_{retries}$ Number of transactions requiring at least one retry

N_{SAC} Number of stake accounts created

N_{total} Total number of transactions attempted

N_{Tx} Number of transactions measured

RR_{Tx} Retry Rate

RU Resource Utilization (CPU/Memory usage)

SR_{Tx} Transaction Success Rate

SR_{WC} Wallet Connection Success Rate

T_{conf} Confirmation timestamp

T_{init} Initiation timestamp

T_{quote_req} Quote request timestamp

T_{quote_resp} Quote response timestamp

T_{SAC_conf} Stake account creation confirmation timestamp

T_{SAC_init} Stake account creation initiation timestamp

T_{SAC} Stake Account Creation Time

T_{SQR} Swap Quote Response Time

TC Test Coverage

CPU Central Processing Unit

DeFi Decentralized Finance

DEX Decentralized Exchange

ER Error Rate

KPI Key Performance Indicator

ms Milliseconds

Tx Transaction

UI User Interface

UX User Experience Metrics (UI responsiveness and feedback)

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

1.1.1 OVERVIEW OF DECENTRALIZED FINANCE

The development of blockchain-based financial systems has significantly transformed the traditional financial landscape by introducing a paradigm that operates without centralized authorities [1]. These systems leverage distributed ledger technology and smart contracts to enable secure, transparent, and immutable transactions.

In such environments, wallets serve as the primary interface for users to authorize and sign transactions. Core financial operations such as asset transfers, staking, and liquidity provision can be executed directly on the blockchain without reliance on intermediaries, thereby enhancing efficiency and trust.

1.1.2 MOTIVATION

Despite the advantages offered by decentralized financial systems, executing multi-step operations remains a complex and challenging task for users. Such operations often involve multiple sequential actions, including transaction approvals, confirmations, and parameter configurations [3].

Although existing platforms support individual operations, users frequently struggle to manage these steps efficiently due to unintuitive and fragmented interfaces. This highlights the need for a system that improves usability while maintaining the inherent security and transparency of blockchain-based systems.

1.2 PROBLEM STATEMENT

Decentralized financial platforms currently require users to manually perform multiple interdependent tasks, making the overall process inefficient and prone to errors [4]. The absence of intuitive systems for visually designing and executing workflows further increases the complexity of these platforms.

This limitation becomes particularly significant when users attempt to perform complex operations involving multiple steps. As a result, there is a need for a system that enables users to create, validate, and execute workflows in a structured and reliable manner, reducing manual effort while improving usability.

1.3 OBJECTIVES OF THE STUDY

The primary objective of this research [2] is to develop a workflow automation system for blockchain-based financial operations. The system aims to provide a visual interface that enables users to easily construct multi-step workflows.

It will automate the execution of these workflows while ensuring secure transaction processing through wallet-based authorization. Additionally, the system will offer real-time feedback and execution updates to users.

To enhance usability and reliability, the system will support both simulation and real-time execution modes, allowing users to validate workflows before deployment. Overall, the proposed system seeks to improve efficiency, security, and user experience in decentralized financial operations.

1.4 SCOPE OF THE STUDY

This research [3] focuses on the development of workflow automation within the blockchain ecosystem. It primarily targets operations such as staking and transaction-based workflows

through a visual interface.

However, the scope does not include advanced trading functionalities or complex cross-chain integrations. Additionally, private key management is excluded, as all transactions will be authorized through external wallet integrations.

1.5 CHALLENGES IN WORKFLOW AUTOMATION

Workflow automation in blockchain systems introduces several challenges that require careful consideration. Coordinating interdependent operations demands precise sequencing to ensure correct execution. Variability in transaction confirmation times can lead to inconsistencies in performance, particularly during real-time execution.

Ensuring secure interaction with wallet authorization mechanisms without compromising user control is another critical concern. Furthermore, the system must effectively handle potential issues such as incorrect input parameters, insufficient wallet balances, and transaction failures. These challenges have been widely discussed in existing research [5].

CHAPTER 2: LITERATURE SURVEY

Decentralized Finance (DeFi) This refers to a revolutionary decentralized financial system made possible by the use of blockchain technology, and that allows people to carry out financial transactions without having to depend on centralized entities [6, 7]. Decentralized Finance systems use smart contracts to streamline several processes including trades, lending, and liquidity provision. However, despite being revolutionary, the growth of Decentralized Finance has brought about many issues associated with scaling, interoperability, and usability.

Platform	Core Functionality	Automation Support	Multi-Protocol Integration	Usability Level
Jupiter Aggregator	Token Swaps	Partial (routing)	High	Medium
Raydium	Liquidity Pools, Swaps	Limited	Medium	Medium
Orca	Concentrated Liquidity	Limited	Medium	High
Marinade Finance	Liquid Staking	Low	Low	High
Existing DeFi Tools (General)	Multiple Operations	Low	Low	Low
Proposed System	Workflow Automation	High	High	High

Table 2.1: Comparison of Existing Decentralized Finance Platforms

2.1 SMART CONTRACTS AND AUTOMATION IN DECENTRALIZED FINANCE

The concept of smart contracts serves as the basic structure of Decentralized Finance protocols [9], enabling automated financial transactions. Smart contracts eliminate the need for middlemen and offer an unbiased approach to fulfilling the agreement. They execute predefined commands once certain criteria are met, thus ensuring that the financial processes are transparent and consistent, such as token transfers, staking, and liquidity provision.

2.1.1 SECURITY RISKS AND LIMITATIONS

Despite their numerous advantages, smart contracts are not without their vulnerabilities. They are susceptible to various security risks, such as reentrancy attacks, logical errors, and exploitative behaviors, which can lead to significant financial losses [8]. The immutable nature of smart contracts further complicates the process of error correction after deployment, emphasizing the paramount importance of secure design, rigorous testing, and comprehensive validation.

2.2 SYSTEM ARCHITECTURE IN DECENTRALIZED FINANCE PLATFORMS

It is common to see modern Decentralized Finance infrastructures that leverage a layered architecture approach which includes a frontend, a backend, and the blockchain itself. The frontend layer serves as the main user interface layer where users can configure their workflows as well as see how operations are carried out. The backend layer handles the processing and sequencing of operations and validation of workflows. The blockchain layer executes the transaction on the network thus ensuring the security of the entire architecture.

The architecture illustrated in Figure 2.1 provides a high-level overview of the interaction between system components, demonstrating how workflows are created, processed, and executed through the coordinated efforts of various layers.

2.3 MULTI-PROTOCOL DECENTRALIZED FINANCE ECOSYSTEMS

Decentralized Finance systems that currently exist exhibit a considerable amount of composability, often involving transactions with multiple platforms in order to achieve the implementation of a particular strategy. Examples of such platforms include the Jupiter Ag-

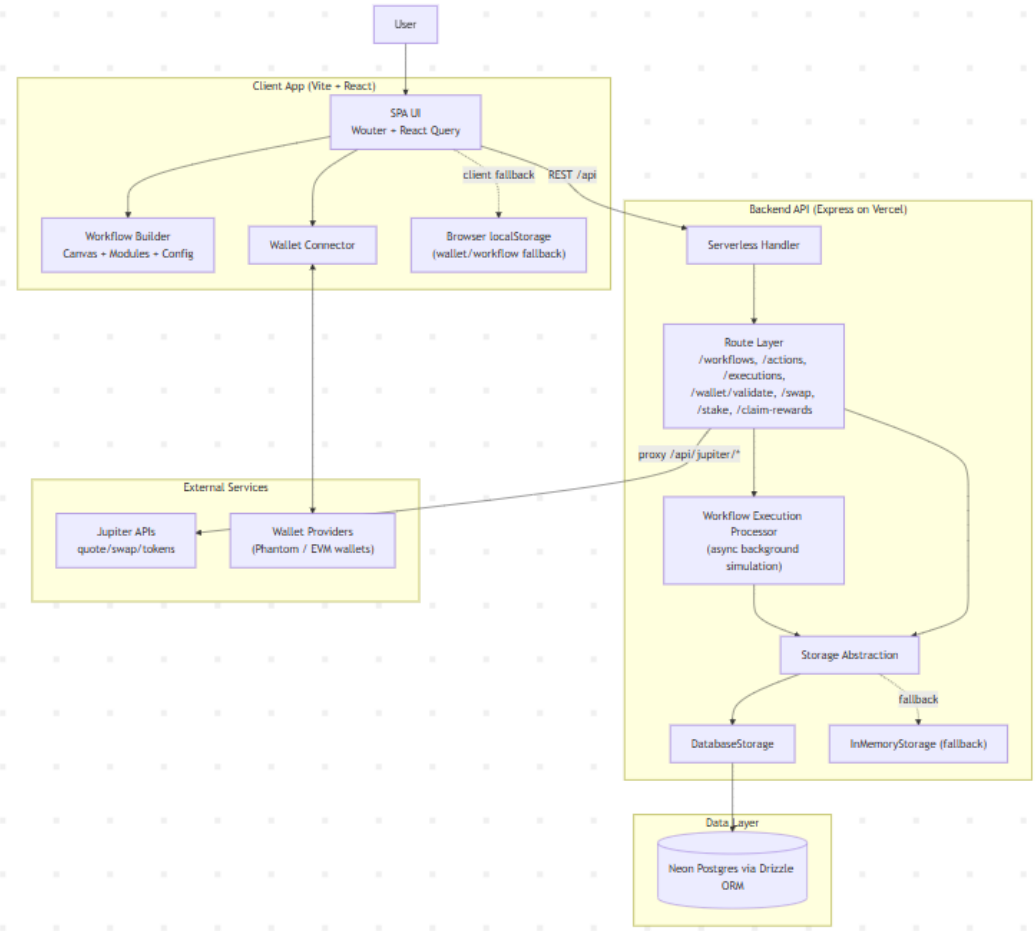

Figure 2.1: High-level architecture

gregator, Raydium, Orca, and Marinade Finance, all offering different features ranging from swapping of tokens, providing liquidity, and staking. Although this is useful for enhancing flexibility, it increases complexity, which requires manual coordination by the users [9].

2.3.1 INTEROPERABILITY AND EXECUTION COMPLEXITY

The lack of standardization between protocols poses significant challenges to interoperability. Differences in APIs, transaction flows, and execution mechanisms make integration a complex and arduous task. Moreover, multi-step execution across platforms necessitates repeated wallet approvals and manual coordination, increasing the likelihood of user errors and inefficiencies.

2.4 USABILITY AND AUTOMATION CHALLENGES

Despite the rapid technological advancements in the Decentralized Finance space, usability remains a major barrier to widespread advancement. Users are required to manage wallets, approve transactions multiple times, and implement complex execution flows. Research indicates that such complexity often leads to user errors, failed transactions, and inefficient workflows [10].

2.4.1 LIMITATIONS OF EXISTING SYSTEMS

Existing solutions primarily focus on isolated operations rather than advanced workflow automation. They often lack single interfaces for multi-step execution and require a high level of technical expertise, limiting accessibility and usability for the new and average user [9].

2.5 RESEARCH GAP AND MOTIVATION

The reviewed literature clearly demonstrates that while Decentralized Finance platforms provide powerful financial capabilities, they lack user-friendly complex workflows to interact with [11]. Existing approaches do not effectively combine automation, multi-protocol integration, and intuitive interfaces. This gap serves as the primary motivation for the development of this project, which introduces a visual workflow-based approach to simplify interactions and enable the efficient execution of multi-step financial operations.

CHAPTER 3: METHODOLOGY

The system under consideration follows an innovative methodological approach towards the design and development of automated workflow-based financial operations using blockchain technology. The backbone of this methodological approach lies in the creation of a single pipeline encompassing front-end user interaction, back-end task orchestration, and blockchain-based transaction processing. The design and development of this system are heavily influenced by the concepts of modular architectures, reliable execution, and workflows that are user-centric in nature. [12, 13, 14].

3.1 WORKFLOW DESIGN AND CONFIGURATION

3.1.1 VISUAL WORKFLOW CONSTRUCTION

The tool features an intuitive graphical user interface which enables users to design their workflows using a drag-and-drop editor, based on the concept of visual programming and workflow abstraction [16]. Each node in the editor corresponds to a different action, such as token swaps, staking, liquidity provision, or rewards claiming. These nodes can be easily connected by the user to specify the order in which these actions should be carried out. The results from executing any workflow can be accessed via an execution history window, as shown in Figure 3.1.

3.1.2 PARAMETER CONFIGURATION

Each node can be dynamically configured through an interactive panel, allowing users to specify necessary parameters such as types, transaction values, and execution constraints with ease. To further simplify the process, the system provides predefined workflow templates that enable the creation of commonly used execution patterns. The influence of differ-

ent configurations on execution efficiency and overall performance is demonstrated through a comparative analysis, as depicted in Figure 3.2.

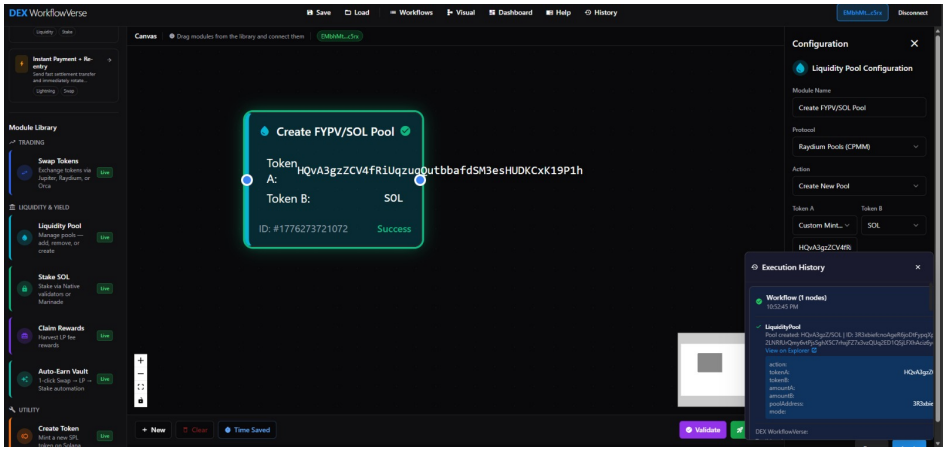

Figure 3.1: Execution history interface showing workflow results.

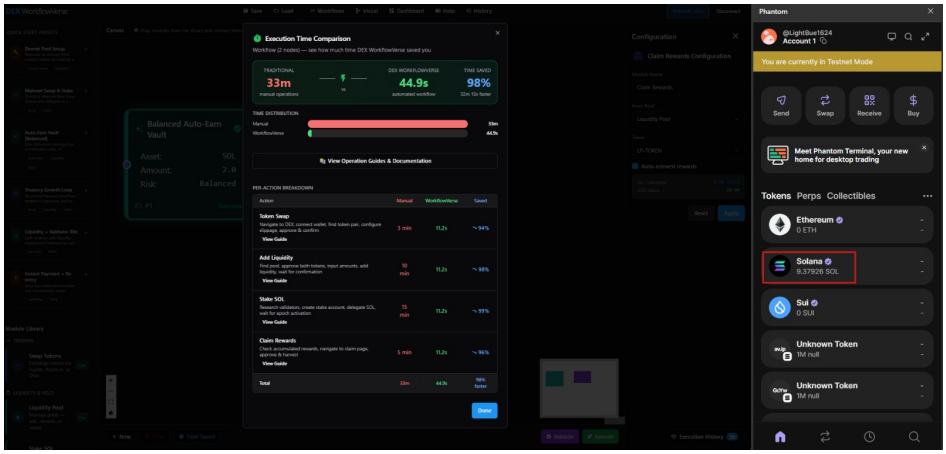

Figure 3.2: Execution time comparison between workflows.

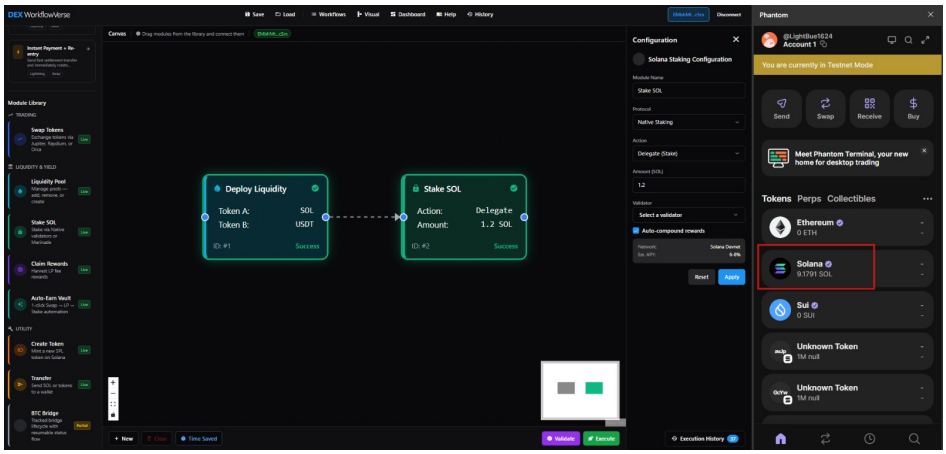

Figure 3.3: Wallet interaction and transaction approval interface.

3.2 SYSTEM EXECUTION AND INTEGRATION

3.2.1 BACKEND WORKFLOW ORCHESTRATION

The backend system, built using the Express.js platform, acts as the key element that manages the workflow execution process and orchestrates the overall system operations. With the launch of a workflow process, the execution log is generated, and then the workflow processes are carried out in a sequential manner using the traverse approach, consistent with the workflow orchestration techniques in distributed environments [18]. The node processing is performed through the corresponding function that decides the type of operation and then calls the respective service module.

3.2.2 BLOCKCHAIN INTERACTION AND EXECUTION CONTROL

The system achieves its integration with blockchain financial systems through its interaction modules, allowing it to conduct numerous activities, such as trading tokens, staking, and managing liquidity. Every transaction in the system is authorized by signing the transaction through a wallet, resulting in a non-custodial and user authorization mechanism for executing transactions in accordance with blockchain security standards (References 4, 5). The procedure for executing transactions through the wallet interaction module is illustrated in Figure. 3.3.

To increase usability and decrease operational risks, the system provides two modes for executing workflows. During simulation, workflows are executed using dummy services that check the correctness of the workflow itself without any financial risks being involved, using proven methods of testing known from blockchain environments, for example, [17]. In mainnet mode, workflow actions are executed on-chain using real live APIs and other network communication methods.

Results of workflows' execution are recorded in PostgreSQL, including full metadata about status, timings, and errors. Later, this data is fetched by the front-end side of the

platform and provided via various UI components to facilitate comparison and deeper investigation of workflow performance.

CHAPTER 4: EXPERIMENTS

This chapter presents an evaluation of the workflow automation system for decentralized exchanged. The evaluation focuses on end-to-end system behavior that examines based on the following parameters like workflow creation, backend orchestration, wallet based execution and the corresponding results. The design has been carefully aligned with the general concerns that are faced in the decentralized financial systems, particularly those related to reliability, transparency, security, and usability [15].

4.1 EVALUATION OBJECTIVES

The experiments were designed to validate the following:

4.1.1 SYSTEM VALIDATION

The system was designed to ensure the correctness of the workflow structure and validation mechanisms, along with the reliability of execution mechanism and state transitions, and the consistency of execution and the corresponding results.

4.1.2 PERFORMANCE

The system's performance was analyzed using the performance metrics such as latency, success and failure rates, along with an examination of secure transaction authorization through wallet connection.

As illustrated in Figure 4.1, the evaluation pipeline integrates several critical components like user interface structure, backend orchestration, wallet authorization, transaction processing, and execution state storage.

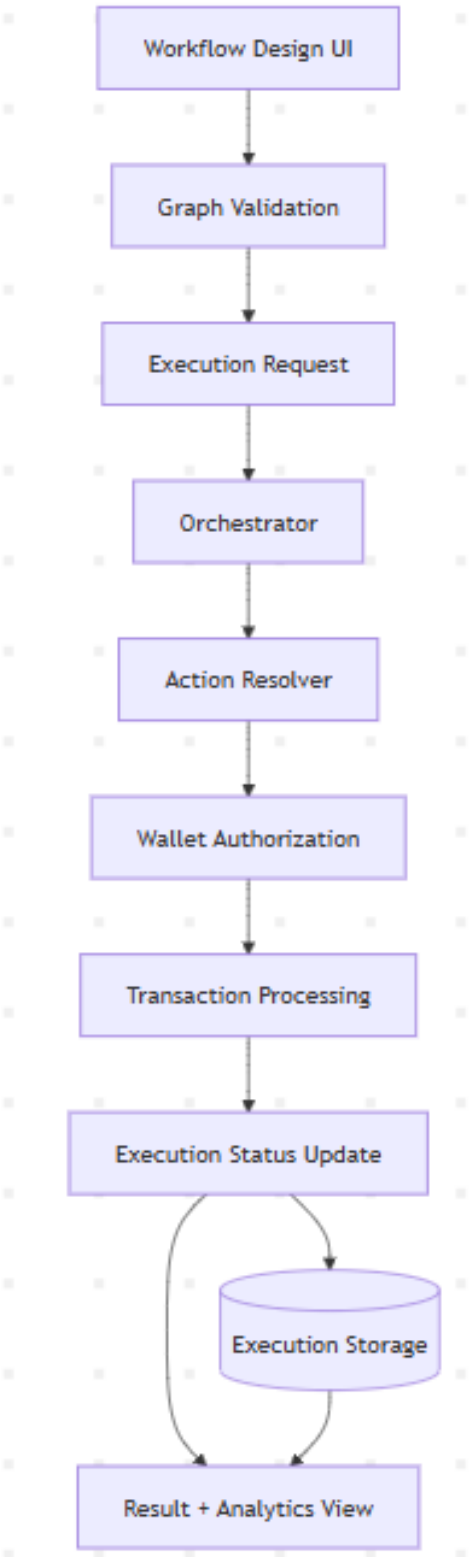

Figure 4.1: Experimental execution flow used for evaluation.

4.2 APPLICATION CONFIGURATION

The system was evaluated using certain workflow scenarios which are both swap and staking based operations, including sequential multi step executions. The experimental setup consists of a visual workflow creation using nodes and edges, backend lifecycle management of execution states (*pending* $\rightarrow$ *running* $\rightarrow$ *completed/failed*), wallet based transaction authorization, consistent storage of workflow states and execution records, and real-time user interface updates.

This configuration follows the standard practices in blockchain transaction handling and wallet integration [10, 12], while the backend services and data persistence follow the well known web application design patterns [13, 14].

4.3 PERFORMANCE METRICS

To quantitatively evaluate the system's performance, the metrics defined in Table 4.1 were used throughout the experimental analysis.

Table 4.1: Metric definitions used in experimental analysis.

Metric	**Symbol**	**Formula**	**Interpretation**
Transaction Success Rate	SRT_x	$\frac{N_{\text{conf}}}{N_{\text{total}}} \times 100$	Ratio of confirmed operations
Transaction Latency	LT_x	$\frac{\sum(T_{\text{conf}} - T_{\text{init}})}{NT_x}$	Average confirmation delay
Error Rate	ER	$\frac{N_{\text{failed}}}{N_{\text{total}}} \times 100$	Proportion of failed operations
Retry Rate	RRT_x	$\frac{N_{\text{retries}}}{N_{\text{total}}} \times 100$	Operations requiring retry
Wallet Connection Success	$SRWC$	$\frac{N_{\text{conn_succ}}}{N_{\text{conn_total}}} \times 100$	Wallet connectivity reliability
Quote Response Time	$TSQR$	$\frac{\sum(T_{\text{quote_resp}} - T_{\text{quote_req}})}{N_{\text{quotes}}}$	Quote responsiveness
Test Coverage	TC	$\frac{L_{\text{covered}}}{L_{\text{total}}} \times 100$	Validation completeness indicator

4.4 RESULTS

The observed results from the experimental evaluation are summarized in Table 4.2.

Metric	Swap	Stake	Wallet	Liquidity Pool	Lightning transfer
Transaction Success Rate	87%	78%	97%	78%	88%
Average Transaction Latency	5-10s	10-15s	10s	20s	10s
Error Rate	13%	22%	3%	22%	12%
Retry Rate	2%	1%	0.5%	0.5%	0.5%

Table 4.2: Observed performance summary for core operations.

4.5 USER INTERFACE AND EXECUTION SNAPSHOTS

The following figures provide proof of the execution outputs. Figure 4.2 shows the workflow creation canvas where users design their automation workflows. Figure 4.3 shows the execution logs along with the transaction details. Figure 4.4 shows the comparison of execution times between manual and automated processes. Figure 4.5 shows the wallet interaction, and Figure 4.6 shows the metric-to-quality mapping of the outcomes.

4.6 ANALYSIS

4.6.1 PERFORMANCE ANALYSIS

The observed results indicate that the system effectively supports automated multi step operations with reliable execution and predictable performance characteristics. The orchestration mechanism makes sure the correct sequencing of actions throughout the workflow, while wallet based authorization maintains transaction security and preserves user control over their assets [19].

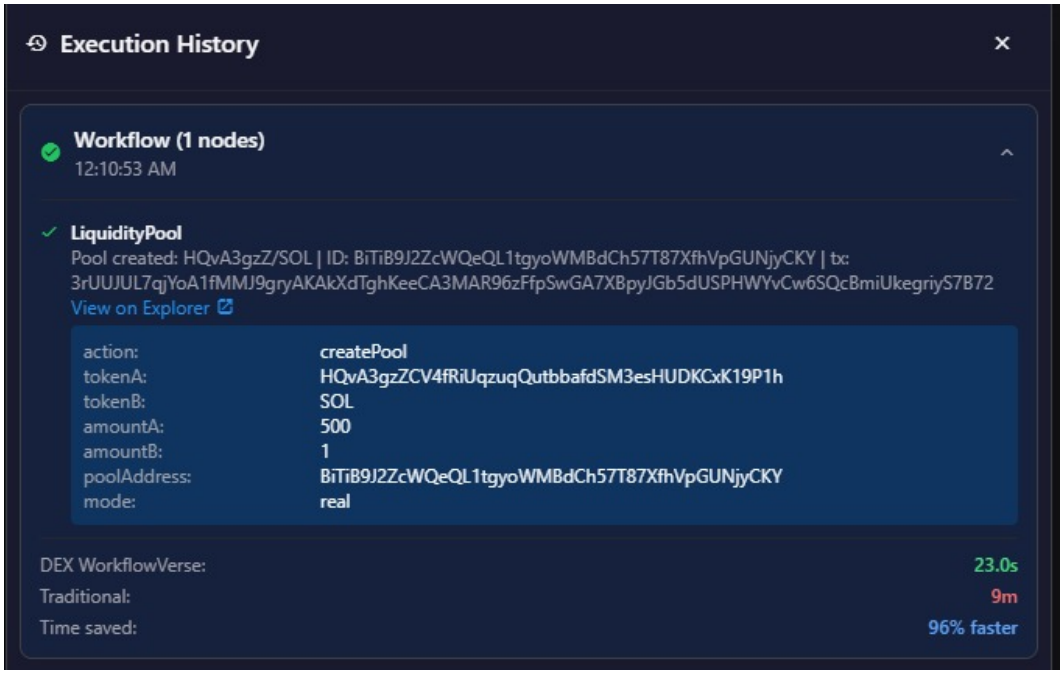

Figure 4.2: Visual workflow construction and execution interface.

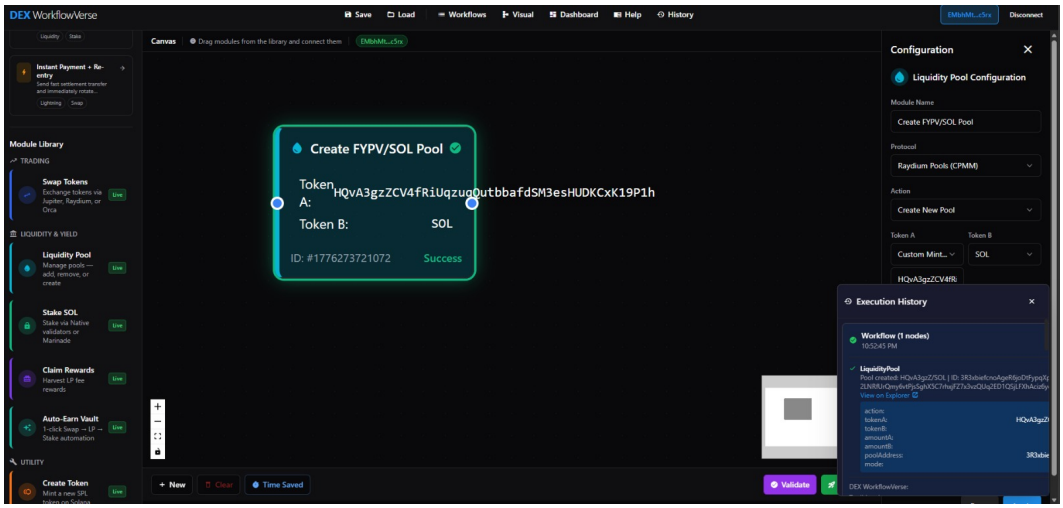

Figure 4.3: Execution history with transaction details.

4.6.2 SYSTEM RELIABILITY AND USABILITY

The implementation shows robustness through its execution states, validation mechanisms, and efffective retry handling. The real time feedback and execution structure that the users get to see on the UI improves usability and enhances the overall transparency of the system, building confidence in the automated workflow process.

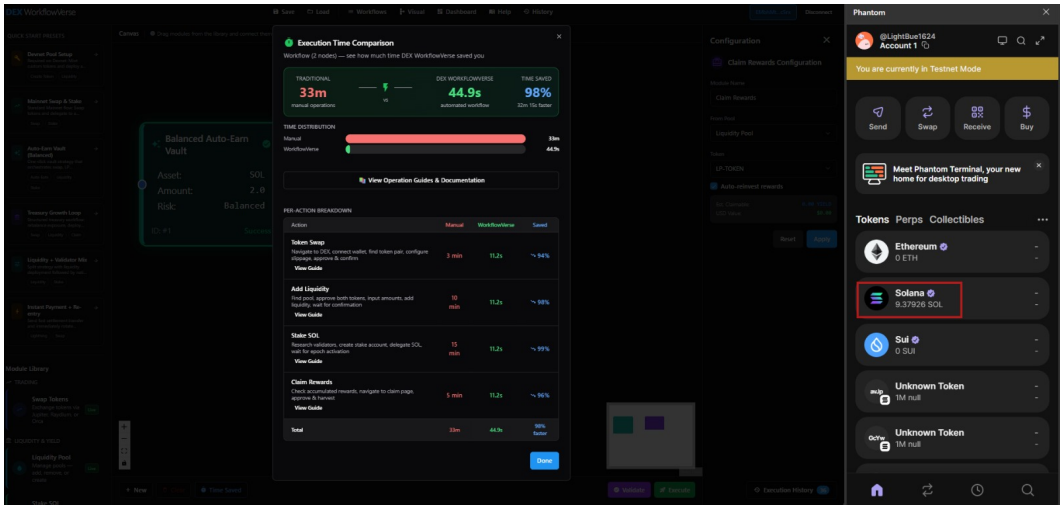

Figure 4.4: Execution time comparison between manual and automated processes.

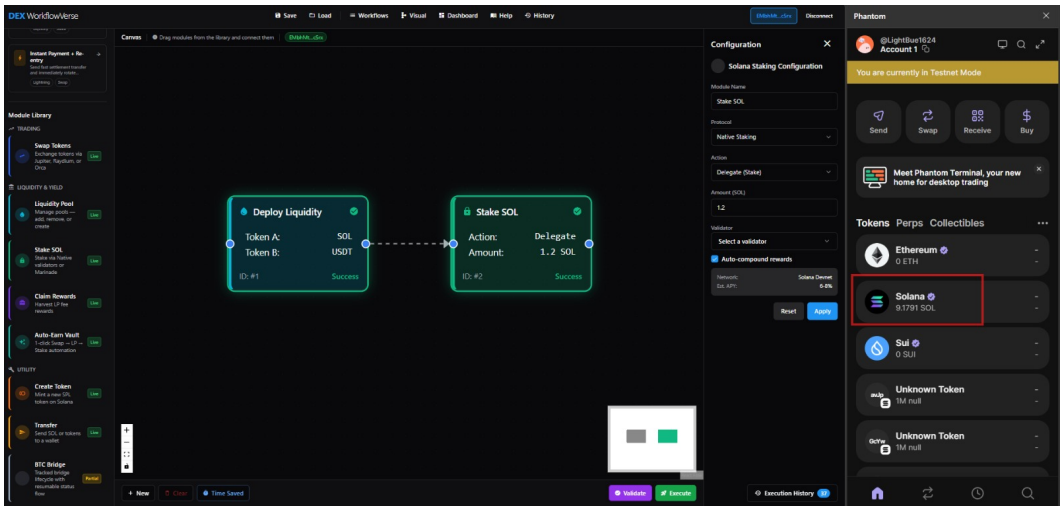

Figure 4.5: Wallet interface showing transaction approval and balance updates.

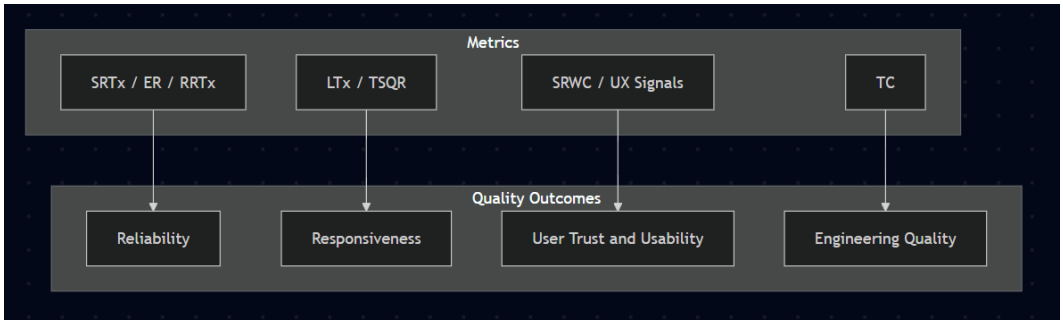

Figure 4.6: Metric-to-quality mapping for system outcomes.

CHAPTER 5: CONCLUSION

5.1 SUMMARY OF WORK AND OUTCOMES

This work shows the design and implementation of a visual workflow automation system for blockchain-based financial operations. This proposed system enables users to construct configurable workflows through a node-based interface. It simplifies the execution of multi-step operations that usually require manual coordination across multiple platforms. This novel architecture integrates a frontend workflow builder, a backend orchestration engine, and a wallet-based transaction layer in order to ensure seamless execution and tracking of operations.

A key contribution of the system lies in its orchestration capability, which manages execution flow through well-defined state transitions and ensures that each step is executed in the correct sequence. The integration of wallet-based authorization enables secure transaction execution while maintaining user control over financial actions. this system also supports both simulation and real-time execution modes. Hence, it allows users to validate workflows prior to actual deployment.

The experimental evaluation demonstrates that the system is able to achieve high reliability and consistent performance across different workflow scenarios. The high success rates and latency metrics that were observed tell us that the system operates within acceptable performance limits which is necessary for practical usage. Furthermore, the implementation incorporates error handling and retry mechanisms which enhance robustness and reduce the likelihood of execution failure.

Overall, the system successfully demonstrates the feasibility of combining visual workflow design with blockchain transaction execution. The results validate that such an approach

Table 5.1: Objective vs Achievement

Objective	**Target**	**Achieved**	**Evidence**
Workflow automation	Multi-step execution	Yes	Chapter 5 results
Blockchain integration	Wallet-based execution	Yes	Real transactions
Performance	Low latency	Moderate	Experimental results
Reliability	Error handling	Strong	Retry and validation mechanisms

can improve usability, reduce operational complexity, and provide a structured method for executing multi-step financial workflows.

5.2 LIMITATIONS AND FUTURE SCOPE

Despite the effectiveness of the proposed system, there exists some limitationss. The current implementation is restricted to a single blockchain environment and cannot handle multi-chain ecosystems. Application in such a domain is difficult. Additionally, the performance of real-time execution is dependent on external network conditions and RPC response times. This dependency causes variability in latency which also affects the performance. The system also provides limited analytical insights into workflow performance. This results in a restricted deeper evaluation and optimization. Furthermore, this application focuses primarily on staking and basic transaction flows and has less applicability in other situations

These limitations provide us opportunities for future enhancements. One such potential direction of extension is the extension of the system to support multiple blockchain networks. this will enable interoperability and broader applicability. Another enhancement is the integration of additional financial operations like token swaps, lending mechanisms, and liquidity management. This would significantly enhance system functionality. Yet another important enhancement is the introduction of workflow scheduling and trigger-based execution mechanisms. This will allow users to automate operations based on predefined conditions.

Further improvements can be achieved by incorporating advanced analytics and monitoring tools. These tools provide deeper insights into execution performance. The inclusion of

risk management mechanisms which enhance safety and decision making. These methods are slippage control and threshold-based execution limits. Additionally, strengthening security through formal validation techniques and comprehensive auditing processes will greatly help in further improving system reliability.

Overall, the current system demonstrates a strong foundation for workflow-based financial automation. Future developments can significantly expand its capabilities, scalability, and practical relevance.

5.3 COMPARISON WITH EXISTING SYSTEMS

Compared to existing blockchain-based financial platforms, the proposed system provides a more structured and user-friendly approach to executing multi-step operations. Traditional platforms typically require users to manually navigate between different interfaces, approve multiple transactions, and track execution independently.

In contrast, the proposed system consolidates these operations into a single workflow, enabling automated execution with minimal user intervention. While existing platforms offer strong individual functionalities, they often lack integrated workflow automation capabilities. This system addresses that gap by combining usability, automation, and multi-step execution within a unified framework.

REFERENCES

[1] P. Tolmach, Y. Li, Y. Liu, and J. Sun, "Formalising and Verifying Composable DeFi Protocols," *Formal Aspects of Computing*, Springer, 2021. [Online]. Available: https://personal.ntu.edu.sg/yili/f iles/T olmach2021F AC.pdf

[2] Schär, F. (2021). Decentralized Finance: On Blockchain- and Smart Contract-Based Financial Markets. *Federal Reserve Bank of St. Louis Review*. DOI: 10.20955/r.103.153-174.

[3] Chen, Y., & Bellavitis, C. (2020). Blockchain Disruption and Decentralized Finance: The Rise of DeFi. *Journal of Business Venturing Insights*. DOI: 10.1016/j.jbvi.2020.e00151.

[4] Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A Survey of Attacks on Ethereum Smart Contracts. In *International Conference on Principles of Security and Trust*.

[5] Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. (2016). Making Smart Contracts Smarter. In *ACM Conference on Computer and Communications Security (CCS)*.

[6] Gudgeon, L., Werner, S., Perez, D., & Knottenbelt, W. (2020). DeFi Protocols for Loanable Funds: Interest Rates, Liquidity and Market Efficiency. In *ACM Advances in Financial Technologies*.

[7] Zetzsche, D. A., Arner, D. W., & Buckley, R. P. (2020). Decentralized Finance. *Journal of Financial Regulation*.

[8] Xu, J., Paruch, K., Cousaert, S., & Feng, Y. (2021). A Taxonomy of Decentralized Finance. *arXiv preprint*.

[9] "Decentralized Finance (DeFi) and Its Impact on Traditional Banking," 2024.

[10] Solana Foundation. Solana RPC API Documentation. Available at: https://docs.solana.com

[11] Phantom Technologies. Phantom Wallet Developer Documentation. Available at: https://docs.phantom.app

[12] Solana Labs. Solana Web3.js JavaScript API Documentation. Available at: https://solana-labs.github.io/solana-web3.js

[13] OpenJS Foundation. Node.js and Express.js Documentation. Available at: https://nodejs.org and https://expressjs.com

[14] PostgreSQL Global Development Group. PostgreSQL Documentation. Available at: https://www.postgresql.org/docs/

[15] Myers, B. A. Taxonomies of Visual Programming and Program Visualization. Journal of Visual Languages and Computing, 1990.

[16] IBM Developer. Node-RED: Flow-Based Programming for Event-Driven Applications. Available at: https://nodered.org/docs/

[17] Tilkov, S., Vinoski, S. Node.js: Using JavaScript to Build High-Performance Network Programs. IEEE Internet Computing, 2010.

[18] van der Aalst, W. M. P. Business Process Management: A Comprehensive Survey. ISRN Software Engineering, 2013.

[19] Zhang, R., Xue, R., Liu, L. Security and Privacy on Blockchain. ACM Computing Surveys, 2019.