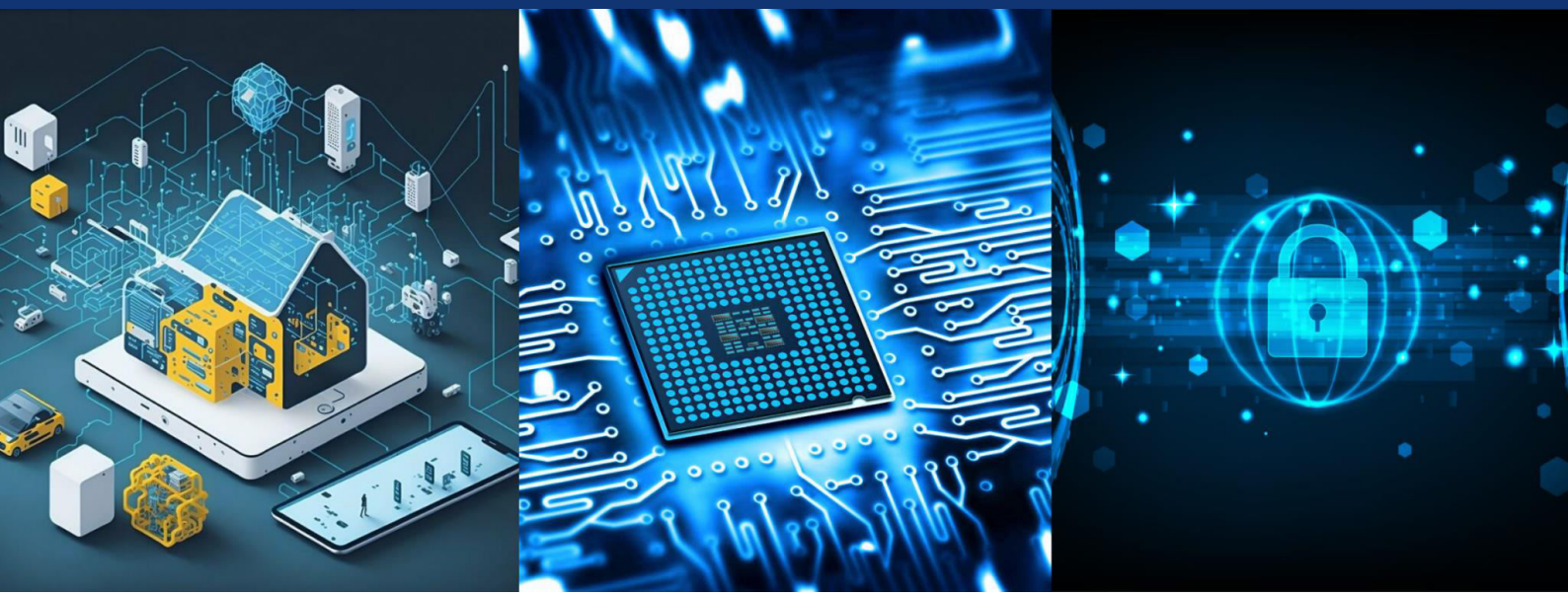
IJIRCCE

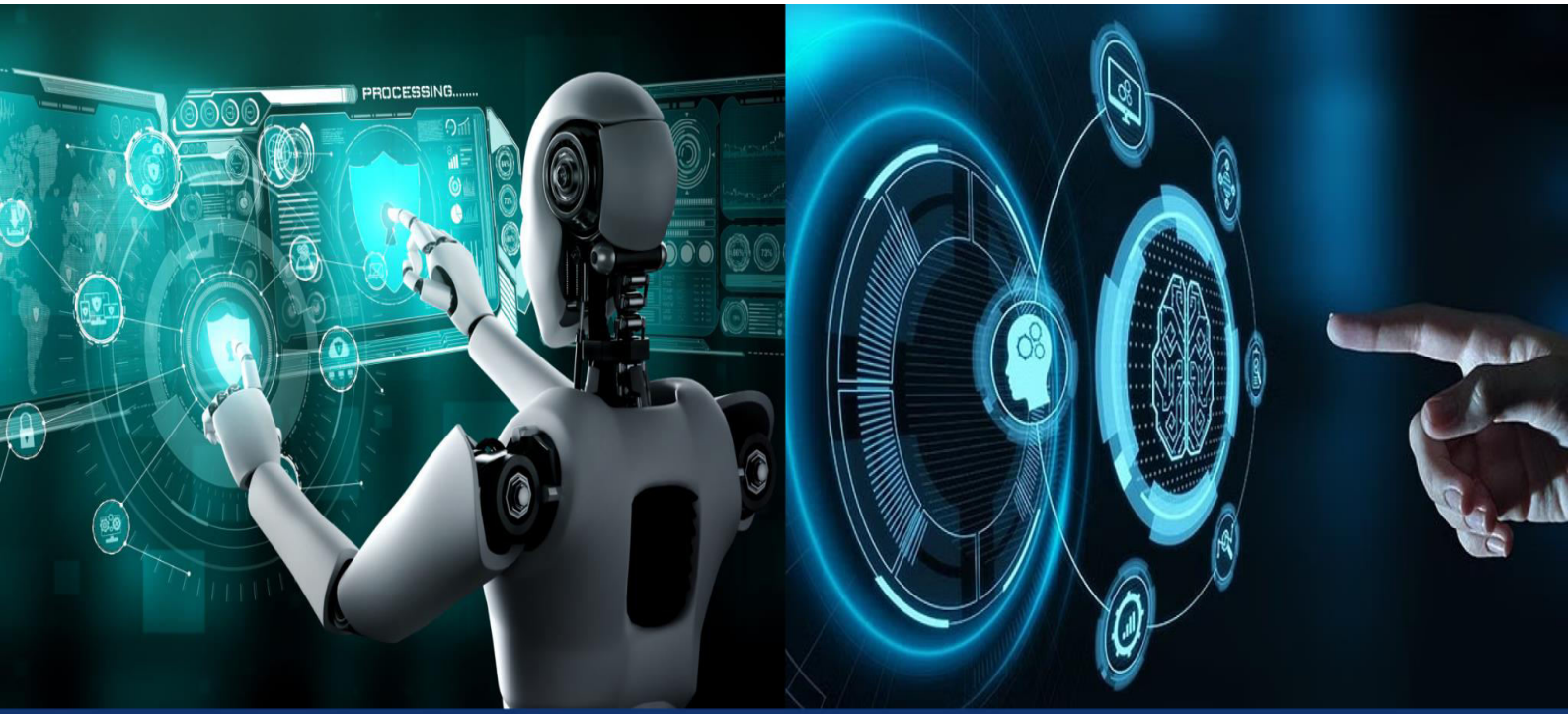
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 6, June 2026

© 2026 IJIRCCE | Volume 14, Issue 6, June 2026 DOI: 10.15680/IJIRCCE.2026.1406048

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Implementation of Java Data Structure Using Swing

Khanitha Imen T[1], N. Ramesh[2], A. Sudhakar[3]

PG Student, Department of Computer Science and Engineering, Bharathidasan Engineering College, Tirupattur, Tamil Nadu, India[1]

Assistant Professor, Department of Computer Science and Engineering, Bharathidasan Engineering College, Tirupattur, Tamil Nadu, India[2]

Professor and Head, Department of Computer Science and Engineering, Bharathidasan Engineering College, Tirupattur, Tamil Nadu, India[3]

ABSTRACT: Data structures constitute the fundamental building blocks of computer science and software engineering applications. Although data structures are extensively taught in academic curricula, students frequently encounter difficulties in understanding their internal operations and dynamic behaviour through traditional teaching methodologies. Static diagrams and theoretical explanations often fail to provide sufficient insight into the execution of operations such as insertion, deletion, traversal, and searching. To address this challenge, an interactive visualization platform has been developed using Java Swing to facilitate the effective learning of both linear and non-linear data structures.

The proposed system integrates Arrays, Stacks, Queues, Circular Queues, Singly Linked Lists, Doubly Linked Lists, Binary Search Trees, and Graph Traversal algorithms within a unified graphical environment. Users can perform various operations through a graphical interface and observe real-time structural changes. The system was developed using Java and Eclipse IDE following a modular architecture. Experimental evaluation demonstrated successful implementation of all supported operations and confirmed the effectiveness of visualization in improving conceptual understanding. The developed application serves as a valuable educational tool for students, instructors, and self-learners studying data structure concepts.

KEYWORDS: Data Structure Visualization, Java Swing, Educational Software, Binary Search Tree, Graph Traversal, Linked List, Queue, Stack.

I. INTRODUCTION

Data structures are essential components of modern software systems. They provide organized methods for storing, managing, and processing data efficiently. The performance of algorithms and applications largely depends on the selection and implementation of appropriate data structures. Commonly used structures such as arrays, stacks, queues, linked lists, trees, and graphs form the foundation of numerous computing applications.

Despite their significance, students often experience challenges while learning data structures due to the abstract representation of concepts. Traditional teaching methods generally rely on classroom lectures, textbooks, and static diagrams. Although these methods provide theoretical knowledge, they are often insufficient for demonstrating dynamic operational behavior.

Visualization-based learning has emerged as an effective educational approach that enhances comprehension through graphical representation. By observing the internal modifications occurring during execution, learners gain a deeper understanding of algorithmic behavior and structural transformations.

The primary objective of this work is to design and implement an interactive visualization platform capable of demonstrating multiple linear and non-linear data structures within a single application. The proposed system enables users to perform operations and instantly observe corresponding changes through a graphical user interface developed using Java Swing.

© 2026 IJIRCCE | Volume 14, Issue 6, June 2026 DOI: 10.15680/IJIRCCE.2026.1406048

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. RELATED WORK

Data structure visualization has attracted significant attention in computer science education. Various educational tools have been developed to improve student understanding through graphical demonstrations of algorithms and data manipulation processes.

Earlier visualization systems primarily focused on demonstrating individual algorithms or isolated data structures. While such systems improved learner engagement, they often lacked integration and required separate interfaces for different structures. Additionally, several visualization tools were developed as web-based applications, introducing dependency on internet connectivity and browser compatibility.

Java-based educational systems have gained popularity due to their portability, object-oriented design, and graphical capabilities. The Java Swing framework provides a rich collection of user interface components that facilitate the development of interactive educational software.

However, many existing systems concentrate on specific structures such as stacks, queues, or trees. The absence of a unified platform capable of visualizing both linear and non-linear structures limits educational effectiveness. The proposed work addresses this limitation by integrating multiple data structures into a single desktop application with a consistent graphical interface.

III. LITERATURE REVIEW

The visualization of data structures and algorithms has emerged as a significant research domain within computer science education due to its potential to enhance conceptual comprehension, computational thinking, and learner engagement. Conventional instructional methodologies primarily rely on theoretical explanations, static illustrations, and classroom demonstrations, which often fail to effectively convey the dynamic behavior of data structures during execution. Consequently, numerous visualization-based frameworks have been proposed to facilitate deeper cognitive understanding of algorithmic processes.

Garrido Abenza et al. introduced **VisualJVM**, an interactive visualization environment designed to support Java programming instruction. The framework provides graphical representations of object interactions, memory allocation, and execution flow, enabling learners to develop a more comprehensive understanding of object-oriented programming principles. Experimental findings demonstrated notable improvements in conceptual clarity and learning efficacy through the integration of visualization techniques.

Cattaneo et al. proposed the **Java Interactive Visualization Environment (JIVE)**, a software visualization platform that enables real-time observation of program execution. The system supports visual tracking of object creation, method invocation, and execution states, thereby facilitating program comprehension, debugging proficiency, and analytical reasoning. The study highlighted the pedagogical value of visualization in improving software engineering education.

Chen and Sobh developed a framework for **data structure visualization and algorithm animation**, emphasizing interactive graphical representation as a mechanism for improving algorithmic cognition. Their research demonstrated that step-by-step visualization significantly enhances the understanding of complex operations and promotes active learner participation during the instructional process.

Yang et al. investigated the impact of static and dynamic visualization methodologies within programming education. Their findings indicated that visualization-assisted learning environments contribute to improved knowledge retention, enhanced problem-solving capabilities, and stronger conceptual understanding when compared with traditional instructional approaches.

A critical analysis of existing literature reveals that most visualization platforms are designed to address specific algorithms, programming constructs, or individual data structures. Although these systems provide valuable instructional support, they often lack a unified framework capable of demonstrating a broad spectrum of linear and non-linear data structures within a single interactive environment. Furthermore, many existing solutions offer limited opportunities for user-driven experimentation and real-time operational analysis.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

To address these limitations, the proposed system presents a comprehensive visualization framework developed using Java Swing. The platform integrates Arrays, Stacks, Queues, Circular Queues, Singly Linked Lists, Doubly Linked Lists, Binary Search Trees, and Graph Traversal algorithms within a unified graphical architecture. By enabling real-time visualization of insertion, deletion, traversal, searching, Breadth First Search (BFS), and Depth First Search (DFS) operations, the system enhances conceptual comprehension of abstract computational structures and provides a robust instructional platform for advanced computer science education and algorithmic analysis.

IV. EXISTING SYSTEM

Data structures are traditionally taught through classroom lectures, textbooks, and static diagrams. While these approaches provide a theoretical foundation, they do not effectively demonstrate the dynamic behavior of data structure operations. Students often find it difficult to visualize how elements are inserted, deleted, traversed, or searched during program execution.

Several educational visualization tools have been developed to improve learning outcomes. However, many existing systems focus only on specific algorithms or individual data structures and lack integration within a single platform. Some tools require internet connectivity, complex configuration, or specialized software environments, making them less accessible for beginners. Additionally, existing visualization systems often provide limited user interaction and insufficient support for both linear and non-linear data structures.

Limitations of Existing System

- Dependence on static diagrams and theoretical explanations.
- Difficulty in understanding dynamic data structure operations.
- Limited visualization support for multiple data structures.
- Lack of an integrated learning environment.
- Reduced learner interaction and practical experimentation.
- Insufficient support for both linear and non-linear data structures in a single application.

V. PROPOSED SYSTEM

The proposed system is an **Interactive Visualization System for Linear and Non-Linear Data Structures** developed using Java Swing and Eclipse IDE. The system provides a graphical learning environment that enables users to perform various operations and observe structural changes in real time.

The application integrates multiple data structure modules including Arrays, Stacks, Queues, Circular Queues, Singly Linked Lists, Doubly Linked Lists, Binary Search Trees, and Graph Traversal algorithms. Users can interact with each module through a graphical user interface and visualize operations such as insertion, deletion, searching, and traversal.
The system follows a modular architecture where each data structure is implemented independently while maintaining a common interface. This design improves usability, maintainability, and extensibility. By providing real-time visualization and user interaction, the proposed system enhances conceptual understanding and supports effective learning of data structure concepts.

Advantages of Proposed System

- Provides real-time visualization of data structure operations.
- Integrates multiple linear and non-linear data structures in a single platform.
- Offers an interactive and user-friendly graphical interface.
- Improves understanding of insertion, deletion, searching, and traversal operations.
- Supports educational learning and classroom demonstrations.
- Developed using Java Swing, ensuring platform independence and ease of deployment.
- Reduces complexity in understanding abstract data structure concepts.

Modules of Proposed System

1. Array Visualization Module
2. Stack Visualization Module

© 2026 IJIRCCE | Volume 14, Issue 6, June 2026 DOI: 10.15680/IJIRCCE.2026.1406048

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

3. Queue Visualization Module
4. Circular Queue Visualization Module
5. Singly Linked List Visualization Module
6. Doubly Linked List Visualization Module
7. Binary Search Tree Visualization Module
8. Graph Traversal Visualization Module (BFS and DFS)

VI. SYSTEM ARCHITECTURE

The architecture of the proposed system consists of three major layers:

A. Presentation Layer

Provides graphical interaction through Java Swing components including buttons, labels, text fields, and dialog boxes.

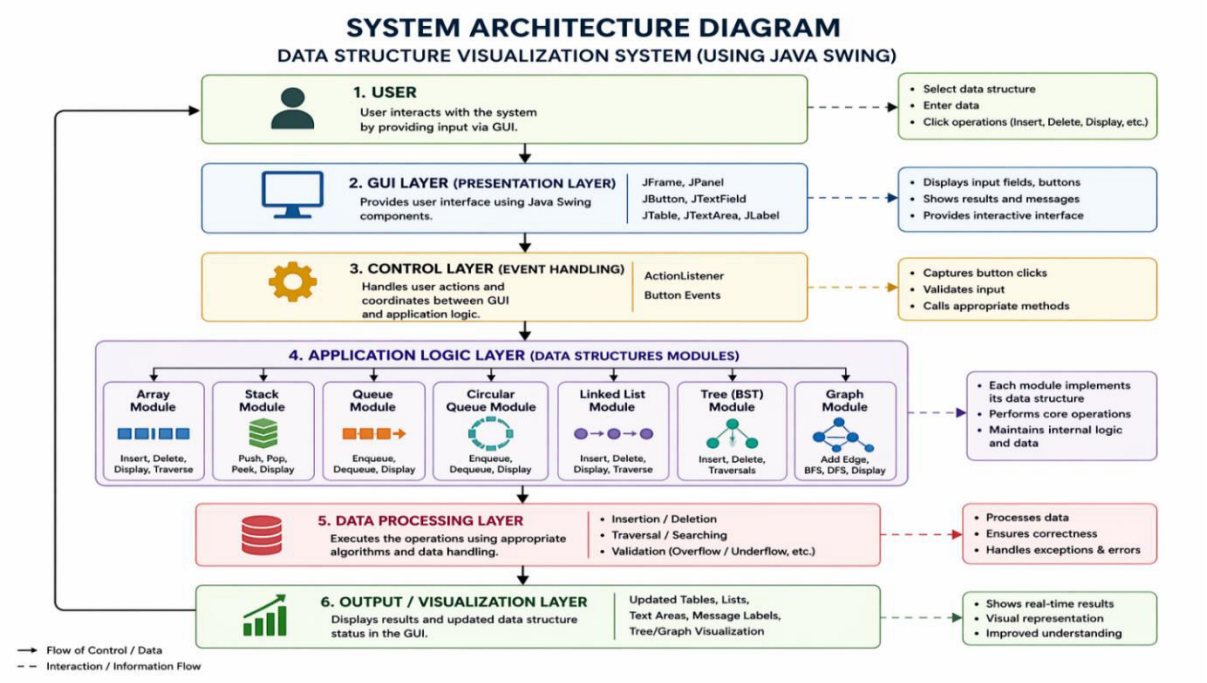
B. Processing Layer

Contains the implementation logic of individual data structures and algorithms.

C. Visualization Layer

Displays execution results and structural modifications generated by operations.

Fig 1. System Architecture



Input → GUI Layer → Processing Layer → Visualization Layer → Output Display

VII. IMPLEMENTATION

The application was implemented using Java Swing within Eclipse IDE. The system consists of multiple modules that visualize different data structures.

A. Main Menu Module

The Main Menu acts as the entry point of the application and provides access to all visualization modules.

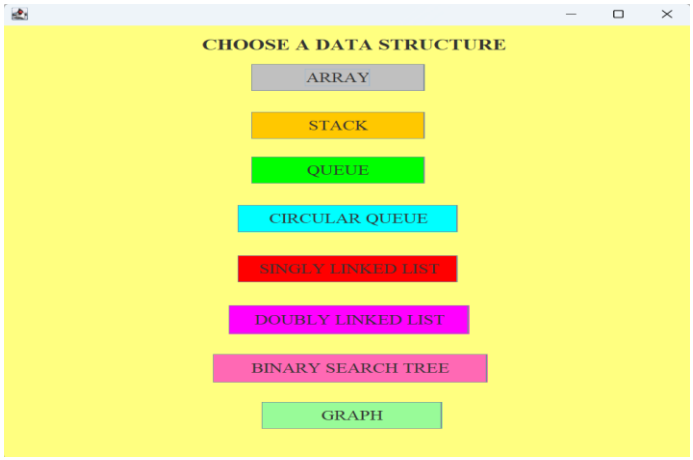
© 2026 IJIRCCE | Volume 14, Issue 6, June 2026 DOI: 10.15680/IJIRCCE.2026.1406048

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

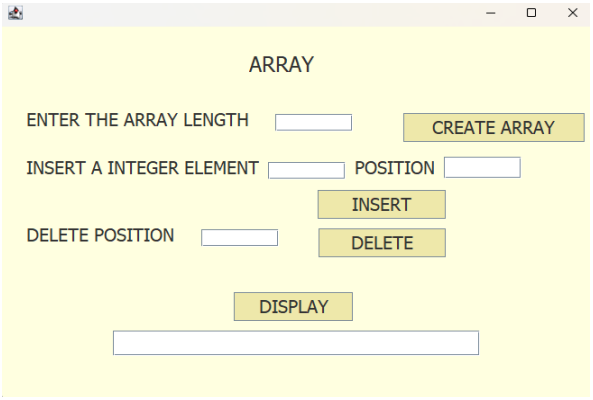
Fig. 2. Main Menu Interface



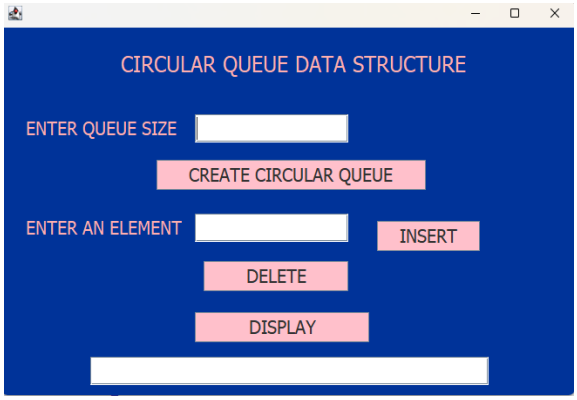
B. Linear Data Structure Modules

The system provides visualization support for Arrays, Stacks, Queues, and Circular Queues. Users can perform insertion, deletion, display, and searching operations depending on the structure being used.

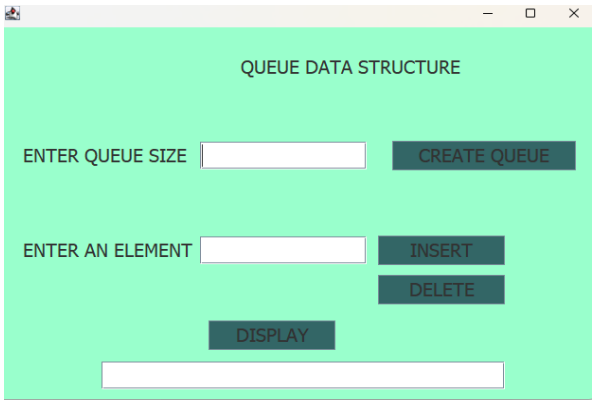
Fig. 3. Visualization of Linear Data Structure Modules



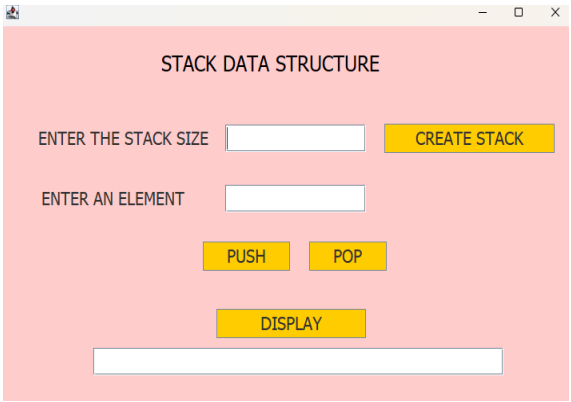
(a) Array Module



(b) Circular Queue Module



(c) Queue Module



(d) Stack Module

© 2026 IJIRCCE | Volume 14, Issue 6, June 2026 DOI: 10.15680/IJIRCCE.2026.1406048

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

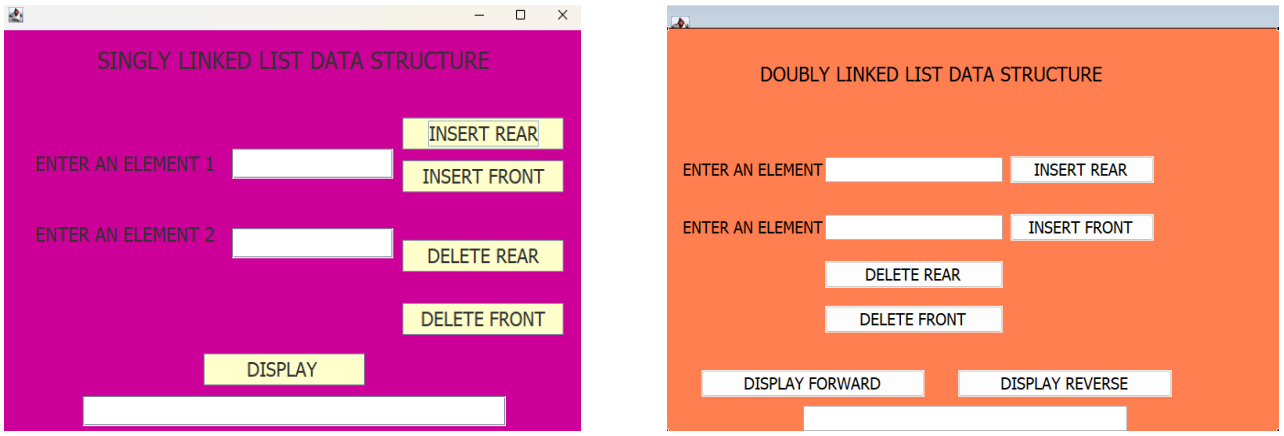
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

C. Linked List Modules

The application supports both Singly Linked Lists and Doubly Linked Lists. Users can perform insertion and deletion at different positions while observing node connectivity.

Fig. 4. Linked List Visualization Modules

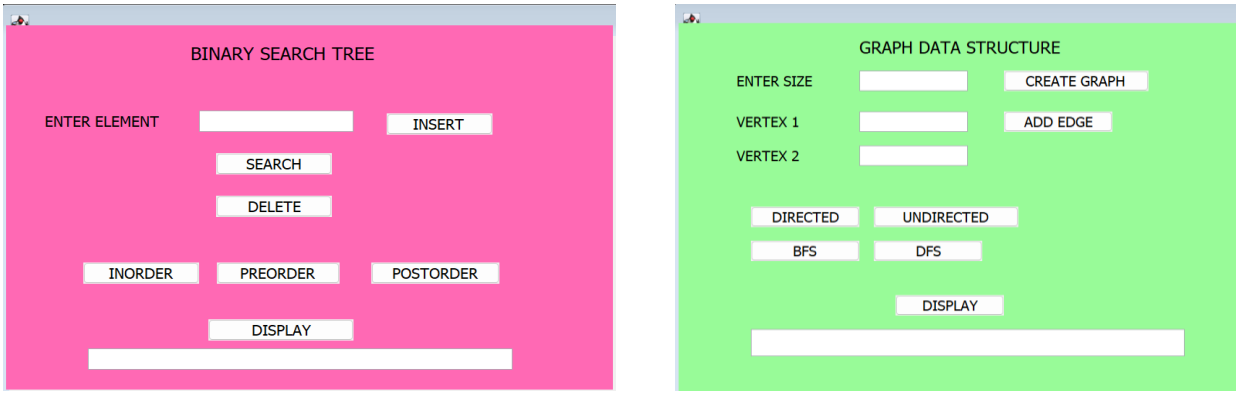


(a) Singly Linked List (b)Doubly Linked List

D. Non-Linear Data Structure Modules

Binary Search Tree and Graph Traversal modules demonstrate hierarchical and graph-based structures. The Graph module supports Breadth First Search (BFS) and Depth First Search (DFS) traversal.

Fig. 5. Non-Linear Data Structure Visualization Modules



(a) Binary Search Tree (b) Graph Traversal

VIII. RESULTS AND DISCUSSION

Testing was conducted across all modules to verify operational correctness.
The developed system successfully demonstrated:

- Array manipulation operations.
- Stack push and pop functionality.
- Queue and Circular Queue processing.
- Linked List node insertion and deletion.
- Binary Search Tree traversal.
- Graph traversal using BFS and DFS.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The graphical interface enabled users to observe dynamic structural changes immediately after execution.
The results indicate that the proposed system effectively demonstrates both linear and non-linear data structures within a unified environment.

IX. CONCLUSION

This paper presented the design and implementation of an Interactive Data Structure Visualization System using Java Swing. The developed application integrates multiple data structures into a single graphical platform and enables users to observe operations through real-time visualization.

The system improves understanding of insertion, deletion, traversal, and searching operations while providing an interactive learning experience. Experimental evaluation confirmed the successful implementation of all modules. The proposed platform can be effectively utilized in academic institutions for laboratory sessions, classroom demonstrations, and self-learning activities.

X. FUTURE SCOPE

Future enhancements may include:
1. AVL Tree visualization.
2. Red-Black Tree implementation.
3. Sorting algorithm animations.
4. Dijkstra's shortest path visualization.
5. Web-based deployment.
6. Mobile application support.
7. Complexity analysis visualization.
8. Interactive step-by-step algorithm execution.

REFERENCES

[1] T. Chen and T. Sobh, "A Tool for Data Structure Visualization and User-Defined Algorithm Animation," Frontiers in Education Conference, IEEE.
[2] Enhancing Object-Oriented Programming Education Using Visualization (IEEE)
[3] Robert Lafore, Data Structures and Algorithms in Java, Sams Publishing, 2003.
[4] Herbert Schildt, Java: The Complete Reference, McGraw-Hill Education, 2018.
[5] Y. Daniel Liang, Introduction to Java Programming and Data Structures, Pearson Education, 2019.
[6] "Java Swing Lab," ResearchGate, 2023. Accessed: Jun. 2026.
[7] S. P. Jadhav, "Graph Traversals and Its Applications," *International Journal of Computer Science and Mobile Applications (IJCSMA)*, vol. 10, no. 7, pp. 1–7, Jul. 2022, doi: 10.5281/zenodo.6838779.
[8] T. Sharma, S. Das, and A. Pal, "Swings – A Study," *International Journal of Research (IJR)*, vol. 1, no. 10, pp. 953–958, Nov. 2014.
[9] Sahni S., Data Structures, Algorithms and Applications in Java, Universities Press, 2005.
[10] "Data Structures Implementation through Java," ResearchGate, 2023. Accessed: Jun. 2026.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details