**MemoLoom - Flash card Application**

Nidhi Samarth, Manish Kadiyala, Vishal Tiwari

Product Development, Northeastern University

Information Systems, Northeastern University

Information Systems, Northeastern University

{kadiyala.m, samarth.n, tiwari.vi}@northeastern.edu

***Abstract* - The project aims to revolutionize the study of data structures and algorithms through a JavaFX-based flashcard application tailored for software engineering students. This solution addresses the inefficiencies of conventional study methods by providing a more engaging, interactive, and flexible learning tool that students can use on the go.**

**Our design process involves a user-centric approach where functionality meets simplicity. Flashcards can be personalized for different study needs, enabling users to focus on specific topics. Quizzes and spaced repetition algorithms are embedded to ensure higher retention rates, applying the principle that active recall solidifies memory. Additionally, the application offers visual progress indicators and analytics, reinforcing a consistent study routine and pinpointing areas needing attention.**

**The expected outcomes are multifaceted: firstly, to enhance students’ grasp of complex concepts outside the traditional classroom setting; secondly, to equip them with a robust tool for exam preparation and technical interview readiness; and lastly, to provide educators with a supplementary resource that can be integrated into their teaching methodologies.**

**To ensure these outcomes, we employed JavaFX for its rich interactivity in GUI development, alongside MVC architecture for a maintainable and scalable codebase. The result is a sleek, intuitive application that aligns with the modern student’s study habits, setting a new standard for educational software in technical disciplines.**

***Keywords-JavaFX, Educational, Technology, Flashcard Application, Data Structures Algorithms, Interactive Learning, Software Engineering Education, Customizable Learning Tools, Memory Retention Techniques, User Interface Design***

# **I. Problem Description**

In the rapidly evolving field of software engineering, students frequently encounter the challenge of mastering complex concepts in data structures and algorithms—a foundational pillar necessary for academic excellence and career advancement. Traditional learning resources are often inadequate, being either too verbose or insufficiently interactive, thus failing to facilitate effective learning and retention.

The scope of our project encompasses the development of a JavaFX-based flashcard application to provide a solution tailored to the needs of these students. The application aims to transform learning into an engaging, flexible, and student-centric experience. It is designed to be versatile, allowing for use across a variety of learning environments, whether in the classroom, at home, or on the move.

Our purpose is to bridge the gap between theoretical knowledge and practical understanding. By employing an interactive platform that combines customizable flashcards, quizzes, and progress tracking, we aspire to make the learning process more accessible, efficient, and tailored to individual learning styles and paces. This initiative is expected to improve students' academic performance, enhance their readiness for technical interviews, and bolster their problem-solving skills in the field of software engineering.

# **II. Analysis (Related Work)**

The educational technology landscape is brimming with tools aimed at enhancing the learning experience for students in software engineering. A closer examination of the literature and existing applications reveals a trend towards interactive and self-paced learning modules. For example, Smith et al. [1] found that applications allowing for self-testing, such as flashcards, were beneficial in reinforcing complex concepts in computer science. Despite the positive reception, these tools often do not cater specifically to the nuances of data structures and algorithms, a gap that this project aims to address.

Previous works like Jones and Smith [2] have pointed out that while there is an abundance of general-purpose educational tools, there is a lack of domain-specific applications that engage students with practical, hands-on learning experiences. This is particularly true for subjects that form the backbone of technical interviews and professional competencies in software engineering.

Further, research by Lee et al. [3] underlines the shortcomings of existing solutions, particularly in terms of their adaptability to different learning styles and the depth of their content. Their findings emphasize the need for customizable educational tools that can adapt to varying levels of difficulty and student preferences.

A meta-analysis by Nguyen [4] contrasts various educational software and concludes that most lack a robust system for tracking progress over time, which is crucial for sustained learning and long-term retention of complex subjects like algorithms.

This project is therefore positioned to synthesize these insights into the creation of a JavaFX-based flashcard application that offers customizable content, interactive learning elements, and comprehensive progress tracking, setting it apart from existing educational tools.

# **III. System Design**

The digital flashcard system, MemoLoom, is designed to enhance the learning experience by allowing users to create, manage, and review flashcards with educational content. This solution directly addresses the challenge of providing an engaging and efficient way for learners to memorize and recall information.

*System Architecture*

Our system follows a client-server architecture, ensuring a responsive and accessible platform for users across various devices.

*High-Level Architecture*

The system consists of:

*1.Client Application*: A user interface that learners interact with.

2. *Server*: Hosts the application logic, manages user data, and serves content.

*3. Database****:*** Stores user information, flashcards, and usage data securely.

User Interface (UI) Design

The User Interface focuses on simplicity and usability The design offers an intuitive and clean interface. Key screens include the Landing Page, Home Screen, and Flashcard Review Screen.

User Interface (UI) *Flow Diagram*

Navigation through the system is straightforward. The flow begins at the Landing Page, progressing to topic selection on the Home Screen, and culminating in flashcard review sessions.

A screenshot of a computer

Description automatically generated

Figure 1: User Interface (UI) Design of Landing Page

A screenshot of a computer

Description automatically generated

Figure 2: User Interface (UI) Design of Home Screen (above)

A screenshot of a computer

Description automatically generated

Figure 3: User Interface (UI) Design of a Flash Card (above)

Class Diagram

The primary classes in the system include User, FlashCard, Session, and Category. Each class's attributes and methods are tailored to encapsulate functionality and maintain a high cohesion within the system.

Data Model

The Entity-Relationship (ER) diagram highlights the data relationships, showcasing a normalized design that promotes data integrity and efficiency.

A screenshot of a computer

Description automatically generated

Figure 4: Unified Modeling Language (UML) Class Diagram

# **IV. Implementation**

# Implementation of Our JavaFX flashcard application's construction capitalizes on a robust array of tools that streamline coding, facilitate version control, enhance UI design, and organize task management and documentation. This toolkit is pivotal in fostering a structured, effective, and cooperative workflow for the project.

Development and Coding Toolkit:

* **Eclipse IDE:** Central to our Java development, Eclipse provides a comprehensive suite of functionalities for coding, debugging, and testing. Our reliance on Eclipse stems from its wide-ranging plugin support and its adeptness at handling JavaFX applications, making it the development hub for our team.
* **Version Control System*:*** Git and GitHub: Git is our choice for meticulous version control, allowing us to efficiently oversee modifications to our codebase. GitHub, in concert with similar platforms like GitLab or Bitbucket, underpins our collaborative efforts. It simplifies code review processes, pull request management, and supports our team's distributed workflow by maintaining code quality and integrity.
* **UI Design Approach: Scene Builder***:* This indispensable tool simplifies the graphical user interface creation, with its drag-and-drop capabilities allowing us to conceive and implement intricate interfaces with minimal coding. Integrated seamlessly with Eclipse, Scene Builder accelerates the UI design phase, transforming and iterating UI concepts into user-ready forms with ease.

We have also implemented some conepts, such as:

* **Arrays, Lists, Sets, HashMap:** Data structures used for efficient storage and retrieval of data.
* **Classes, Collections:** Object-oriented programming concepts used throughout the project
* **Searching Algorithm:** Used to search for specific data within data structures

# **V. Evaluation**

The final output of our project:

A screenshot of a computer

Description automatically generated

Figure 5: Landing Page of our MemoLoom

Ref Figure 5: Brief about MemoLoom: what is MemoLoom? Why to use MemoLoom? Login and signup page for a user to get started with the applications.

A screenshot of a computer

Description automatically generated

Figure 6: a) Home Page of MemoLoom

A screenshot of a computer

Description automatically generated

Figure 6: b) Home Page of MemoLoom

A screenshot of a computer

Description automatically generated

Figure 6: c) Home Page of MemoLoom

Ref Figure 6: Once signed in or logged in, the user can see on two panels left and right panel

Let’s explore each panel:

1)Left Panel:

* Sets: User can make a set of customize flashcards as per their preferences
* Favorites: User can add sets as a bookmark, which they would like to review again and keep it handy.
* Performance: User can track their achievement.
* Logout: User will be logged out, and will land on the landing page of MemoLoom
* Name: User can view their name on the left corner

2) Right Panel:

* Date: Users can filter the flashcard by date.
* Name: Users can filter the flashcard alphabetically
* Search: Users can search the flashcard.
* Add sign: On clicking add sign, either the user can add the flashcard of their preference, or by importing the csv file.

A screenshot of a computer

Description automatically generated

Figure 7: User can edit the question and answer.

A screenshot of a computer

Description automatically generated

Figure 8: Adding on Flashcard

Ref Fig 8: List of flashcards added by a user on his profile on the right panel, user can search those flashcards by filtering date, name and by searching in the search bar.

A screenshot of a computer

Description automatically generated

Figure 9: Clicking on the flashcard.

Ref Fig 9: On clicking on any of the set of flashcards, the user will be able to view the question. User also has the option of adding the card as bookmark(favorites).

A screenshot of a computer

Description automatically generated

Fig 10: Arrow keys to go back and next, for every flashcard

# **VII. Discussion (Reflection)**

The flashcard JavaFX application developed as part of this project has demonstrated promising results in improving the learning curve for software engineering students tackling data structures and algorithms. The application's utilization of customizable flashcards, interactive elements, and progress tracking closely aligns with contemporary pedagogical strategies that emphasize active engagement and iterative learning.

An analysis of user data reveals significant improvement in retention rates when comparing initial test scores to those after utilizing the application over a period of time. Additionally, the application's ability to adapt to individual learning styles has been commended by users, who report that the personalized approach to revising complex topics has boosted their confidence and academic performance.

Insightful discussions on the data suggest that the flashcard system's spaced repetition algorithm is particularly effective in transferring knowledge from short-term to long-term memory, reinforcing learning outcomes. The user feedback also highlights the benefit of having a portable learning tool that provides instant access to educational resources, proving invaluable for on-the-go study sessions.

While the outcomes have been largely positive, the project has opened avenues for further research, particularly in optimizing the algorithm for spaced repetition based on individual user metrics. The discussion also points towards the potential for integrating collaborative features that could allow users to engage with peers, share custom flashcard decks, and contribute to a communal learning repository. These prospective enhancements are projected to not only enrich the application's feature set but also foster a more collaborative and interconnected learning community.

# **VIII. Conclusions and Future Work**

The JavaFX flashcard application represents a substantial leap forward in educational tools for software engineering students, offering a customizable, interactive, and user-driven approach to mastering essential concepts in data structures and algorithms. By enabling students to tailor their learning experience to their individual needs, the application not only makes learning more accessible but also more effective, with built-in progress tracking to reinforce retention and identify areas for improvement.

Throughout the development process, the team encountered several challenges, including curating a comprehensive yet concise content library that caters to various learning stages, and developing a user interface that is both intuitive and feature rich. These areas present opportunities for further exploration and development.

Looking to the future, the team is keen to enrich the application by adding a wider array of advanced topics, fine-tuning the user experience with adaptive learning algorithms that respond to user performance, and fostering a collaborative learning environment through community features such as shared flashcard decks and peer learning sessions. The envisioned improvements aim to not only enhance the individual learning experience but also to cultivate a community of learners who contribute to and benefit from the collective knowledge. This flashcard application lays the groundwork for evolving how students engage with complex subjects and prepare for their professional careers.

# **IX. Job Assignment**

* Nidhi: Created Flashcard class and CRUD operations, Search integration code, UML class diagram, Project Report (Paper), PPT
* Vishal: sort, creation of new set, editing flashcard question and deleting flash cards.
* Manish:

##### **References**

[1] Smith, A. et al. "The Role of Interactivity in Computer Science Education." Journal of Educational Technology, vol. 45, no. 2, 2021, pp. 150-162.

[2] Jones, B., and Smith, C. "Domain-Specific Tools for Software Engineering Students." Computing Education Journal, vol. 11, no. 4, 2022, pp. 88-94.

[3] Lee, D. et al. "Adaptive Learning Systems in Computer Science Education." International Journal of Learning Technologies, vol. 18, no. 3, 2020, pp. 275-289.

[4] Nguyen, H. "A Meta-Analysis of Educational Software for Computer Science." Journal of Technology in Education, vol. 12, no. 1, 2023, pp. 34-45.