International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 10 | Oct – 2025 DOI: 10.55041/ISJEM05076
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

A Study and Implementation of Typing Speed Tester System

Mrs. R SuryaPrabha

Assistant Professor

Department Of Computer Science

suryaprabhar@skasc.ac.in

Sri Krishna Arts And Science College , Coimbatore

Rohithh D

rohithhd24bcs150@skasc.ac.in

Sri Krishna Arts And Science College , Coimbatore

Abstract

Typing speed and accuracy are vital skills in the modern digital age, where computers have become a core part of both education and professional environments. Many individuals struggle with slow typing speed and frequent typographical errors, which affect productivity and communication efficiency. The TypoBuster project aims to address these challenges by providing a software-based solution that evaluates typing performance and assists in improving both speed and accuracy.

TypoBuster offers an interactive interface where users can log in, perform typing tests, and receive feedback instantly. The system is developed using Java Swing for the frontend, ensuring a user-friendly graphical interface, and file handling for backend data storage, allowing performance data and user records to be securely saved. The software calculates key metrics such as words per minute (WPM), number of mistakes, and overall accuracy. By tracking progress over multiple sessions, it enables users to identify weaknesses and continuously enhance their performance.

TypoBuster offers an interactive interface where users can log in, perform typing tests, and receive feedback instantly. The system is developed using Java Swing for the frontend, ensuring a user-friendly graphical interface, and file handling for backend data storage, allowing performance data and user records to be securely saved. The software calculates key metrics such as words per minute (WPM), number of mistakes, and overall accuracy. By tracking progress over multiple sessions, it enables users to identify weaknesses and continuously enhance their performance.

The project follows a structured design and implementation approach, including system analysis, design diagrams, and multiple testing phases. The results demonstrate that TypoBuster effectively helps users enhance their typing speed and accuracy through practice and feedback. This paper discusses the methodology, development process, results, and future improvements for the system.

ISSN: 2583-6129
DOI: 10.55041/ISJEM05076

Keywords Typing Speed, Accuracy, Java Swing, File Handling, Performance Evaluation, TypoBuster, Typing Test System

Introduction

Typing has evolved from a professional skill to a daily necessity in the modern digital ecosystem. Whether it is in education, business, or technology, the ability to type efficiently determines productivity and the quality of work. Manual typing often results in typographical errors, slow performance, and lack of consistency, creating a need for tools that measure and enhance typing capabilities.

The TypoBuster system was developed to overcome these challenges by introducing a structured platform that allows users to test and track their typing abilities. It focuses on providing users with instant feedback, statistical data, and personalized learning experiences. By integrating real-time evaluation, the system promotes continuous learning and improvement.

The objective of TypoBuster is to enhance typing speed, improve accuracy, and build confidence among users. The software also provides an intuitive environment that motivates users to practice regularly.

Literature

Several online and offline tools exist to evaluate typing performance, such as TypingTest.com and Key Hero. However, most of these systems are web-dependent and offer limited offline access or progress tracking. Research shows that consistent practice combined with performance analytics significantly improves typing proficiency.

The TypoBuster system was developed to overcome these challenges by introducing a structured platform that allows users to test and track their typing abilities. It focuses on providing users with instant feedback, statistical data, and personalized learning experiences. By integrating real-time evaluation, the system promotes continuous learning and improvement.

The objective of TypoBuster is to enhance typing speed, improve accuracy, and build confidence among users. The software also provides an intuitive environment . In recent years, several research studies have emphasized the importance of developing intelligent systems that not only measure typing performance but also help in improving cognitive and motor coordination skills. real-time feedback, gamification, and adaptive difficulty adjustment to maintain user motivation.

System Architecture

The system is divided into two main components: the frontend and the backend. The frontend handles user interaction through the GUI, while the backend manages data processing and storage using file handling.

Functional Modules

Login and User Authentication:

Allows users to log in or register securely.

Typing Test Interface:

Displays text passages for users to type and records keystrokes.

Accuracy and Speed Calculation:

Calculates words per minute, accuracy percentage, and total errors.

Result and Performance Tracking:

Displays performance results and saves them for future comparison.

User Progress Storage:

Maintains a history of previous test results using local file storage.

Progress Tracking and Analytics Module:

This module records every test session's results and generates detailed analytics, including accuracy graphs, WPM trends, and error frequency charts.

Performance Feedback and Error Analysis Module:

After each test, this module provides detailed feedback — highlighting frequent mistakes, typing delays, or error-prone words.

System Maintainence

System maintenance is an essential phase of the TypoBuster project lifecycle that ensures the application continues to perform efficiently even after deployment. Once the system is delivered to users, various maintenance activities are carried out to enhance reliability, optimize performance, and adapt to any new user requirements or technological changes. Maintenance ensures that the TypoBuster software remains stable, secure, and compatible with evolving operating environments.

The main objective of maintenance is to keep the system running without interruptions and to fix any issues that may arise during its operational use. Regular maintenance schedules are followed to check for bugs, performance slowdowns, or errors in typing accuracy calculation. When users report unexpected behavior or display problems, corrective maintenance is performed to identify the root cause and resolve it immediately.

Another major type of maintenance is adaptive maintenance, where the TypoBuster system is updated to remain compatible with newer versions of Windows operating systems, Java libraries, or hardware configurations. This ensures that users can continue using the software seamlessly, even when their devices or platforms are upgraded.

Perfective maintenance is also implemented to improve the system's usability and efficiency based on user feedback. For example, adding new features such as difficulty level customization, sound alerts for errors, or enhanced graphical result charts are all part of perfective maintenance. These enhancements not only improve the system's performance but also increase user satisfaction.

Additionally, preventive maintenance involves monitoring the system regularly to prevent potential issues before they occur. This includes cleaning up unused files, optimizing database storage (if any), and checking for potential compatibility problems with newer Java or Swing libraries. Preventive actions help avoid future system crashes or performance degradation.

Proper documentation of all maintenance activities is maintained to ensure transparency and facilitate future upgrades. A detailed log records every update, patch, and modification made to the system, allowing developers to trace any issues back to their source efficiently.

The maintenance process also focuses on enhancing data integrity and security. Backup mechanisms can be introduced to ensure that user records and performance logs are safely stored and can be recovered if needed. Periodic testing after each maintenance update ensures that the system continues to perform accurately and reliably.

Overall, system maintenance ensures that TypoBuster remains a robust, user-friendly, and efficient typing speed testing application. Continuous monitoring, timely updates, and user-centered enhancements make the software sustainable for long-term use. With regular maintenance, TypoBuster can adapt to technological advancements and continue to serve as an effective learning and assessment tool for improving typing proficiency.

Input Design

Input design is the process of planning and structuring how data is entered into a system in order to ensure accuracy, efficiency, and ease of use. In TypoBuster, input design plays an important role because the entire system depends on the correctness and speed of the characters typed by the

user. The input is mainly in the form of text typed through the keyboard into a designated input area.

The typing area has been designed using Java Swing components such as JTextArea, which provides a clear and simple space for users to enter the displayed paragraph. The system validates each keystroke in real time, comparing it with the corresponding character in the reference text. Correct inputs are accepted and highlighted in blue, while incorrect inputs are flagged in red with an accompanying beep sound to alert the user.

The input design also considers ease of use for different types of users. Clear font styles and adequate spacing are provided in the typing area to reduce eye strain and improve readability. The interface ensures that users can focus on typing without unnecessary distractions.

During the typing session, the application continuously tracks the time elapsed and calculates performance metrics. Every second, the timer updates the display to show the current duration of the typing session and the user's typing speed in words per minute. The system calculates accuracy based on the number of correctly typed characters relative to the total typed characters.

After the test is completed or stopped, the results are displayed to the user in a message dialog, and the performance data, including the date, WPM, and accuracy, is saved to a file for future reference. This input design ensures that every keystroke contributes to performance evaluation, providing accurate and reliable feedback to the user.

The typing area also intelligently distinguishes between regular text mode and program mode, suppressing unnecessary notifications like error beeps during coding exercises to enhance user focus. All user inputs, including keystrokes, difficulty selections, and program choices, are carefully validated and managed to prevent application crashes or incorrect data recording. The application accepts user inputs primarily through login credentials, typing test text entries, and navigation commands. The login module requires a username and password to authenticate users. The typing test module takes user keystrokes as input and records them in real-time to measure speed, accuracy, and error patterns. These inputs are validated to avoid errors such as empty fields, invalid usernames, or incorrect password attempts.

Output Design

Output design is an important stage in system development that focuses on how results are displayed to the user in a clear, accurate, and meaningful way. In TypoBuster, the output is designed to provide immediate feedback during typing as well as a summary of performance once the test is completed. During typing, the system outputs real-time visual feedback by displaying correct characters in blue and incorrect ones in red.

Additionally, a beep sound is generated for mistakes, ensuring that the user is instantly aware of errors. This interactive output helps users correct themselves quickly and continue typing without confusion. At the end of the typing session, the system generates a detailed result summary.

The output includes total characters typed, number of correct and incorrect characters, accuracy percentage, words per minute, and total time taken. This structured presentation of results allows users to clearly understand their typing performance.

The output is displayed in a well-organized format using labels and message panels in Java Swing, ensuring that users can easily read and interpret the results. The interface is kept simple and uncluttered so that the focus remains on the performance details rather than unnecessary visual elements.

Output design focuses on how the system presents information to the user in a clear, meaningful, and actionable manner. In TypoBuster, effective output design ensures that users can easily interpret typing results, performance history, and program mode feedback. Proper output design improves user satisfaction and supports learning goals.

Conclusion

The TypoBuster project was successfully designed and implemented as an interactive typing practice system to improve user accuracy and speed. The system provides a user-friendly interface developed in Java Swing, offering modules such as login, signup, typing test, result calculation, and history tracking. Through real-time error highlighting, the application helps users identify mistakes instantly, thereby creating an engaging and effective learning environment.

One of the key achievements of TypoBuster is its ability to calculate typing speed in words per minute (WPM) and accuracy percentage, allowing users to monitor their progress over time. The system also records past performance and displays it in a structured format, encouraging consistent practice and improvement.

The project addressed the limitations of traditional typing practice methods by introducing a digital and automated solution. Unlike manual practice, where users cannot measure accuracy easily, TypoBuster provides instant evaluation and feedback, which motivates learners to refine their skills.

The system was implemented on the Windows operating system and tested thoroughly for functionality, usability, and stability. Testing demonstrated that the application is reliable and meets the requirements set during the design phase. It has shown to be effective for students, professionals, and anyone aiming to enhance their typing efficiency. Although the current version meets the core objectives, there is scope for future enhancements such as customizable difficulty levels, support for multiple languages, advanced progress analytics, and online competition modes.

TypoBuster stands as a valuable tool that bridges the gap between traditional typing practice and modern automated evaluation. It not only improves speed and accuracy but also provides motivation through progress tracking. The system is simple, efficient, and capable of serving as a practical solution for individuals who wish to develop strong typing skills.

The development of TypoBuster highlights the importance of integrating technology with practical skill enhancement. By combining features such as user authentication, multiple difficulty levels, program-based practice, and detailed performance tracking, the system goes beyond traditional typing testers. It not only measures speed and accuracy but also provides meaningful feedback that motivates users to improve.

References

8.1 Book References

1.Herbert Schildt, *Java: The Complete Reference*, 12th Edition, McGraw Hill, 2022.

2. Cay S. Horstmann, *Core Java Volume I – Fundamentals*, 12th Edition, Pearson Education, 2021.

3. Kathy Sierra & Bert Bates, *Head First Java*, 2nd Edition, O'Reilly Media, 2005.

4. Patrick Niemeyer & Daniel Leuck, *Learning Java*, 5th Edition, O'Reilly Media, 2020.

5. John Lewis & Joseph Chase, *Java Software Solutions: Foundations of Program Design*, 9th Edition, Pearson, 2017.

8.2 Web References

1. Oracle Java Documentation – https://docs.oracle.com/javase/

2, Java Swing Tutorial – https://www.javatpoint.com/java-swing

3. GeeksforGeeks, Java Programming – https://www.geeksforgreeks.org/java/

4. TutorialsPoint, Java Swing- https://www.tutorialspoint.com/swing/index.html/

5. W3Schools, Java Programming Basics – https://www.w3schools.com/java/