Digital Technologies Research and Applications
http://ojs.ukscip.com/index.php/dtra

Article

Development of a Novel Software Interface for Tracking Student Behavior, Performance and Attendance

Charlie Todd and **Emanuele Lindo Secco** *

School of Computer Science and the Environment, Liverpool Hope University, Hope Park, Liverpool L16 9JD, UK
* Correspondence: seccoe@hope.ac.uk

Received: 2 September 2025; **Revised:** 18 October 2025; **Accepted:** 22 October 2025; **Published:** 25 November 2025

Abstract: This paper presents a novel software interface for tracking student behavior, performance and attendance, exploring how software is used in education and analyzing the benefits and challenges that can come with its use. Technology has become increasingly important to education over recent years, making it a major focus within both technology and education. In this context, the paper presents the implementation of a website, designed to convey how technology can streamline access to information, making it faster and more efficient than the usual paper-based methods used to track attendance, behavior and grades of students. Leveraging on the Technology Acceptance Model (TAM) and the Constructivist Learning Theory (CLT), a Client-Application Server and Database have been integrated into a novel framework where different users (i.e., student, parent, teacher, admin) can query the database. Role-based access is controlled through users' tables which contain the user ID, name, email, and password. The relationships between the tables are enforced using foreign keys to maintain data integrity. The proposed system's design prioritizes accessibility, enhancing the learning capability with a user-friendly interface. The system's performance has been successfully tested in terms of time response, speed, efficiency and clarity. Initial single-user testing showed a load time of 18 ms for the student login page, with further validation needed for multi-user scenarios.

Keywords: Education Software; Education Management System; Virtual Learning Environment (VLE); User Interface (UI); Graphical User Interface (GUI); Human-Centered Design (HCD); Software Prototype

1. Introduction

The objective of this study is to explore student tracking software. Specifically exploring why they are used, the benefits and drawbacks of their usage, and then developing a suitable alternative that can cater to institutions that currently are unable to access such software. Technology has transformed education, particularly in tracking academic performance and monitoring attendance. Software has become increasingly popular in education settings worldwide, with the aim of improving efficiency and providing real-time information to everyone involved, the teachers, parents, students and administrators, with a number of education software, Virtual learning environments and user interfaces. These software systems aim to enhance student engagement and look into any potential trends in attendance and behavior, to potentially help identify and address issues quickly. By shifting from paper-based methods to software applications, administrative tasks, such as registration, have become more efficient.

Despite these benefits, challenges do arise, such as cybersecurity risks, user training, and data privacy. With paper-based systems, the information on who was in a class would not leave the building. Now, with software, the information is stored in the cloud, making it susceptible to hacking, which can lead to the exposure of sensitive information regarding students or staff. In this context, we want to look at how software has been developed to monitor student records. Such an exploration will be literature-based and combined with the proposal of a software prototype as a case study.

Technology has become increasingly integrated in education systems over the past couple of decades, with many institutions adopting software to monitor student behavior, attendance and performance, moving away from paper-based systems with the aim of improving efficiency. Paper-based methods need manual entry and are time-consuming. By comparison, using technology, teachers can send and collect data on a student's attendance, behavior and performance [1]. This change reduces the time taken for administrative issues, allowing educators to focus more on teaching, potentially leading to better student outcomes.

Student tracking software mainly focuses on attendance, behavior, and academic performance, though it can serve additional purposes [1]. The increase in usage of technology has significantly altered administration, meaning registering students' and teachers' attendance and recording behavior is automated and therefore less time-consuming. However, using software introduces new challenges, such as data privacy concerns and technical difficulties. Schools store sensitive data, meaning digital safety must be a priority to prevent unauthorized access. Cybersecurity breaches can damage a school's reputation with parents and students. Therefore, it is important to protect information so there is trust between schools and parents and students [2].

The education system has become increasingly reliant on student tracking software to keep track of students in the previously addressed areas. However, not all institutions can afford such software. For example, larger software packages may be more expensive and not completely worth it for smaller institutions. Moreover, there is an urgent need for innovation and technological updates of these software applications. Accordingly, there is a need for software alternatives such as the system we are going to propose.

1.1. Evolution of Student Tracking Software

Traditionally, student behavior, attendance and performance were tracked through paper records. While this method was effective, it was time-consuming, prone to human error, and lacked viable alternatives to technology. Early student tracking software was limited, focusing on singular aspects, like grades or attendance. However, modern systems provide more comprehensive data. One example of this is Pupil Progress. They state that their software was created to remove heavy lifting for teachers by automating calculations, providing detailed insights and streamlining data management [3].

A major advancement in student tracking is the transition from paper-based records to cloud storage, making data more accessible for teachers, parents, students and administration. If they have internet access and the correct credentials, they are able to access the data remotely [4]. Additionally, cloud-based systems are scalable, which means new features can be added by institutions without requiring major software alterations.

1.2. Benefits of Student Tracking Software

Student tracking software has several benefits. One of the main advantages is that data is collected, stored and accessed far more efficiently compared to when the method was paper-based. Automation reduces the burden on admin staff and teachers, allowing them to focus on their main responsibilities more effectively [4,5]. This allows educators to quickly identify negative trends in attendance, behavior or performance, facilitating quick support.

Another benefit is long-term data tracking, so student progress can be tracked over the course of a school year, or even their entire time at the school. By doing this, institutions are able to identify potential patterns from a student in different areas, which can then enable personalized learning plans fitting individual students' requirements [3,6].

Additionally, tracking software helps to develop stronger relationships between teachers, parents and students by providing real-time updates on attendance, behavior and performance. This means that parents would also be able to intervene, when necessary, working alongside the teachers to support their child's education [7].

1.3. Challenges of Student Tracking Software

While student tracking software offers many benefits, it also presents challenges. One major concern is data privacy. Cloud-based systems increase the risk of data breaches, exposing sensitive information such as names, addresses, contact details and medical records [8]. In this context, statistics convey that schools are an active target for hackers, threatening the data of individuals, as well as the trust between institutions, staff, parents and students. Complying with GDPR is vital for safeguarding sensitive data.

Beyond privacy issues, there are also logistical issues. Staff must be trained to use new software, but adapting to it can be difficult, considering that teachers must also focus on educating students. As such, they may resist using these tools, thus limiting effectiveness [9]. Without proper training, the potential of the software would be underutilized.

Cost is another barrier, especially for smaller institutions with limited budgets. Expenses include hardware, software, training of staff, system maintenance and technical support. Some institutions may conclude that the financial burden would outweigh the benefits [10].

1.4. Impact on Student Performance and Engagement

Student Tracking software is able to positively impact student engagement, as teachers can quickly access records, which allows for early intervention if a student's attendance drops. It also creates healthy competition, as some schools use this software to lighten the tone on attendance. Rewarding those who have outstanding records of attendance can encourage those who do not attend more frequently, so they too can get rewarded [11]. Some institutions push attendance too rigidly, which can stress students and result in them transferring to other schools.

The software can also be used to identify students who struggle with discipline, allowing for immediate interventions to reduce the necessity for harsher punishments, such as suspension [3,12]. By analyzing the data, teachers can inform parents if they have cause for concern, so they can collaborate to prevent the issues from escalating.

Despite the benefits, challenges arise. If the software is not effectively used, it is redundant. There are also concerns regarding discrimination [13]. Ensuring fair use of these tools is vital, as improper use can turn a potential benefit into a disadvantage.

1.5. Key Theories

There are several key theories regarding the use of technology in education [5]. In the context of education, Davis proposed what is called the Technology Acceptance Model (or TAM) in 1989, where it basically expressed the concept that technology can succeed towards the potential end-users, whether it is perceived as useful and, at the same time, easy to use [14]. In other words, TAM means that teachers, administration, parents and students must find the tool beneficial and user-friendly. If not, its potential is reduced. Another key theory is Constructivist Learning Theory (CLT) [15]. Student tracking software provides personalized learning by providing teachers with relevant information, so they can identify individual needs and tailor support accordingly. This provides students with motivation, as they will feel understood and supported.

While extensive research exists on the use of technology, there remain gaps in understanding what the impact is on attendance, behavior and performance. Much of the literature focuses on functionality and ease of use. The analysis on whether or not students' records are being impacted by the software is sparse. Additionally, the long-term effectiveness of the tracking software - whether the usefulness of the software increases, stagnates, or fades – requires further study.

More research is needed on how institutions can address challenges, such as cost and user training. The paper aims to address these gaps in the research by analyzing both the benefits and the challenges of the software and evaluating its effectiveness in improving student outcomes.

The aim of this paper is to explore how software has been used in education to track attendance, behavior and performance, looking at how effective software has been in education and analyzing benefits and challenges that come with its use. The objectives of the research are:

- To explore how technology has evolved in its use to track students' attendance, behavior and performance.
- To identify and analyze the benefits and challenges of using software in institutions to track student records.
- To examine the use of software to try and identify whether there have been improvements from students in any of the key areas.
- To develop a website that shows an example of how technology is used for efficient tracking and management of students' records.

There are several research questions that must also be asked, which are:

- How has software evolved and improved to become able to track student attendance, behavior and performance?
- What are the key benefits and challenges of using software to track students' records?
- Has the use of software to track student records led to any improvement in students' records?
- How can software be improved further to make the system easier and more efficient?

These questions link to the research objectives, which reinforce the paper's key focus.

The paper is organized into 4 more Sections: Section 2 reports the design, Section 3 focuses on the data collection and preparation, Section 4 reports the Results, and it is followed by a Discussion and Conclusion (Section 5).

2. Design

The proposed design and methodology leverage a combination of primary and secondary research. The research focuses on the implementation of a student tracking system, as opposed to conducting surveys or interviews. This means that the research focuses on the system design and development of an improved student tracking software that is more efficient than current models.

This approach aims to align with software engineering methodology, focusing on development, testing and refining or altering the system to ensure usability. Secondary research involves analyzing existing literature on the topic and integrating theories such as the Technology Acceptance Model (TAM) and Constructivist Learning Theory (CLT) to enhance system design, ensuring ease of use and suitability. Using a dual-method strategy helps to enable a comprehensive evaluation of the software's effectiveness in education, bridging the gap between theoretical models, and practicality.

2.1. Requirements and Specifications

The Software Development Life Cycle (SDLC) is incorporated in this research. "The goal of the SDLC life cycle model is to deliver high-quality, maintainable software that meets the user's requirements" [16].

The process follows 5 phases, namely:

1. Requirement Analysis – Identify key functions that are needed for a student tracking software based on prior literature, ensuring a well-defined set of requirements.
2. System Design – Develops a blueprint, including data structure and user interface designs.
3. Implementation and Development – Builds the software using appropriate computer languages and tools, with continuous testing to prevent errors.
4. Testing and Evaluation – Assesses functionality and usability. Debugging and user experience testing were done to assess both the robustness and ease of use.
5. Deployment and Maintenance – Prepares the system for real-world use.

This approach ensures that the system is human-centered and user-friendly, addressing gaps in existing student tracking software.

2.2. Data Collection

Data collection involves performance analysis and secondary resources. The methods to evaluate the system for this paper include:

- Literature Review Data - Guides system functionalities and features based on existing research in student tracking software and education technology. Peer-reviewed articles, industry reports, and case studies provide insights into best practices, challenges, and areas for improvement.
- System Performance Testing - Functional and performance tests evaluate efficiency, data security and usability. The evaluation is focused on accuracy of data storage, processing speed and reliability.

- Comparative Analysis with Existing Systems - Reviews features, usability and limitations to determine whether the new software is an improvement.

To ensure fair comparisons, criteria were developed, considering usability, data protection and interface design.

2.3. Data Analysis Techniques

Data from system performance testing and comparative analysis is evaluated using quantitative and qualitative analysis.

Quantitative analysis measures performance metrics, such as data accuracy and storage efficiency, to assess system effectiveness.

Qualitative analysis involves observations made during development and testing to evaluate usability and ensure alignment with educational goals. Supervisor feedback, recorded through consultations, helped to make informed decisions throughout the evolution of the software. Additionally, self-reflection and developer insights helped refine the implementation to ensure it was suitable for educational needs and met the required objectives.

2.4. Ethical Considerations

Although there is no direct human participation involved, ethical considerations remain vital. The system is designed to collect and store sensitive student information; therefore, strict privacy measures are implemented to ensure compliance with the General Data Protection Regulation (GDPR) guidelines and prevent unauthorized access. Academic integrity is upheld, as all secondary sources are cited, referenced and acknowledged properly. For example, all information is stored in phpMyAdmin; however, password hashing is used to prevent unauthorized access to accounts. Also, there is role-based access on the software. When signing up, each user type has its own sign-up page. Each one requires different criteria, preventing a student from being able to sign up as a teacher and gain access to their timetable. Given the reliance on secondary data, the research avoids bias, misrepresentation and overstatement of findings. Also, potential ethical concerns surrounding surveillance and data ownership were considered during system design to ensure data security.

As stated previously, software like this can be used unfairly, whereby some students might receive more disciplinary action due to discrimination [13]. The software built for this study is designed to be fair to all students, as there is equity in monitoring, meaning that no group of students or staff is scrutinized. The software is built without any form of profiling built in, ensuring that the software is as fair as possible.

2.5. Implementation Details

This section aims to provide a deeper understanding of how the implementation was designed. It includes a system architecture diagram showing how the client, application server and database are connected. It also includes a technical stack, showing what technologies were required to build the implementation.

Focusing on the technical stack, the frontend is developed using HTML5, CSS3 and JavaScript. The backend is developed using PHP 8.2.0 which runs on an Apache 2.x web server within UniServerZ. PHP handles authentication, the Role-Based Access Control (RBAC). The implementation forces boundaries based on the user type (i.e., student, parent, teacher, admin) using structured request handling to ensure data security. UniServerZ is a lightweight stack that is ideal for localized testing. The implementation uses MySQL as the database management system, which is accessed via phpMyAdmin within UniServerZ. The database schema is designed to support the core management functions for the school. There are tables for each of the user types. For example, the students' table contains their name and student ID and the institution that they are in. There is also a points table which shows how many achievements and behavior points a student receives in a lesson, with a description, with the table also uses the student ID. The MySQL transactions ensure atomic updates to points during concurrent access. Role-based access is controlled through a table called users. The users table contains the user ID, their names, emails, and passwords (which are hashed to protect them). There is also a user type section in the table which links the user to their relevant table. The relationships between the tables are enforced using foreign keys to maintain data integrity. **Figures 1** and **2** show how the Client, Application Server and Database interact and the Entity Relationship Diagram.

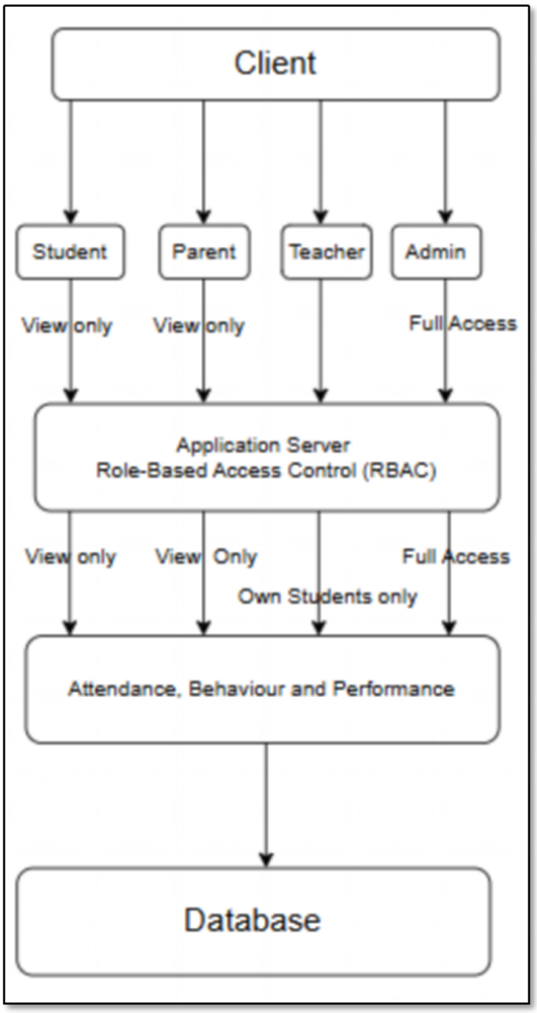


Figure 1. A visual representation showing how the Client, Application Server and Database interact.

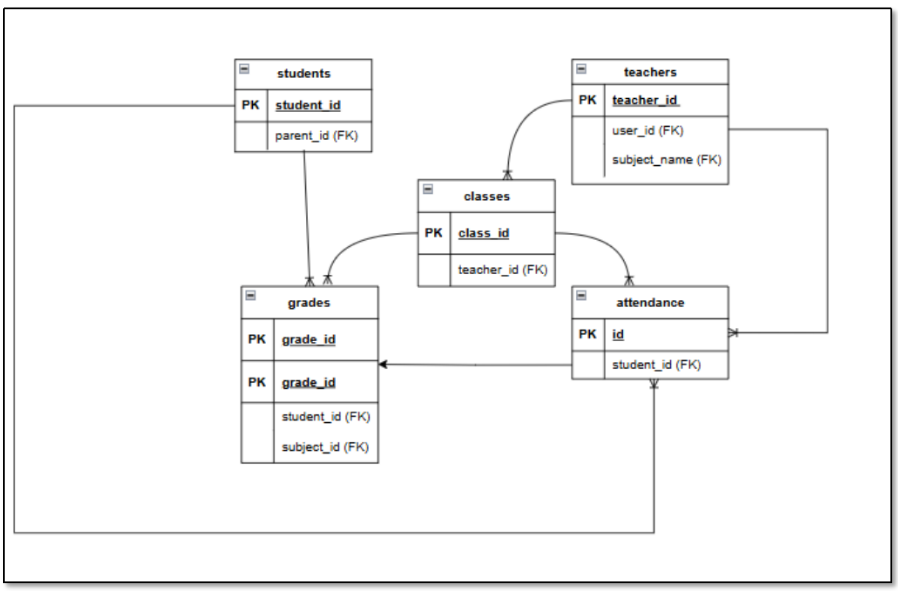


Figure 2. Entity Relationship Diagram.

2.6. Limitations of Methodology

While the methodology has strengths, it also has limitations. Lack of real-world implementation means the system is tested in a controlled environment, without being adopted in an actual school. Limited direct user feedback arises as the study does not include surveys or interviews with teachers or students, restricting insights into user experience. Comparative analysis constraints exist, as the assessment of existing software is based on available documentation and reviews, as opposed to direct interaction with the software.

Despite the limitations, the methodology provides a robust approach to designing, developing and testing student tracking software, enhancing other variants of the software, through theoretical insights and system performance analysis.

3. Data Analysis

This section presents the data collected from the development and evaluation of the student tracking system. The data is collected through testing, analysis and qualitative information and feedback. Quantitative and qualitative data are analyzed to evaluate the system's efficiency, reliability and usability.

The section includes Microsoft Edge DevTools screenshots, performance statistics, showing system speed with different user loads, and a comparison table, highlighting key differences between the current systems and the one that has been designed, and screenshots of the prototype. **Figure 3** conveys how user-centered and friendly the software is, with a clear home page with clear indications of how to log in or sign up.

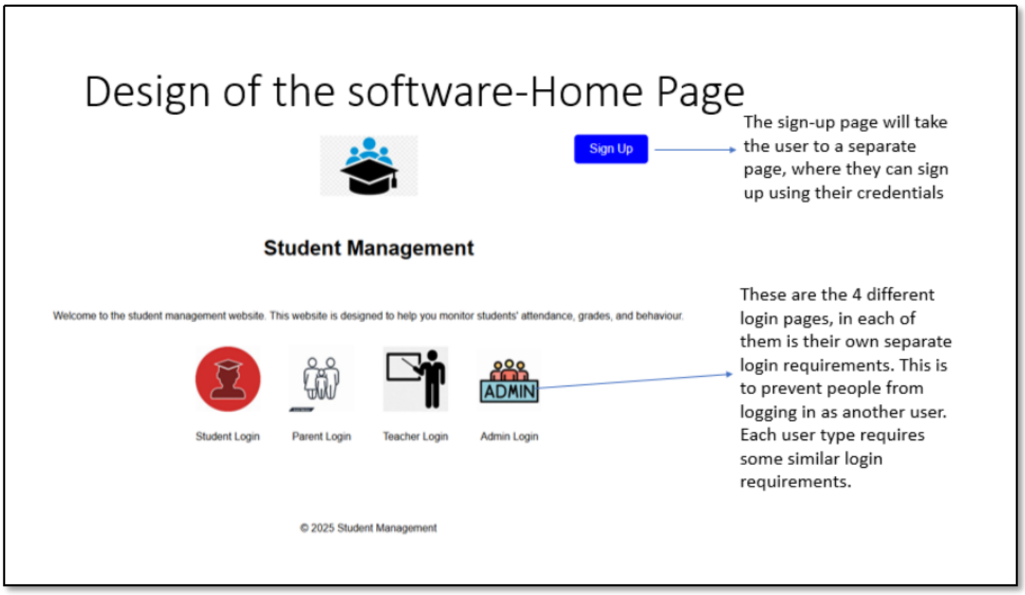


Figure 3. Design of the proposed system's Home Page.

3.1. System Performance Testing

Performance testing was conducted using Microsoft Edge's Developer Tools to analyze system efficiency, according to the following criteria:

- Page Load Time—Logging in as a student, as shown in **Figure 4**, the page (student_login.php) loads in 18ms, with DOMContentLoaded at 48 ms and full load event at 49 ms. This indicates a highly optimized system, suitable for education, where speed and reliability are essential. Faster software minimizes the time wasted on administrative tasks, allowing more focus on lessons.
- Data Handling Efficiency—As shown in **Figure 2**, the student_login.php page transfers 1.3 Kb of data, contributing to quick response times, which is beneficial for schools with limited internet capacity. The system uses 3.1 Kb out of a total potential of 20.3 Kb, ensuring efficient performance.

- Accessibility—Effective data transfer enhances scalability, enabling smooth handling of increased users, while reducing server strains in multi-user environments. Moreover, the system's minimal data transfer requirements improve energy efficiency, allowing usability on older systems, making it much more accessible than other options.

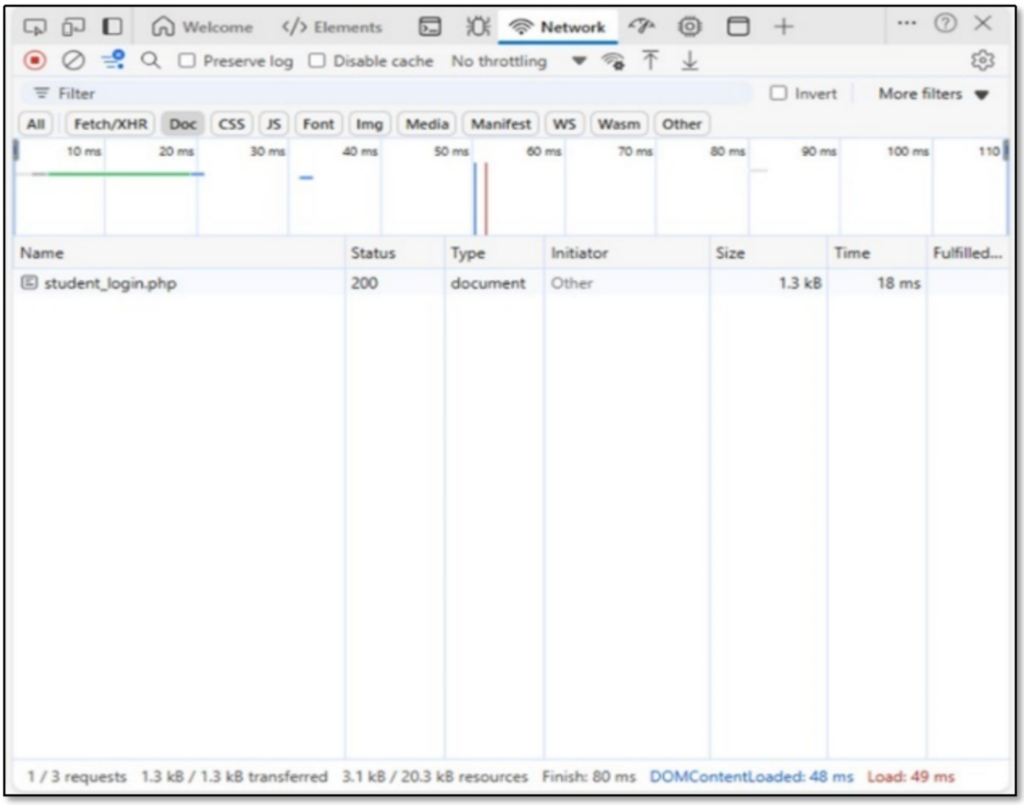
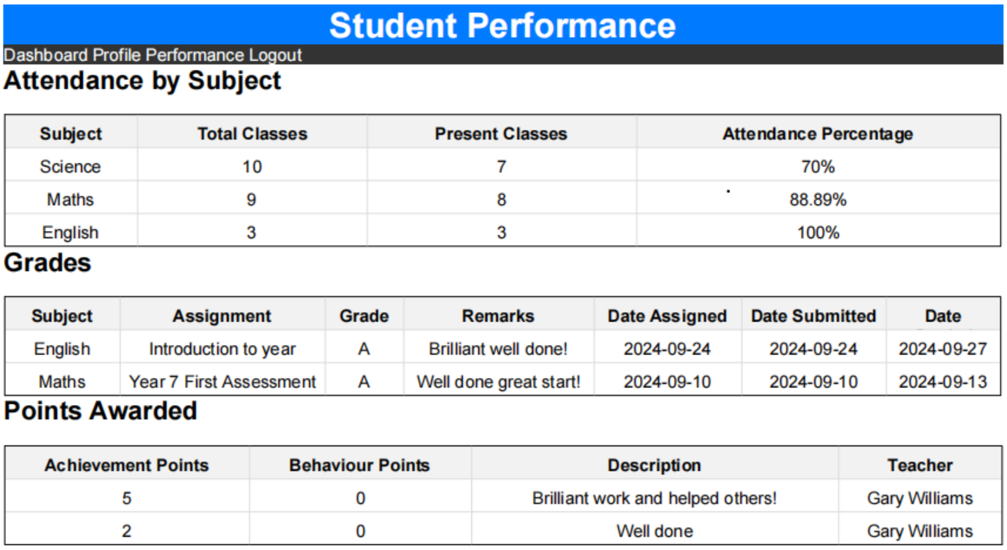


Figure 4. Screenshot of student_login.php loading speed at 18 ms, size at 1.3 kb, DOMContentLoaded speed being 48 ms, and total load time being 49 ms.

To test the student tracking software, simulations were conducted to estimate system performance under various user loads. The system was tested with a single user, where the system loaded in 18 ms, with 1.3 Kb of data transferred (shown in **Figure 4**). Using these values, potential user loads were simulated to evaluate responsiveness; however proper multi-user load testing was not performed at this stage. Whilst testing has been considered, real-world validation is required to confirm performance in actual scenarios.

3.2. Usability Evaluation

During the development, the system was frequently refined to enhance usability, clarity and responsiveness. There was a particular focus on navigation and speed, ensuring accessibility for all users, which is especially important in education settings, as not everyone will be familiar with complex software. Final testing showed no console errors, which confirms stable performance in the code, including PHP and JavaScript. For ease of use and accessibility, the approach remained exclusively software-based, as opposed to incorporating additional hardware. The User Interface (UI) design should be simple and effective, minimizing clutter and maintaining user-friendly navigation [17–20]. These priorities help reduce the amount of training required and improve system effectiveness, making it suitable for education settings. **Figure 5** demonstrates one example of the usability of the software, as the student performance page is easy to navigate and information is legible.



Student Performance

Dashboard Profile Performance Logout

Attendance by Subject

Subject	Total Classes	Present Classes	Attendance Percentage
Science	10	7	70%
Maths	9	8	88.89%
English	3	3	100%

Grades

Subject	Assignment	Grade	Remarks	Date Assigned	Date Submitted	Date
English	Introduction to year	A	Brilliant well done!	2024-09-24	2024-09-24	2024-09-27
Maths	Year 7 First Assessment	A	Well done great start!	2024-09-10	2024-09-10	2024-09-13

Points Awarded

Achievement Points	Behaviour Points	Description	Teacher
5	0	Brilliant work and helped others!	Gary Williams
2	0	Well done	Gary Williams

Figure 5. Student's Performance page reporting all the information and a set of functions on the top left corner of the panel (i.e., navigation tools such as the dashboard, the profile, the performance mask and the logout button).

3.3. Comparative Analysis

The system can be compared to other student tracking systems, such as SIMS, which is widely used in schools. **Table 1** below compares existing systems with the system designed for this paper.

Table 1. Comparison between currently available software and the proposed software.

Feature	Current Software Capability	New Software
Access Speeds	Moderate	Fast
UI simplicity	Moderate/Complex	Very Simple
Data Entry Process	Complex, Several Steps	Simple, easy process
GDPR	Compliant	Compliant
Cost	Expensive	Cheap
Scalability	Suitable	Not yet suitable

The designed system shows improvements, particularly in speed and UI simplicity, making it more efficient. However, it is currently less flexible than other alternatives (**Table 1**).

Performance data confirms the system's efficiency, and ability to handle a lot more data without compromising functionality. Usability testing and comparative analysis support its viability as a suitable alternative to existing student tracking software. While particularly useful in smaller education settings, the system can be scaled for larger environments. Further real-world and multi-user testing will help assess its effectiveness and validate previous simulated results.

4. Results

This section interprets the findings shown in the previous section, analysing their significance in relation to the research questions (Section 1) and theoretical considerations (Section 2). It examines how the developed student tracking software addresses challenges in education, such as software efficiency. Findings will also be examined in relation to current literature, technological theories such as TAM and CLT, and the broader implications of using this and similar systems in real-world education settings [21–23].

4.1. Addressing the Research Questions

How has software evolved and improved to become able to track student attendance, behaviour and performance?

Student tracking systems have advanced due to web technologies and the increasing demand for detailed student records. The system designed for this paper represents a modern, lightweight approach, prioritizing speed and efficiency over excessive clutter that could hinder usability. An example of how a student profile appears on the website can be seen in **Figure 6.** Unlike older desktop-only systems, modern systems, like the one developed here, allow for real-time access from different devices, improving accessibility. This evolution is due to improved browser technology.

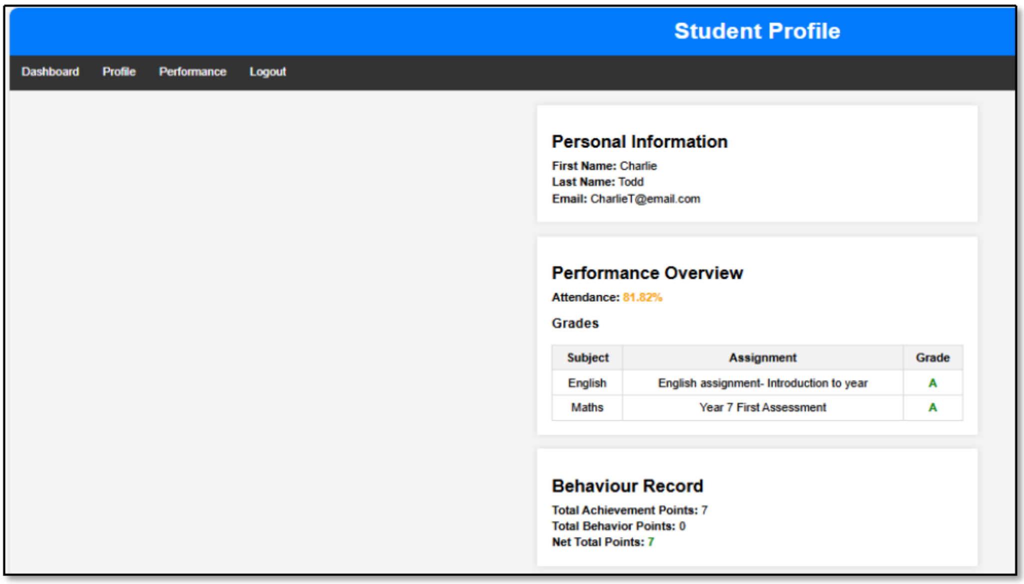


Figure 6. Student Profile's page.

What are the key benefits and challenges of using software to track students' records? There are several benefits based on the findings, including:

- Efficiency—Automates attendance tracking, saving teachers time.
- Speed—The 18 ms load time for the student_login.php page shows responsiveness.
- Clarity—The UI simplifies complex tasks.

However, several challenges have emerged from the findings, including:

- Scope of Features—Balancing additional functionality without affecting usability remains a challenge.
- User Training—Basic UI minimizes training needs, but users with minimal experience may still need support.

Has the use of software to track student records led to any improvement in students' records?

Whilst the research does not directly track student records, existing literature suggests that having quick, accessible data improves student outcomes. The system's design encourages frequent updates, which supports early identification of issues and therefore faster intervention. As stated in the literature review, real-time awareness of behavior or attendance issues allows teachers and parents to intervene promptly, reducing the risk of the situation escalating. If the software is used effectively, there could be better engagement by students and consequently better performance.

How can software be improved further to make the system easier and more efficient? Further improvements should address key limitations, such as:

- Customization: Schools could benefit from having the ability to modify layouts when needed.
- Mobile Optimization: Based on current estimations, this is feasible, but testing the system on mobile devices is still required.

Third-Party Integration: Adding email system support could streamline processes, such as contact between teachers and parents.

4.2. Linking Findings to Theoretical Frameworks

Here the focus is mainly on the TAM and CLT approach, namely:

- TAM—Users adopt new technology when it is useful and easy to use. Automation of attendance tracking and grade monitoring enhances efficiency, making it a valuable tool. The system has a simple layout, fast performance, and simple navigation, reinforcing its ease of use and aligning with TAM theory.
- CLT—This theory puts emphasis on simplicity, ensuring that the UI contains only essential components, reducing clutter on the page so that the system is understandable regardless of prior experience. The system also aligns with this theory, as its streamlined design prevents confusion for novice users, while maintaining efficiency.

Figure 7 shows the implementation of these theories, ensuring there is a clear focus on simplicity, efficiency, and effectiveness.

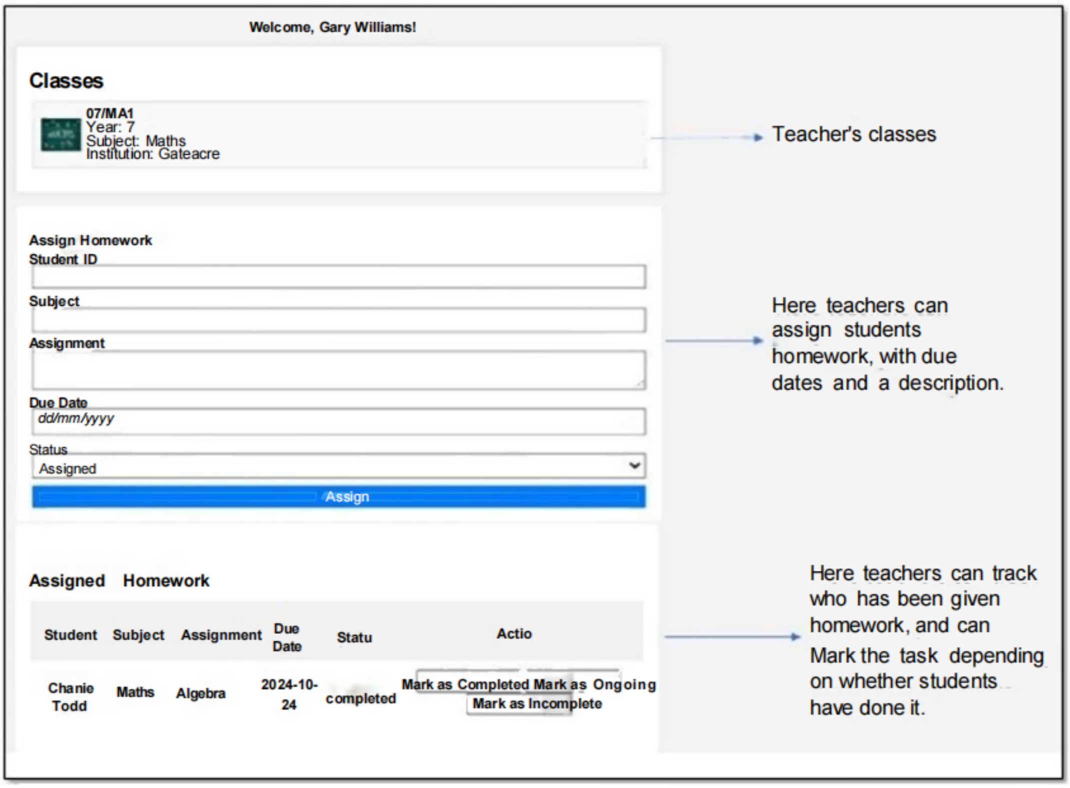


Figure 7. The main teacher's mask integrates a user-friendly design: the teacher's dashboard displays the class and students' data.

4.3. Results' Implications

This section examines how the findings impact education and technology sectors, identifying beneficiaries and other key observations from the perspective of Schools, Developers and Policymakers:

Schools—The system is a faster, more responsive option compared to larger software like SIMS. Smaller capacity schools will benefit the most from the system, although the software has the potential to be scalable.

Developers and Designers—Findings highlight the need to prioritize speed and responsiveness over excessive data, ensuring accessible designs.

Policymakers—Supports the case for investing in accessible technology, showing that even a simplistic system can have significant benefits.

The literature review highlighted the complexity of systems like SIMS and supported the idea that simple, user-friendly designs are more effective and inclusive. Over time, student tracking systems have become increasingly complex, with companies adding new functionalities without fully considering usability. This system addresses this by emphasising clarity and simplicity, while still benefiting schools. The findings reinforce this, even suggesting that simplicity can enhance engagement and encourage user adoption.

Despite the strengths of the software developed, several limitations must be addressed:

- Single-user testing: Only tested with one user; further tests required before school-wide implementation.
- Simulation over real-world testing: The simulation predicted what may happen, but actual school testing is needed for confirmation.
- Lack of features: Compared to larger systems like SIMS, it offers fewer functionalities, which may reduce its appeal to larger schools

Despite these limitations, the system effectively meets its intended goals, demonstrating a feasible student tracking system that has the potential to improve attendance, behavior and performance tracking for education settings. Future improvements include real-world testing, implementing notifications for updates, such as new homework, and optimizing mobile accessibility to further enhance usability. These improvements would transform the system from a functional prototype into a fully operational educational tool.

Finally, these results confirm that the system aligns with its core aims, which were usability, efficiency and accessibility. It addresses key research questions that were identified in Section 1 by demonstrating how student tracking systems can evolve to meet modern demands, while preserving speed and efficiency. The project integrates TAM and CLT theories, reinforcing literature that emphasizes inclusive software design, to remove barriers for those with less technological experience. Analytics and results show that the system is a viable alternative to larger, more complex software. Despite its limitations, the system provides a strong foundation that can be refined. The research supports the idea that thoughtful design can create powerful, effective and user-friendly student tracking software for real-world educational use. **Figure 8** provides another example of the website meeting its original aims, with the student dashboard.

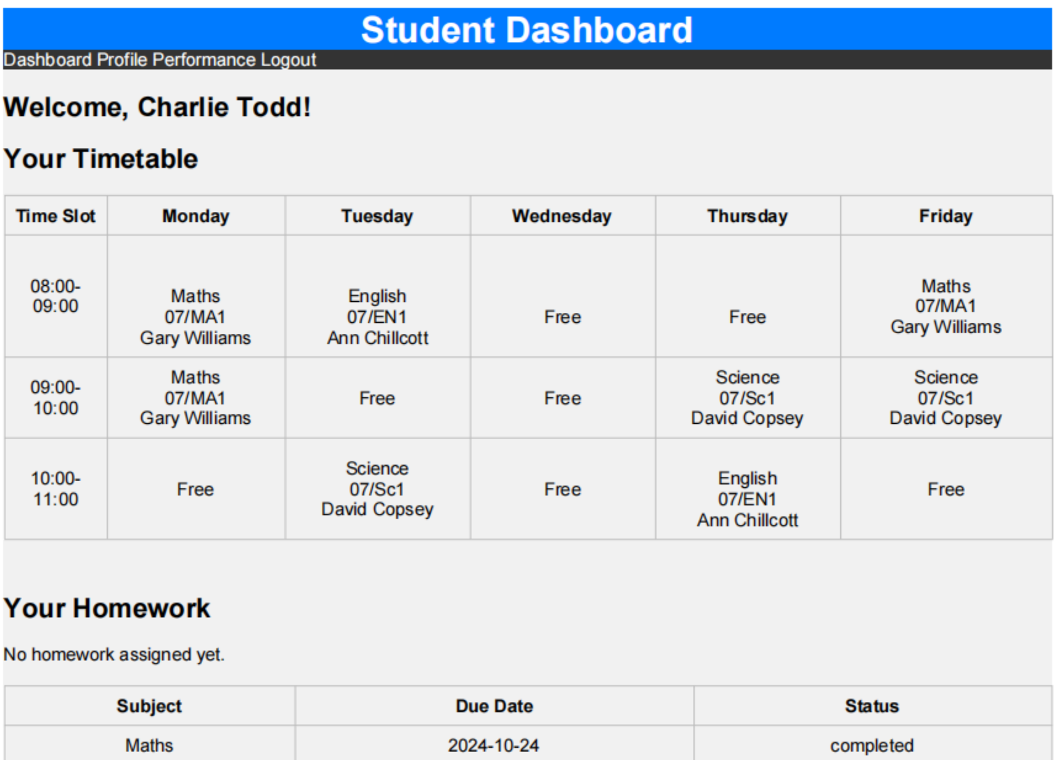


Figure 8. Overview of the Student's Dashboard.

5. Discussion

The research explored how software can track student attendance, behavior and performance, aiming to develop a prototype that meets educational requirements, while contributing to the broader debate on technology adoption in schools. This piece emphasized technical implementation, user experience, and software impact. This section will reflect on the previous sections, summarizing findings, the implications for the research questions, and future improvements to advance the system beyond a prototype.

The results highlight the need for simplified educational systems that prioritize efficiency by minimizing clutter and excessive functions. The system can perform all essential tasks, logging in, tracking attendance, behavior and grades, whilst maintaining speed and clarity, as shown with the student_login.php page loading at 18 ms. Research reinforces that simpler designs enhance usability, without sacrificing functionality. Unlike larger systems like SIMS, the prototype focuses purely on the core features. This aligns with TAM theory, emphasizing ease of use, and CLT theory, which advocates removing unnecessary complexities. Simulation results suggest that simpler systems can effectively support multiple users without having extreme issues, even if performance under multi-user load remains untested and is a priority for future work.

The proposed design showed that high-performance educational tools do not require major resources or complexity. The prototype shows that it is possible for all schools to be able to benefit from student tracking software, regardless of how large the institution is or its finances, as it is built with the main goal of providing easy-to-use software for all users, regardless of their experience. The implications for schools are that they can gain access to tools that reduce workload while offering deeper data insights. This results in more time for educating students, less stress for the staff when analyzing a student's attendance or performance, and easier access to information such as timetables, homework due dates, and grades for previous work or examinations. For developers and designers, the prototype could be used to show that efficiency is key, not the sheer number of features. The main focus for all software developers and designers should always be the user; therefore catering to all levels of experience is a necessity, as shown by the prototype. For policymakers, the research highlights the need for more investment in school technology, ensuring equitable access for all schools, not just those in wealthier areas. Investing in such technology would reduce the gap in quality of education between the social classes, resulting in a fairer education system, and by extension, a fairer society. It could also benefit student behavior [11, 12], but it can also, if the software is not used fairly, result in discrimination [13]. Whilst there is a lack of empirical data used in the study, previous research conveys that there could be a link between the software and student behavior.

Despite the prototype being a success, several limitations remain:

- Testing was limited; it requires real-world testing to confirm functionality. Real-world testing would be the next step in ensuring feasibility of the prototype, which in turn would strengthen the idea that a more simplistic software can be of use to institutions.
- Purposefully simplistic, though it may be less suitable for larger institutions. The idea was to create simple and easy-to-use software. However, with this in mind, the software might not be suitable in its current form for institutions with more elements, such as after-school clubs, large numbers of students and staff or large numbers of departments. The prototype could be developed to make it possible to allow larger volumes of data; however, this could mean it moves away from its initial objectives.
- User feedback was limited to conversations about the design with the supervisor. This means that there is only a limited number of considerations. If more people accessed the prototype, a wider understanding of its strengths and drawbacks would be established.

Addressing these in the future would help enhance the prototype further.

6. Conclusions

The research highlights that simplicity, speed and usability are essential for effective student tracking software, that addresses challenges of the past. It also contributes to the push for greater accessibility in educational technology, benefiting both schools and users with minimal experience. Moving forward, prioritizing the user experience will enhance student tracking software, ensuring effective support for students so they receive the best possible education. The prototype designed for this study could be adapted for mobile phone usage, so that it is easier to access, allowing for third-party integration, and providing customisation options for the software so that schools can alter layouts to their needs, which would make the software more flexible.

Author Contributions

Conceptualization, C.T.; methodology, C.T.; software, C.T.; validation, C.T.; supervision, E.L.S.; writing—original draft preparation, C.T. and E.L.S. Both authors have read and agreed to the published version of the manuscript.

Funding

This work received no external funding.

Institutional Review Board Statement

This study does not contain any studies with human or animal subjects performed by any of the authors. This study has been approved under the Liverpool Hope University Research Ethics Policy.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Acknowledgments

This paper is based on the dissertation submitted by Charlie Todd in partial fulfillment of the requirements for the degree of Bachelor of Arts in Computer Science and Education at the School of Computer Science and the Environment, Liverpool Hope University. We would like to express our thanks to the Team of the School of Computer Science and the Environment for their support and, in particular, to Mr Ian Steel, Mrs Kirstie Mather, Mrs Jo Bond and Ms Shelley Wynne.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Budiarto, M.; Audiah, S.; Astuti, E.D.; et al. Enhancing School and College Attendance Using Advanced Technology. In Proceedings of the 3rd International Conference on Creative Communication and Innovative Technology (ICCIT), Tangerang, Indonesia, 7–8 August 2024; pp. 1–6.
2. Vain, C. Strategies for Digital Safety and Cybersecurity in Schools. Available online: https://cpdonline.co.uk/knowledge-base/safeguarding/strategies-digital-safety-cybersecurity-schools (accessed on 10 March 2025).
3. Marston, D.; Fuchs, L.S.; Deno, S.L. Measuring Pupil Progress: A Comparison of Standardized Achievement Tests and Curriculum-Related Measures. *Diagnostique* **1986**, *11*, 77–90. [CrossRef]
4. Unsworth, R. Teaching through the Cloud: An Ethnography of the Role of Cloud-Based Collaborative Technologies in the Formation of Teachers' Classroom Practices. *Anthropol. Educ. Q.* **2024**, *55*, 24–42. [CrossRef]
5. Kotevski, Z. On the Technologies and Systems for Student Attendance Tracking. *Int. J. Inf. Technol. Comput. Sci.* **2018**, *10*, 44–52.
6. Proctur. The Importance of Student Attendance Tracking Software in Modern Schools. Available online: https://proctur.com/blog/student-attendance-tracking-software/ (accessed on 10 March 2025).
7. Susilowati, T.; Ihsan Dacholfany, M.; Aminin, S.; et al. Getting Parents Involved in Child's School: Using Attendance Application System Based on SMS Gateway. *Int. J. Eng. Technol.* **2018**, *7*, 167–174.
8. Beaudin, K. College and University Data Breaches: Regulating Higher Education Cybersecurity under State and Federal Law. *J. Coll. & Univ. L.* **2015**, *41*, 657.

9. Friend, M. Importance of ICT in Schools & Benefits for Education. Available online: https://www.classroom365.co.uk/ict-in-schools/ (accessed on 10 March 2025).
10. Potashnik, M.; Adkins, D. *Cost Analysis of Information Technology Projects in Education: Experiences from Developing Countries*; Education Section, Human Development Department, World Bank: Washington, DC, USA, 2003.
11. Flannery, N. The Effectiveness of Attendance-Based Rewards at an Alternative High School. *Eiu. Edu.* **2022**, *1*, 1–53.
12. Lorenzen, S.; Hjuler, N.; Alstrup, S. Tracking Behavioral Patterns Among Students in an Online Educational System. *arXiv preprint* **2019**, arXiv:1908.08937.
13. Center for Democracy & Technology. Students and Teachers Warn That Schools' Use of Monitoring Software Is Harming Students. Available online: https://cdt.org/press/students-and-teachers-warn-that-schools-use-of-monitoring-software-is-harming-students/ (accessed on 3 August 2022).
14. Thompson, P. Technology Acceptance Model. Available online: https://open.library.okstate.edu/foundationsofeducationaltechnology/chapter/2-technology-acceptance-model/ (accessed on 15 August 2019).
15. Kurt, S. Constructivist Learning Theory. Available online: https://educationaltechnology.net/constructivist-learning-theory/ (accessed on 21 February 2021).
16. GeeksforGeeks. Software Development Life Cycle (SDLC). Available online: https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/ (accessed on 26 February 2020).
17. Ormazabal, M.; Secco, E.L. A Low-Cost EMG Graphical User Interface Controller for Robotic Hand. In *Proceedings of the Future Technologies Conference (FTC) 2021*; Springer: Cham, Switzerland, 2021; 2, pp. 459–475.
18. Baines, C.; Secco, E.L. Design of a Set of Interfaces to Estimate Whether Computer Games Improve User's Skills and Abilities. *Acta Sci. Comput. Sci.* **2022**, *4*, 22–29.
19. Bell, D.; Secco, E.L. Design of a 3D-Printed Accessible and Affordable Robotic Arm and a User-Friendly Graphical User Interface. In *Fourth Congress on Intelligent Systems*; Springer: Cham, Switzerland, 2024; 1, pp. 195–205.
20. Secco, E.L.; McHugh, D.D.; Buckley, N. A CNN-Based Computer Vision Interface for Prosthetics' Application. In *Wireless Mobile Communication and Healthcare*; Springer: Cham, Switzerland, 2022; pp. 41–59.
21. Yabaku, M.; Ouhbi, S. University Students' Perception and Expectations of Generative AI Tools for Software Engineering. In Proceedings of the 36th International Conference on Software Engineering Education and Training (CSEE&T), Würzburg, Germany, 29 July 2024–1 August 2024; pp. 1–5.
22. Pouyioutas, P.; Zantides, E.; Poveda, M.; et al. Redesigning the User Interface of the InterLearning Software - Graphic Design Meets Computer Science. In Proceedings of the Tenth International Conference on Information Visualisation (IV'06), London, UK, 5–7 July 2006; 811–816.
23. Cardell-Oliver, R.; Zhang, L.; Barady, R.; et al. Automated Feedback for Quality Assurance in Software Engineering Education. In Proceedings of the 21st Australian Software Engineering Conference, Auckland, New Zealand, 6–9 April 2010; pp. 157–164. [CrossRef]

Copyright © 2025 by the author(s). Published by UK Scientific Publishing Limited. This is an open access article under the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Publisher's Note: The views, opinions, and information presented in all publications are the sole responsibility of the respective authors and contributors, and do not necessarily reflect the views of UK Scientific Publishing Limited and/or its editors. UK Scientific Publishing Limited and/or its editors hereby disclaim any liability for any harm or damage to individuals or property arising from the implementation of ideas, methods, instructions, or products mentioned in the content.