International Journal of Education, Leadership, Artificial Intelligence, Computing, Business, Life Sciences, and Society - https://journalviral.org
DOI: https://doi.org/10.65222/VIRAL.2026.2.10.30

Design and Development of Mobile Freelance Tutoring Platform for College Students

Allen Alfred D. Beato
Information Technology Department
Eastern Visayas State University
Tacloban City, Philippines
allenalfred.beato@evsu.edu.ph

Czarro Angelo V. Baldoza
Information Technology Department
Eastern Visayas State University
Tacloban City, Philippines
czarroangelo.baldoza@evsu.edu.ph

Alrie Gian R. Salvedia
Information Technology Department
Eastern Visayas State University
Tacloban City, Philippines
alriegian.salvedia@evsu.edu.ph

Sarah Jane L. Cabral
Information Technology Department
Eastern Visayas State University
Tacloban City, Philippines
sarahjane.cabral@evsu.edu.ph

Diane B. Remot
Information Technology Department
Eastern Visayas State University
Tacloban City, Philippines
diane.remot@evsu.edu.ph

Benito V. Badilla Jr.
Information Technology Department
Eastern Visayas State University
Tacloban City, Leyte
benito.badilla@evsu.edu.ph

***Abstract*:** At the tertiary education level, a notable gap persists in connecting college students seeking academic assistance with peers capable of providing tutorial services. The lack of an efficient and accessible platform for peer-to-peer tutoring often results in students struggling to obtain reliable academic support within their institutions. This study addresses this issue through the development of an Android-based mobile application designed to facilitate peer-to-peer tutoring among college students. The Agile software development methodology was employed to ensure a flexible and iterative development process, allowing for timely responses to evolving system requirements and unforeseen challenges. Serving as an intermediary between tutors and tutees, the application features key functionalities such as user registration, tutor application and approval, session scheduling, and a secure token-based payment system. Additional features include a real-time chat interface, notification system, tutor profiling, session management, tutor rating and review system, and a user reporting mechanism. The design and development process adhered to user-centered design principles and incorporated rigorous testing to enhance usability, reliability, and overall user experience. By effectively bridging the gap between student tutors and those in need of academic support, the mobile tutoring application fosters a collaborative learning environment while also offering an opportunity for student tutors to earn. This platform not only improves access to academic assistance but also encourages peer-driven knowledge sharing within the university community.

Keywords: mobile application, peer-to-peer tutoring, token-based payment, agile methodology, freelance platform

I. Introduction

The digital economy's rapid growth has fostered a major transformation in the world of work: specifically, it covers the emergence of cross-border, web-based digital labor platforms. These are portals of virtual outsourcing, such as Upwork and Fiverr, which have changed the concept of freelancing for knowledge-based work, for example, programming, designing, writing, and tutoring. On such sites, work is assigned and executed virtually by contract employees who can cover almost any field, including information technology, marketing, or even research and development [1]. Freelance work is flexible, allowing individuals to set their own pace and work from any location. Freelancing is becoming a suitable source of extra income for college students, allowing them to have better control over their academic-work balance. College students have limited freelancing skills compared to experienced and established freelancers in the freelancing industry. Thus, the most ideal option for them is to focus on academic-related freelance work like peer tutoring in which they can offer tutorial assistance about course subjects to their fellow students. Moreover, focusing on topics they are better at while fostering a collaborative environment around their peers makes it easier for them to conduct the job comfortably.

However, there is an absence of an online freelancing platform, particularly for peer tutoring, that specifically caters to the needs of college students. There are no existing tutoring platforms for skilled college students wherein they can specifically offer peer tutoring and for students who seek tutorial services from their peers. College students who need tutorial assistance have no access to a tutorial platform that can support their specific needs. College students who aspire to work as freelancers are in a tight position where they must compete against well-established and experienced freelancers to meet the standards of their clients. Moreover, current freelancing platforms are designed to cater to a broad audience and do not account for the constraints faced by college students, such as skill gaps and the necessity for part-time work.

To address the difficulties faced by college students, the study focused on the development of an Android mobile application for peer-to-peer tutoring where tutors can offer tutorial services for tutees who seek academic assistance. By ensuring fair competition and fostering collaboration, the mobile application will empower student-level freelancers to showcase and grow their tutoring skills and gain practical experiences in a controlled and equitable setting. Specifically, the study aims (1) to create a collaborative environment between Peer-Tutors and Peer-Tutees, including an in-app messaging system. (2) to develop a comprehensive profiling system for the Peer-Tutors; and (3) to develop a token-based payment scheme.

DOI: https://doi.org/10.65222/VIRAL.2026.2.10.30

II. Related Studies

A. Digital Freelancing for Students

Freelancing platforms serve as intermediaries where individuals, including students, can offer their services or expertise to clients in exchange for payment. According to a study [2], freelancing involves engaging in contract-based work that leverages one's skills, education, and experience. The terms "freelancer," "free-lancer," or "freelance worker" commonly refer to individuals who are self-employed and not bound to a single employer over the long term [3]. Freelancers have the freedom to choose their clients and work with multiple clients concurrently [2].

Online freelancing fosters a flexible environment, enabling working students to earn income from home and adjust their work schedules around academic commitments. Digital labor platforms - often web or mobile applications - allow freelance tasks to be conducted online and remotely [1]. These platforms facilitate real-time hiring for a wide array of tasks such as IT programming, virtual assistance, marketing, graphic design, language teaching, and even research and development. Freelancing thus supports a globally distributed workforce, removing geographical constraints and enabling participation from anywhere [4].

Despite the opportunities freelancing offers, student freelancers often face limitations due to their restricted experience and skill set. This makes it difficult to gain trust and secure projects, especially from high-demand clients. Consequently, students frequently begin by offering services to peers, a less pressured and more relatable environment. The growing digital economy has led to the rise of freelancing platforms that now increasingly target students, providing accessible avenues for income generation and skill development.

B. Platforms for Freelance Tutoring

Freelancing platforms such as Upwork and Fiverr host millions of users globally [5], employing a service- or project-based employment model [6]. Upwork functions like a job board where clients post project descriptions and freelancers apply based on qualifications. It is particularly suitable for large projects or specialized tasks and covers diverse fields including IT, marketing, writing, and design.

These platforms primarily support knowledge-intensive, independent work that demands coordination and specialized skills [6][7]. By structuring interactions between clients and freelancers, such platforms have significantly influenced the gig economy by streamlining the sourcing, management, and delivery of freelance work.

C. Limitations of Existing Freelancing Platforms for Tutoring

While global platforms such as Upwork and Fiverr offer freelance opportunities, they are highly competitive and not optimized for working students. A study [8] observed that platform complexity, ambiguous ranking systems, and opaque rules often disadvantage freelancers, especially newcomers. These systems can demotivate student freelancers, who may lack the resources and time to engage fully.

Moreover, the gig nature of these platforms can lead to fragmented and unstable work experiences. As noted by [9], freelancers are concerned about how platform dependency diminishes both monetary and non-monetary job value. Students, already managing academic responsibilities, find this instability especially challenging. The prevalence of unpaid labor, driven by market competition and lax regulations, further complicates the freelancing landscape [10].

Transnational platforms also fail to ensure equitable wages across borders, often exacerbating pay disparities [11]. Furthermore, most platforms do not prioritize user wellbeing or professional development. According to [5], current systems lack features that support mentorship, community building, or peer feedback, which are crucial for growth, particularly for students. Trust within virtual communities is also key: [12] found that a strong sense of community and work engagement increases job satisfaction, yet building such trust is difficult for part-time or novice freelancers.

Several mobile applications facilitate academic tutoring by connecting students with professional tutors. These apps typically include features like tutor profiles, availability scheduling, and direct messaging. However, most focus on professional-to-student interactions rather than peer-to-peer tutoring. Moreover, many lack functionalities that support income generation, student profiling, or intelligent tutor-student matching, thereby missing opportunities to support working students comprehensively.

D. Profiling Systems

Some educational institutions utilize student information systems with basic profiling features, collecting data on academic performance, financial aid status, and demographics [13]. Profiling helps in developing recommender systems and personalized learning environments. While these systems are beneficial for administrative purposes, they often lack the granularity required for effective peer tutor matching and do not incorporate income-generating capabilities.

Recommender systems are widely used in various industries and can be applied in education to match students with suitable peer tutors based on criteria such as academic level, subject expertise, learning preferences, and availability. However, the success of these systems is highly dependent on the volume and accuracy of collected data.

E. Challenges Faced By Working Students

Freelancing platforms, although promising, often overlook the unique challenges faced by working students. These platforms cater to a general audience, favoring seasoned freelancers with established portfolios. A case study on Freelancer.com [14] revealed systemic bias towards experienced users, leaving newcomers—including students—struggling to secure projects due to fewer positive reviews or completed jobs.

DOI: https://doi.org/10.65222/VIRAL.2026.2.10.30

This imbalance makes it difficult for students to compete effectively and discourages long-term engagement. Furthermore, the irregular schedules common among working students pose challenges in maintaining consistent academic and work performance. Shift work can interfere with class attendance, leading to reduced academic productivity and increased stress [15].

The need for financial support is another critical issue. A study [16] found that 77.9% of working students emphasized the importance of scholarships, flexible job options, and reduced tuition fees. These findings underscore the importance of part-time jobs, such as freelancing, which offer flexibility and income without compromising academic goals.

By integrating freelancing into a student support framework - especially one tailored to peer tutoring - students can enjoy financial stability while gaining relevant work experience. Such a platform can promote both personal and academic development. However, current freelancing systems rarely meet these nuanced needs. A mobile application combining student profiling, tutor recommender, and token-based payment scheme features presents a practical solution for creating a student-centered, sustainable peer tutoring ecosystem.

III. Method

This study adopted the Systems Development Life Cycle (SDLC) using the Agile development methodology to guide the design and implementation of the mobile application. The Agile approach was selected for its iterative and user-centered nature, which allowed for continuous feedback, rapid prototyping, and incremental improvements. This ensured that the mobile app evolved in alignment with user needs and project goals throughout the development process.

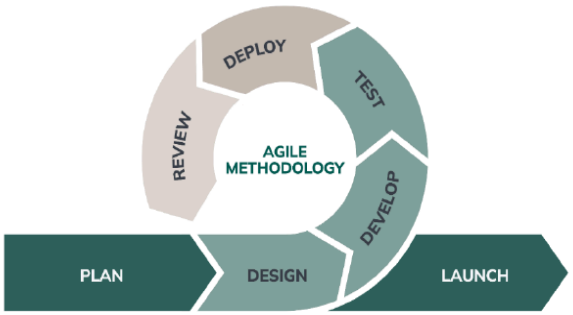


Fig. 1. The Agile Model

Figure 1 illustrates the Agile Software Development Model implemented in this study. The researchers adopted the Agile methodology to guide the Software Development Life Cycle (SDLC) of the mobile tutoring application. As a hybrid of iterative and incremental models, Agile breaks the system into multiple development cycles or iterations, each consisting of smaller, incremental components. This approach emphasizes constant communication, regular feedback, and continuous improvement throughout the development process.

By employing the Agile approach, the researchers were able to maintain flexibility, address challenges as they arose, and adapt efficiently to changes throughout the development of the mobile tutoring application. Phases of the Agile Development Model

The agile software development life cycle methodology follows a sequential order of phases or steps that require careful practice and implementation. This approach consists mainly of seven (7) phases: planning, design, development, testing, deployment, review/evaluation, and launching.

1) Planning Phase

The planning phase is the most crucial step in the agile development model. This phase involves a systematic approach in defining the study's objectives, scope and limitations, and the stakeholders, as well as the system's development timeline. The researchers of the study conducted brainstorming sessions in defining, gathering, and finalizing the foundational requirements of the study.

2) Design

The design phase serves as the blueprint stage of system development, where detailed specifications are created based on the requirements gathered during the analysis phase. This phase translates functional and non-functional requirements into a structured plan that will guide system implementation. Key components of this phase include the System Architecture Diagram, which outlines the system's structural framework and data flow; and the User Interface (UI) Design, which illustrates the visual layout and user experience of the application. These design artifacts ensure that both technical and user-centric aspects of the system are clearly defined before development begins.

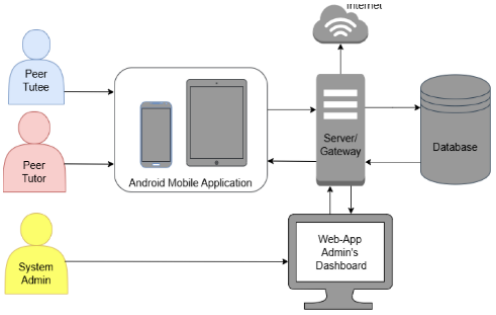


Fig. 2. System Architecture

Figure 2 shows the system architecture, which refers to the overall structure and organization of a system, including its components, their relationships, and the principles and guidelines governing their design and evolution. It provides a high-level view of the system, defining its major components, their responsibilities, and how they interact with each other and with external systems. Moreover, it is concerned with the system's overall design philosophy, the allocation of functionality to different components, and the coordination and integration of those components [17].

The user interface (UI) design of the mobile tutoring application was developed with a focus on simplicity, accessibility, and user engagement. Guided by user-centric design principles, the interface aimed to provide a seamless

DOI: https://doi.org/10.65222/VIRAL.2026.2.10.30

experience for both peer tutors and tutees. The design prioritized intuitive navigation, clean layout structures, and responsive elements to ensure that users—especially college students—could easily access key features.

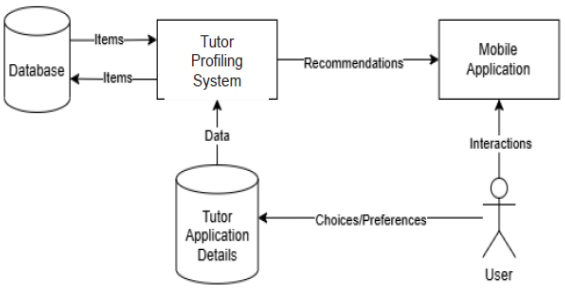


Fig. 3. Profiling System Diagram

Figure 3 shows that the profiling system works by taking data from the user's options or preferences through a score-based application form. The gathered data was used to create the tutor's profile, which is then used to suggest or recommend to the tutee during tutor selection. The researchers implemented the score-based application form for the mobile tutoring application as the tutor profiling system. This approach aims to significantly enhance the user experience on the mobile application by filtering the tutors based on their status or living conditions and to identify which among them are the "most needy" for an income. The tutor application form contains input and selection fields that follow a certain criterion which will determine the (most needy) score of the tutor applicant. Moreover, the score will dictate their placement in the pool of recommended tutors for the tutees.

3) Development

The development phase in agile approach started by building the first iteration of the software. The development phase encompasses all relevant production tasks in the software development life cycle, such as UX/UI design and coding [18]. The development (or iteration) stage is critical to the agile approach as it allows the team to create a product with few features and then add more functionality later. Once the development stage is completed, it is time to do quality assurance activities, prepare technical documentation, and conclude the iteration [19]. In the development phase, the researchers started the development of the mobile tutoring application by focusing on the fundamental and foundational functionalities followed by the major and more complex functionalities.

a) Programming Environment

The development of the mobile tutoring application involved a combination of front-end and back-end technologies to ensure a responsive, scalable, and user-friendly system. For the front-end, Jetpack Compose was utilized as the UI framework for the Android mobile application. Jetpack Compose is an open-source declarative UI toolkit for Android development using Kotlin and is known for its modern design and ease of use in creating native user interfaces. For the web-based administrator dashboard, React JS was employed due to its efficiency and flexibility in building dynamic and responsive user interfaces.

On the back-end, Kotlin was used as the primary programming language for mobile development. Kotlin is a general-purpose, cross-platform, high-level language that supports type inference and static typing, making it well-suited for robust Android development. The application also incorporated Firebase Database, a NoSQL database known for its rich feature set and ease of use, making it ideal for mobile applications. For the web-based admin dashboard, Node.js served as the server-side framework, providing the necessary backend support for managing administrative functions efficiently.

4) Testing

The testing phase allowed the researchers to conduct a variety of testing techniques to ensure the code is clean and safe from potential bugs, errors, or vulnerabilities before the deployment of the system. This is the process of testing by executing the software under certain conditions to confirm its correctness and completeness and to meet the required quality. Software testing is intended primarily to reveal flaws, voids or any defects that exist between the current and desired states of the software. It encompasses diverse approaches and levels like unit testing, integration testing, and system testing, among others that focus on distinct sections of the software [20]. The researchers created a test plan to outline the testing objectives to be followed and activities that must be carried out during the testing phase.

5) Deployment

The deployment phase refers to the process of making a system available for use. This process is a crucial part of the software development life cycle, as it ensures that the software is functional, user-friendly, and ready for public or private use. During the deployment phase, the developers ensure that the software is ready and available for use on a target device [21].

6) Review

The review phase focuses on verifying and validating the quality and performance of the system before, during, and after its deployment through the pilot testing. The researchers conducted an overall system review to verify if the mobile application correctly implements its intended functionalities and validated if it is aligned with the users' requirements. Furthermore, the review phase highlights if the system requires major or minor changes. The researchers utilized a project implementation checklist and implementation contingency for the system review of the mobile tutoring application.

B. Evaluation

1) User Acceptance

In the context of system evaluation, it is essential to consider the users' perceptions of the system's quality to

DOI: https://doi.org/10.65222/VIRAL.2026.2.10.30

ensure they are aligned with recognized standards. To verify and validate if the mobile tutoring application meets the specific requirements or standards, the researchers will conduct a User Acceptance Test (UAT) on the deployment phase of the mobile application. User Acceptance Testing (UAT), also known as end-user or application testing, as a testing method to validate software in real-world conditions by its intended audience [22]. The User Acceptance Test (UAT) will serve as the system's quality evaluation and will utilize the ISO 25010 software quality standards or criteria.

The researchers utilized the Likert scale to quantitatively measure the participants' ratings for the mobile application's characteristics or quality as stated in the evaluation form for the User Acceptance Test (UAT). The rating scale will have a 5-point Likert scale that allows the participants to express their varying degrees of agreement or disagreement on the system's specific attributes without overwhelming them with too many options. The 5-point Likert scale will be structured as follows:

5 – Strongly Agree
4 – Agree
3 – Neutral
2 – Disagree
1 – Strongly Disagree

IV. Results and Discussion

The study achieved its objectives in developing a peer-to-peer mobile tutoring application designed for college students. The following discussions reflected the findings of the specified objectives in developing the mobile tutoring application:

A. Create a collaborative environment between the skilled peer-tutors and peer-tutees, including an in-app messaging system

A collaborative environment between peer tutors and peer tutees was established through the integration of a real-time in-app messaging system. This feature enabled seamless communication, allowing users to coordinate tutoring sessions and clarify academic concerns efficiently. Usability tests indicated that users found the messaging interface intuitive and essential in fostering interaction and engagement between users.

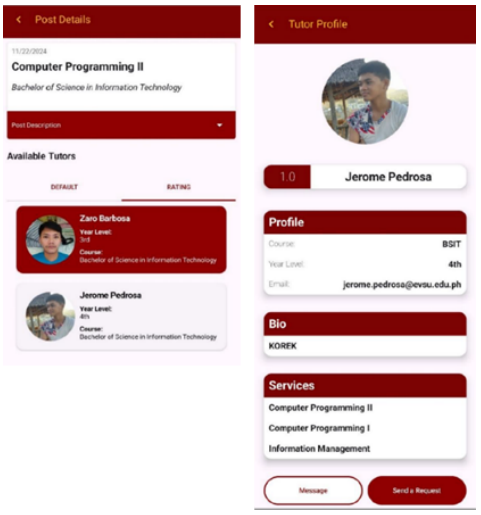


Fig. 4. Recommendation list of tutors

Figure 4 shows the recommended tutor list with key details and profile access for peer tutees to initiate academic support. Tutor profile interface showing subject expertise for services, bio, and contact options to encourage informed selection and collaboration.

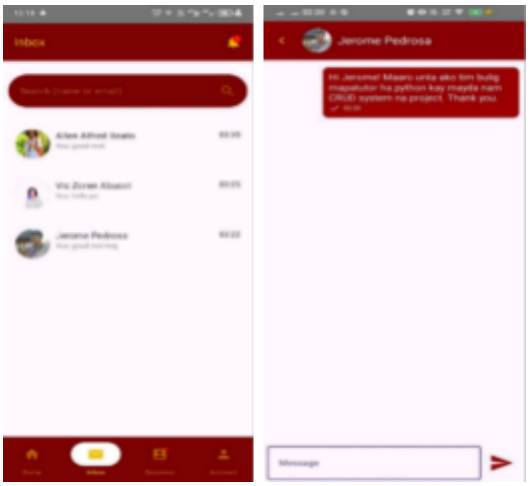


Fig. 5. Inbox & Messaging Screens

In figure 5, users can communicate with each other without the need for external messaging or chat applications. The users can easily navigate to the inbox screen as it is part of the navigation bar at the bottom of the mobile application. The user's inbox contains existing conversations of the user. It also contains a search bar so that the users can search for other users to chat with using their names or email addresses. Users are also allowed to delete conversations by sliding a conversation item to the left and confirming by clicking the trash icon on the right side of the selected item. The conversation screen, just like other messaging applications, contains the name of the user's chatmate, the messages in ascending order based on date and time sent, indicators if the message is sent or seen by the recipient, and the message input field and send button.

DOI: https://doi.org/10.65222/VIRAL.2026.2.10.30

B. Develop a comprehensive profiling system for the skilled Peer-Tutors.

A comprehensive profiling system was developed for peer tutors. This module included essential details such as academic year, subject expertise, schedule availability, and ratings received from past sessions. In addition, the system considered the tutor's expressed need to earn supplemental income, which helped prioritize those who sought tutoring opportunities not only for academic collaboration but also for financial support. This feature allowed tutees to make informed decisions in selecting their tutors based on transparent and relevant criteria. Feedback from user testing revealed that the profiling system contributed to building trust and professionalism among users, while also promoting social equity by providing income-generating opportunities to students in need.

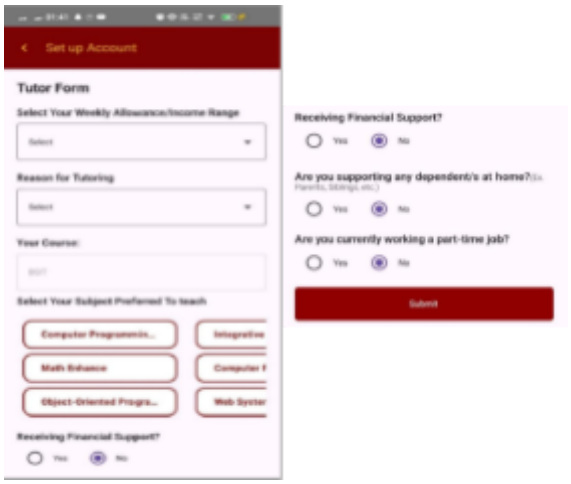


Fig. 6. Tutor Application & Tutor Profiling

Figure 6 shows the developed tutor application form for students who want to become a tutor. The data from the tutor application form will be reflected in the tutor's profile once the administrator approves the application. The tutor application form will undergo a score-based profiling system which will identify if the tutors are needy for an income based on their overall scores.

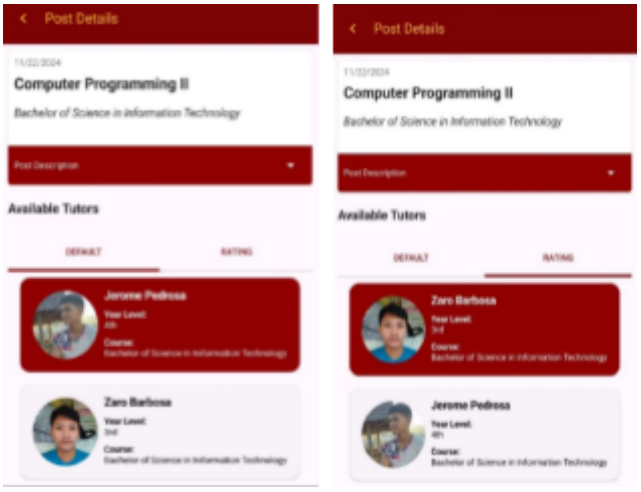


Fig. 7. Tutor Recommendation

Figure 7 shows the developed and implemented pool of recommended tutors for the tutee's selected subject. The pool of tutors is arranged based on two options, which are by default arrangement or based on the tutors' average ratings. The default arrangement pertains to the "most needy of income" tutors, which sorts them based on their "most needy" score. On the other hand, the tutors can also be arranged based on their average ratings. The purpose of having two kinds of arrangements or sorting is to give the tutees options on what to look for in a tutor and to give tutors fair chances of getting selected for the job.

C. Develop a token-based payment scheme

The token-based payment scheme was successfully implemented. This scheme enabled a secure and manageable payment process, wherein tutees could purchase tokens and use them to book sessions with tutors. The transaction system ensured fairness and transparency while minimizing the need for direct monetary exchange. The design also included a transaction history module for both tutors and tutees, which was positively rated by users for its simplicity and accountability.

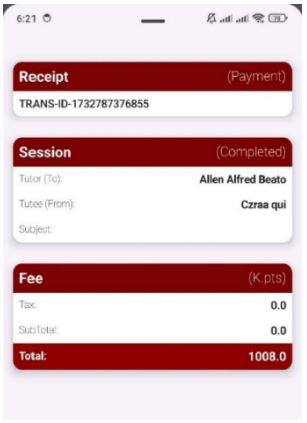
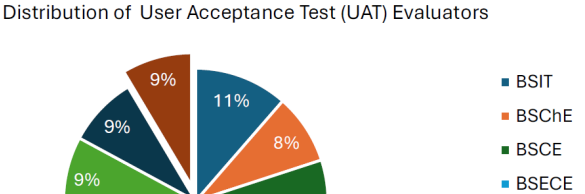


Fig. 8. Token-based Payment Scheme

Figure 8 shows a sample transaction receipt which shows that the researchers have successfully developed and implemented a token-based payment scheme for tutorial sessions between Tutees and Tutors. The payment scheme requires the tutees to load tokens to their accounts by submitting a token purchase request to the administrator through the mobile application. The payment for this type of transaction will be handed personally by the tutee to the administrator for confirmation. All pending transactions (invoices) and completed transactions (receipts) are in the user's transactions history.

D. System Evaluation

The researchers conducted the user acceptance test for the software evaluation during the pilot testing of the mobile tutoring application. This test determines if the system meets the client's needs and expectations as they anticipate. The researchers presented the data acquired through the survey questionnaire in both graphical and tabular forms.



DOI: https://doi.org/10.65222/VIRAL.2026.2.10.30

Fig. 9. Distribution of User Acceptance Test (UAT) Evaluators

Figure 9 presents a pie chart of the distribution of the types of User Acceptance Test (UAT) evaluators. The end-user population (35 students) were randomly selected from the eight (8) programs under the College of Engineering in Eastern Visayas State University – Main Campus. Figure shows that the majority of the evaluators were students from the Bachelor of Science in Civil Engineering (BSCE) with 26% (9 students) from the total population. The second highest number of evaluators were from the Bachelor of Science in Electrical Engineering (BSEE) with 20% (7 students). The Bachelor of Science in Information Technology (BSIT) has the third highest number of evaluators with 11% (4 students). The rest of the engineering programs, which are the Bachelor of Science in Chemical Engineering, Geodetic Engineering, Industrial Engineering, and Mechanical Engineering, have the same number of evaluators with 8-9% (3 students).

TABLE 1. SYSTEM QUALITY EVALUATION RESULTS

Characteristics	Average Rating	Interpretation
Functional Suitability	4.21	Strongly Agree
Interaction Capability	4.34	Strongly Agree
Reliability	4.42	Strong Agree
Performance Proficiency	4.17	Agree
Security	4.63	Strong Agree
OVERALL RATING	4.35	Strong Agree

Table 1 present the mobile tutoring application was evaluated based on key software quality characteristics. The results show high user satisfaction across all categories, with an overall average rating of 4.35 interpreted as Strongly Agree. Security received the highest rating (4.63), indicating strong trust in data protection features. Reliability (4.42) and Interaction Capability (4.34) also scored highly, reflecting stable performance and user-friendly design. Functional Suitability rated 4.21, confirming the app meets its intended purpose. Performance Proficiency, while still positive at 4.17, was the lowest, suggesting room for optimization in system responsiveness.

Overall, the application effectively supports peer tutoring for students, with results confirming its usability, functionality, and relevance to its target users.

V. CONCLUSION

As a final point, the mobile tutoring application successfully provided an accessible and collaborative platform for peer-to-peer tutoring among students in a university. By incorporating major features such as tutor selection, in-app messaging, session management and documentation, and a structured token-based payment system, the app enhances the tutoring experience for both tutors and tutees. The score-based profiling system ensures that student tutors who need financial support have better opportunities to offer tutoring services, while the token-based payment system ensures secure and efficient transactions. Moreover, the results from the User Acceptance Test (UAT) for the system evaluation reflected that the mobile application is functional, efficient, user-friendly, and secure. The positive feedback from end-users highlights the effectiveness of the mobile tutoring application in meeting its objectives, reinforcing its value as a tool for academic support and student engagement. This innovation demonstrates how technology can be maximized to create a supportive learning environment, ultimately benefiting both tutors and tutees in their academic journeys.

The development of the mobile tutoring application is not confined to its current features, as there are several areas for potential enhancement. The researchers recommend the following improvements for future developers. First, adding a feature to extend ongoing tutorial sessions would allow tutees to continue with the same tutor without initiating a new session, saving both time and effort. Second, integrating an online payment gateway would improve the current token-based system, which currently supports only face-to-face transactions. This enhancement would enable seamless payments for remote or online tutoring, especially as the system scales beyond its initial scope. Lastly, enriching the in-app chat system by allowing users to send files, images, and voice messages - similar to widely-used messaging platforms - would improve communication, increase user engagement, and make interactions more flexible and convenient.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Eastern Visayas State University – College of Engineering Information Technology Department for providing the support and resources necessary for the development of this research.

REFERENCES

[1] Shkurupii, O., Diachkov, D., & Holos, A. (2021). FREELANCE AS FORM OF VIRTUALIZED EMPLOYMENT AND PERSONNEL DEVELOPMENT MANAGEMENT. Market Infrastructure, 62. https://doi.org/10.32843/infrastruct62-19

[2] Kaur, B., Manohar, R. M. S. N., Vamsi, R., & Teja, G. E. S. (2020). Online Freelancing website. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 509–513. https://doi.org/10.32628/cseit2172110

[3] Khan, F. S., Khalid, M., Ali, A. H., Bazighifan, O., Nofal, T. A., & Nonlaopon, K. (2022). Does freelancing have a future? Mathematical analysis and modeling. Mathematical Biosciences and Engineering, 19(9), 9357–9370. https://doi.org/10.3934/mbe.2022435

[4] Graham, M., & Anwar, M. A. (2019). The global gig economy: Towards a planetary labour market? First Monday. https://doi.org/10.5210/fm.v24i4.9913.

[5] De La Vega, J. C. A., Cecchinato, M. E., & Rooksby, J. (2022). Design Opportunities for Freelancing Platforms: Online Freelancers' Views on a Worker-Centred Design Fiction. https://doi.org/10.1145/3533406.3533410

[6] De La Vega, J. C. A., Cecchinato, M. E., Rooksby, J., & Newbold, J. (2023). Understanding Platform Mediated Work-Life: A Diary Study with Gig Economy Freelancers. Proceedings of the ACM on Human-computer Interaction, 7(CSCW1), 1–32. https://doi.org/10.1145/3579539

[7] Kinder, E., Jarrahi, M. H., & Sutherland, W. (2019). Gig Platforms, Tensions, Alliances and Ecosystems. Proceedings of the ACM on Human-computer Interaction, 3(CSCW), 1–26. https://doi.org/10.1145/3359314

[8] De La Vega, J. C. A. (2020). Making the gig economy Work for Workers: An Exploration of Freelancing Platforms. Electronic Workshops in Computing. https://doi.org/10.14236/ewic/hci20dc.13

[9] Nemkova, E., Demirel, P., & Baines, L. (2019). In search of meaningful work on digital freelancing platforms: the case of design professionals. New Technology, Work and Employment, 34(3), 226–243. https://doi.org/10.1111/ntwe.12148

[10] Marà, C., & Pulignano, V. (2022). Unpaid labour in online freelancing platforms: between marketization strategies and self-employment regulation. Sociologia Del Lavoro, 163, 130–148. https://doi.org/10.3280/sl2022-163007

[11] Banik, N., & Padalkar, M. (2021). The Spread of Gig Economy: Trends and Effects. Foresight and STI Governance/Forsajt, 15(1), 19–29. https://doi.org/10.17323/2500-2597.2021.1.19.29

[12] Toth, I., Heinänen, S., & Blomqvist, K. (2020). Freelancing on digital work platforms – roles of virtual community trust and work engagement on person–job fit. VINE Journal of

DOI: https://doi.org/10.65222/VIRAL.2026.2.10.30

Information and Knowledge Management Systems, 50(4), 553–567. https://doi.org/10.1108/vjikms-12-2018-0124

[13] Eke, C. I., Norman, A. A., Shuib, L., & Nweke, H. F. (2019). A survey of user profiling: State-of-the-Art, Challenges, and Solutions. IEEE Access, 7, 144907–144924. https://doi.org/10.1109/access.2019.2944243

[14] Green, D. D., Polk, X., O'Donnell, H., Doughty, K., Carr, M., & Costa-Cargill, D. (2021). The gig Economy: A case study analysis of Freelancer.com. Management and Economics Research Journal, 7(2), 1–6. https://doi.org/10.18639/merj.2021.1413412

[15] Khanal, S. P. (2019). Irregular Attendance of University Students at Class and its Relation to their Academic Achievement. Deleted Journal, 33(1), 115–128. https://doi.org/10.3126/tuj.v33i1.28687

[16] Summer, R., McCoy, M., Trujillo, I., & Rodriguez, E. (2023). Support for Working Students: Understanding the Impacts of Employment on Students' Lives. Journal of College Student Retention, 152102512211492. https://doi.org/10.1177/15210251221149209

[17] Fang, Z. (2021). System-of-Systems Architecture Selection: A survey of issues, methods, and opportunities. IEEE Systems Journal, 16(3), 4768–4779. https://doi.org/10.1109/jsyst.2021.3119294

[18] Ghumatkar, R. S., & Date, A. (2023). Software Development Life Cycle (SDLC). International Journal for Research in Applied Science and Engineering Technology, 11(11), 1162–1165. https://doi.org/10.22214/ijraset.2023.56554

[19] EtAl, R. S. (2021). Towards adopting software quality assurance in agile development methodology. Türk Bilgisayar Ve Matematik Eğitimi Dergisi, 12(3), 2152–2157. https://doi.org/10.17762/turcomat.v12i3.1158

[20] Tsahat, C. O. O., & Ndzeli, N. A. (2024). SOFTWARE TESTING AND ITS ASPECTS. *SYNCHROINFO JOURNAL*, *10*(1), 2–7. https://doi.org/10.36724/2664-066x-2024-10-1-2-7

[21] Hossain, M. I. (2023). Software Development Life Cycle (SDLC) Methodologies for Information Systems Project management. *International Journal for Multidisciplinary Research*, *5*(5). https://doi.org/10.36948/ijfmr.2023.v05i05.6223

[22] Dalton, J. (2018). Acceptance testing. In Apress eBooks (pp. 111–112). https://doi.org/10.1007/978-1-4842-4206-3_8