Academic Journal of Education
ISSN: 3080-9940 (Print), ISSN: 3080-9959 (Online) | Vol. 2, No. 1, 2026

A Project-Based Bilingual Instructional Framework for Android Development Education under the New Engineering Education Initiative

Erli Wang *, Jingyi Yang, Wenzhong Zhu, Xiaofang Liu, Chao Chen, Yani Hou

School of Computer Science and Engineering, Sichuan University of Science & Engineering, Yibin, Sichuan, 644000, China

*** Corresponding author:** Erli Wang

Abstract: With the rapid advancement of mobile Internet technologies and artificial intelligence (AI), engineering education increasingly demands interdisciplinary talent, practical engineering competence, and global communication skills. As a core course for Internet of Things (IoT) Engineering students, Mobile Development and Application plays a crucial role in cultivating software development capabilities. However, traditional teaching approaches often suffer from insufficient practical training, limited integration of English technical resources, low student engagement, and inadequate assessment methods. To address these challenges, this study proposes a project-based bilingual teaching model under the New Engineering Education (NEE) framework. The proposed model integrates bilingual instruction, collaborative APP development, project-driven learning, and AI-assisted programming support into Android development education. The reform practice was implemented at Sichuan University of Science and Engineering (SUSE) from 2024 to 2025 with undergraduate students majoring in IoT. Results from academic performance analysis, questionnaire surveys, and competition achievements demonstrate that the proposed approach significantly improved students' programming competence, English technical literacy, teamwork ability, and learning motivation. The findings indicate that integrating bilingual project-based learning (PBL) with AI-assisted educational strategies can effectively enhance engineering education quality and cultivate innovative software engineering talents with international competitiveness.

Keywords: New Engineering Education; Bilingual Teaching; Android Development; Project-Based Learning; Teaching Reform; Mobile Application Development.

1. Introduction

The rapid development of cloud computing, artificial intelligence, and smart terminal devices has significantly accelerated the demand for mobile software development talents worldwide [1]. As one of the most widely adopted mobile operating systems, Android has become a core technology platform in software engineering and IoT education [2]. Consequently, Android application development courses are now essential components of computer science and related engineering curricula.

Meanwhile, the implementation of the NEE initiative in China has highlighted the need to cultivate engineering graduates with interdisciplinary knowledge, practical innovation capabilities, international perspectives, and collaborative competence [3]. However, mobile application development courses remain largely lecture-oriented and focused on fragmented programming exercises, limiting students' opportunities to develop comprehensive engineering skills [4]. This challenge is further intensified by the rapid evolution of Android technologies, the increasing complexity of mobile application development, and the widespread reliance on English-language documentation, development tools, and open-source resources. As a result, students often encounter difficulties in effectively integrating technical knowledge with engineering practice while simultaneously developing professional communication skills.

Previous studies have demonstrated the effectiveness of PBL, bilingual instruction, and AI-assisted programming in improving learning outcomes and student competencies [5–7]. However, limited research has explored the systematic integration of these approaches in mobile development education under the NEE framework. Moreover, many mobile development courses still lack authentic engineering projects, collaborative development experiences, and assessment mechanisms aligned with real-world software engineering practices.

To address these challenges, this study proposes a project-based bilingual teaching model for the course Mobile Development and Application at SUSE. The proposed framework integrates bilingual instruction, PBL, collaborative application development, and AI-assisted programming support into a unified teaching process. This study aims to enhance students' Android development competence, engineering practice ability, and professional English communication skills while evaluating the effectiveness of AI-assisted bilingual programming education. The findings provide practical evidence and pedagogical insights for engineering education reform in the era of intelligent education.

2. Literature Review

PBL has become one of the most widely adopted teaching approaches in engineering education [8,9]. Compared with traditional lecture-centered instruction, PBL emphasizes authentic tasks, collaborative problem solving, and practical engineering implementation [10]. Numerous studies have demonstrated that project-based approaches can significantly enhance students' engagement, engineering competence, and practical problem-solving abilities [11]. In computer science and software engineering education, PBL is particularly

valuable because software development inherently involves iterative design, teamwork, and continuous refinement. By engaging students in complete software development projects, PBL facilitates the integration of theoretical knowledge with engineering practice while promoting software design thinking, project management capabilities, and collaborative skills [12]. For mobile application development courses, project-based activities provide opportunities for students to experience the full software development lifecycle, from requirements analysis and interface design to coding, testing, and deployment.

With the globalization of the information technology industry, English has become the primary language for technical documentation, software development frameworks, open-source communities, and professional communication [13]. In Android development, most official resources, including development guides, APIs, technical forums, and open-source repositories, are published in English. Consequently, students' ability to access and utilize these resources has become an important component of professional competence. Bilingual instruction has attracted increasing attention in engineering education as an effective means of enhancing students' access to international technical knowledge while improving professional communication skills [14]. Previous studies have shown that bilingual teaching can strengthen disciplinary English literacy and improve students' international competitiveness [15]. However, challenges such as language barriers, limited bilingual teaching resources, and differences in students' English proficiency continue to affect learning outcomes. Therefore, effective bilingual teaching strategies should be closely integrated with disciplinary learning and authentic engineering tasks.

Recent advances in generative AI have introduced new opportunities for software engineering education. AI-assisted programming tools can provide intelligent support for code generation, debugging, error diagnosis, and programming explanations, thereby reducing learning barriers and improving programming efficiency [16]. Existing studies suggest that AI-assisted learning can promote self-directed learning and improve programming performance [17]. Despite these benefits, concerns have been raised regarding students' potential overreliance on AI-generated solutions, which may weaken independent problem-solving and software design capabilities [18]. As a result, the educational value of AI tools depends largely on how they are incorporated into instructional activities. Integrating AI assistance within PBL environments may enable students to benefit from intelligent support while maintaining active engagement in software design, implementation, and engineering decision-making processes.

The literature suggests that PBL, bilingual instruction, and AI-assisted programming address complementary dimensions of engineering education. PBL emphasizes engineering practice and collaborative problem-solving, bilingual instruction facilitates access to global technical resources and professional communication, while AI-assisted programming provides personalized learning support and immediate feedback. Together, these approaches have the potential to create a comprehensive learning environment for software engineering education. However, existing studies have primarily investigated these approaches in isolation, and limited attention has been paid to their systematic integration within Android development education under the NEE framework. Furthermore, research remains scarce on instructional models that simultaneously support engineering practice, bilingual competence, collaborative development, and AI-enhanced learning. This gap highlights the need for a comprehensive teaching framework that integrates these dimensions and aligns with the competency-oriented objectives of NEE.

3. Methodology

3.1. Course Context and Participants

The teaching reform was implemented in the undergraduate course Mobile Development and Application for IoT major at SUSE during the 2024–2025 academic year. As a core optional course, it aims to develop students' mobile application development skills and engineering practice capabilities. The course comprised 12 weeks of classroom instruction and 4 weeks of laboratory practice, during which students completed collaborative Android application development projects and related assessments. A total of 110 undergraduate students participated in the study.

3.2. Framework of the Proposed Teaching Model

The proposed teaching model follows a student-centered and outcome-oriented philosophy under the NEE framework. It integrates four key components: PBL, bilingual instruction, collaborative application development, and AI-assisted programming support. PBL serves as the core instructional approach, enabling students to participate in authentic application development throughout the software development lifecycle. Bilingual teaching is embedded into course materials, technical documentation, classroom activities, and project reports to strengthen students' access to international technical resources and enhance professional communication skills. Collaborative development activities are organized through team-based projects to cultivate teamwork awareness and software engineering practices. In addition, AI-assisted programming tools are incorporated to support code generation, debugging, error diagnosis, and technical knowledge retrieval, thereby improving learning efficiency and providing timely feedback during project implementation. Through the integration of these components, the teaching model seeks to promote students' technical competence, engineering practice ability, collaboration skills, and professional English literacy. The overall framework of the proposed model is illustrated in Figure 1.

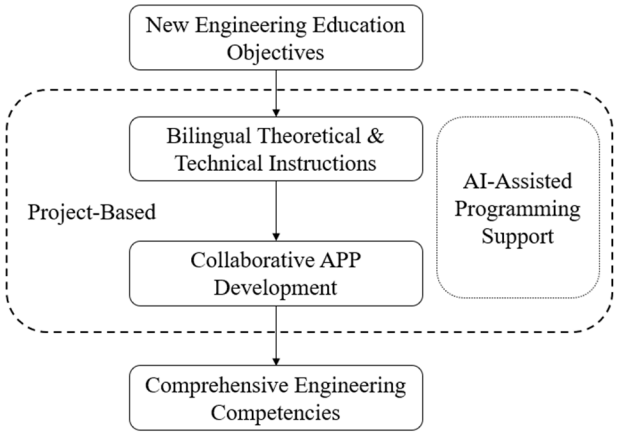

Figure 1. Framework of the teaching reform model

3.3. Projected-based Teaching Design

The course adopted a project-based instructional design centered on Android application development. Students were encouraged to propose project topics based on their interests and subsequently formed development teams. To align learning activities with software engineering practice, the instructor organized course content into task-oriented modules corresponding to key stages of the mobile application development lifecycle, including requirements analysis, user interface design, database implementation, network communication, software testing, and deployment.

Each project was completed through iterative development cycles involving requirement discussion, system design, coding, testing, project presentation, and revision. This process enabled students to gain hands-on experience with practical software engineering workflows while applying course knowledge in authentic development scenarios.

3.4. Integration of Bilingual Resources

Bilingual resources were incorporated throughout the teaching process. Course materials included official Android documentation, API references, technical tutorials, and instructional videos in English. Key concepts, development frameworks, and software engineering terminology were explained using both English and Chinese during classroom instruction.

In project activities, students were encouraged to consult English-language resources, prepare technical documents, and deliver project presentations using bilingual materials. This approach facilitated the gradual integration of language learning with technical learning tasks.

3.5. AI-Assisted Learning Support

AI-assisted programming tools were introduced as supplementary learning resources during project development. Students used AI tools to support code debugging, error diagnosis, API interpretation, technical translation, and programming concept clarification. These tools provided timely assistance when students encountered unfamiliar development frameworks or implementation challenges. To ensure meaningful learning, AI-generated code and solutions were not accepted without verification. During project demonstrations and oral defenses, students were required to explain system architecture, implementation strategies, and key algorithms, thereby demonstrating their understanding of the development process.

3.6. Reconstruction of Assessment System

Table 1. The assessment components and weightings

Assessment Component	Proportion
Project development	40%
Team collaboration	20%
English documentation and presentation	20%
Classroom participation and daily performance	20%

To better evaluate students' engineering competencies and project outcomes, a multidimensional assessment framework integrating both formative and summative approaches was implemented. The assessment comprised four components: project development, team collaboration, English documentation and presentation, and classroom participation. Students' performance was continuously evaluated throughout the semester based on project deliverables, teamwork effectiveness, technical reports, presentations, and learning engagement, providing a more comprehensive measure of learning achievement than traditional examination-oriented assessment. Table 1 summarizes the assessment components and their corresponding weightings.

4. Results

4.1. Academic Performance Analysis

To assess the effectiveness of the proposed teaching reform, course performance data from 2021 to 2024 were analyzed. Table 2 presents the distribution of student scores and average course grades, while Figure 2 illustrates the corresponding trends.

Table 2. Course scores across different years

Year	Score					Average Score
	0-59	60-69	70-79	80-89	90-100	
2021	1.55%	1.55%	6.98%	68.99%	20.93%	80.62
2022	0.70%	2.10%	6.99%	60.84%	29.37%	81.61
2023	0%	1.39%	4.17%	77.78%	16.67%	80.97
2024	0%	0%	0.91%	81.82%	17.27%	81.64

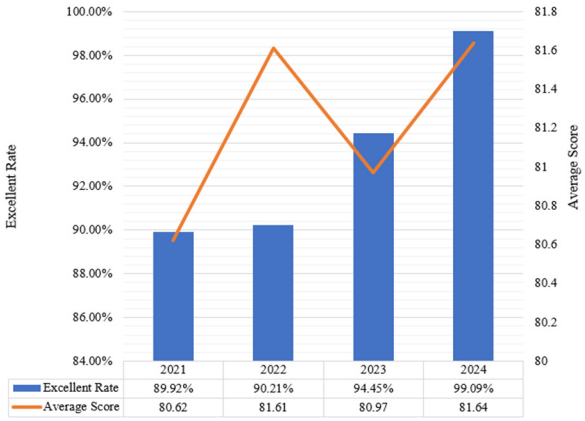

Figure 2. Key performance indicators

As shown in Table 2 and Figure 3, students' performance remained consistently strong across the four years, with average scores exceeding 80 in all cohorts. Prior to the teaching reform (2021–2023), the average score fluctuated slightly between 80.62 and 81.61, while the excellent rate increased from 89.92% to 94.45%. Following the implementation of the teaching reform in 2024, student performance demonstrated further improvement. The average score reached 81.64, the highest among all cohorts, while the excellent rate increased substantially to 99.09%, representing an increase of 4.64 percentage points compared with 2023. This result suggests that nearly all students achieved excellent performance under the reformed teaching model.

The score distribution in Table 2 further supports this trend. The proportion of students scoring below 70 declined steadily from 3.10% in 2021 to 0% in 2024, and no student failed the course in either 2023 or 2024. Meanwhile, the proportion of students achieving scores in the 80–89 range increased from 68.99% in 2021 to 81.82% in 2024, becoming the dominant performance category. Although the percentage of students scoring 90–100 showed some fluctuation across the four years, the overall score distribution shifted toward higher achievement levels after the reform.

These findings suggest that the integration framework was associated with improved learning outcomes. The reform not only maintained stable average performance but also reduced low achievement levels and increased the proportion of

students attaining high scores, thereby contributing to a more balanced and effective learning environment.

4.2. Questionnaire Survey Analysis

To further evaluate students' perceptions of the teaching reform, a structured questionnaire based on a five-point Likert scale was administered at the end of the course. The questionnaire assessed students' views on programming competence, English reading ability, teamwork, learning motivation, and problem-solving skills. The mean scores for each evaluation item are presented in Table 3.

Table 3. Mean scores for key evaluation items

Evaluation Item	Mean Score
Improved programming ability	3.9
Enhanced English reading ability	3.7
Strengthened teamwork competence	3.7
Increased learning motivation	3.8
Improved problem-solving ability	3.7

The survey results reveal generally positive student perceptions of the reformed teaching model, with all evaluation items receiving mean scores above 3.7. Among the assessed dimensions, improvement in programming ability received the highest mean score (3.9), indicating that the integration of project-based development activities and AI-assisted programming support effectively facilitated the acquisition of technical knowledge and practical coding skills.

Learning motivation achieved a mean score of 3.8, suggesting that authentic APP development projects and collaborative learning activities contributed to higher levels of student engagement and participation throughout the course. In addition, English reading ability, teamwork competence, and problem-solving ability each obtained a mean score of 3.7. These results imply that bilingual instruction and project collaboration provided opportunities for students to strengthen their technical communication skills, cooperative learning abilities, and independent problem-solving capabilities.

4.3. Engineering Practice Achievements

Beyond academic performance and questionnaire outcomes, the implementation of the teaching reform was accompanied by a range of engineering and educational achievements. During the reform period, students participated in various innovation and practice activities, resulting in the publication of educational research papers, the registration of software copyrights, the completion of an undergraduate innovation and entrepreneurship project, and several awards in the Chinese Collegiate Computer Design Competition.

The APP development projects completed during the course covered diverse application domains, including smart tourism services, cultural heritage virtual simulation systems, and digital platforms for the preservation and dissemination of traditional culture. These projects required students to complete the full software development process, thereby providing authentic engineering practice experiences.

The engineering outputs generated during the course demonstrate students' ability to apply programming knowledge and software engineering principles to practical development tasks. The diversity of project themes and the resulting achievements suggest that the project-based bilingual teaching model provided students with opportunities to integrate technical knowledge, teamwork, communication skills, and innovation capabilities in real-world development contexts.

5. Discussion

The findings suggest that the proposed teaching model contributed positively to students' learning performance and engineering competency development. By combining authentic project experiences with bilingual learning and AI-supported practice, the model created an engaging learning environment that encouraged active participation and practical knowledge application. The observed improvements in academic achievement, student perceptions, and engineering outputs are generally consistent with previous studies highlighting the benefits of project-based and technology-enhanced engineering education.

This study also provides preliminary evidence that integrating AI-assisted programming into bilingual engineering courses may support students' learning processes and reduce barriers encountered during software development activities. However, the findings should be interpreted with caution due to the single-institution setting and limited observation period. Further studies involving larger samples, multiple institutions, and longitudinal evaluation are needed to examine the generalizability and long-term effectiveness of the proposed framework.

6. Conclusion

This study developed and implemented a project-based bilingual teaching model for the course Mobile Development and Application within the NEE framework. By combining authentic software development projects, English-mediated learning activities, team-based implementation, and intelligent learning assistance, the framework sought to bridge theoretical learning and practical application. The findings suggest that the redesigned course supported students' academic development, participation in learning activities, and engagement in engineering practice. The experience gained from this reform provides useful insights for curriculum innovation in software-related disciplines and contributes to ongoing efforts to cultivate application-oriented engineering talent with international perspectives. Future research will extend the implementation to different educational contexts and further examine its sustained value over time.

Acknowledgments

This work was supported by the Teaching Reform Project of Sichuan University of Science & Engineering (Grant No. JG-24023 and JG-25019) and the Industry–University Collaborative Education Program of the Ministry of Education of China (Grant No. 250803175205823).

References

[1] Sun, L., Jiang, X., Ren, H., et al. (2020). Edge-cloud computing and artificial intelligence in internet of medical things: Architecture, technology and application. *IEEE Access*, 8, 101079–101092. https://doi.org/ 10.1109/ ACCESS. 2020. 2998472.

[2] Zein, S., Salleh, N., & Grundy, J. (2023). Systematic reviews in mobile app software engineering: A tertiary study. *Information and Software Technology*, 164, 107323. https:// doi. org/10.1016/j.infsof.2023.107323.

[3] Wei, L., & Zhang, W. (2025). Strategic initiatives for new engineering education in China. *International Journal of Engineering Education*, 41(2), 476–491.

[4] Yan, S., Meegahapola, L., & Chen, G. (2025). Integrating programming skills into engineering education in the AI-driven world. In *Proceedings of the 36th Australasian Association for Engineering Education Annual Conference* (pp. 543–550).

[5] Probert, G. J. (2024). Developing dual language skills and intercultural communication strategies in a bilingual learning environment: Investigating a project-based learning programme. *British Journal of Education*, 12(2), 63–81.

[6] Nurbekova, Z., Grinshkun, V., Aimicheva, G., et al. (2020). Project-based learning approach for teaching mobile application development using visualization technology. *International Journal of Emerging Technologies in Learning*, 15(8), 1–14. https://doi.org/10.3991/ijet.v15i08.13204.

[7] Sengul, C., Neykova, R., & Destefanis, G. (2024). Software engineering education in the era of conversational AI: Current trends and future directions. *Frontiers in Artificial Intelligence*, 7, 1436350. https://doi.org/10.3389/frai.2024.1436350.

[8] Lavado-Anguera, S., Velasco-Quintana, P. J., & Terrón-López, M. J. (2024). Project-based learning (PBL) as an experiential pedagogical methodology in engineering education: A review of the literature. *Education Sciences*, 14(6), 617. https:// doi. org/ 10.3390/educsci14060617.

[9] Mursid, R., Saragih, A. H., & Hartono, R. (2022). The effect of blended project-based learning model and creative thinking ability on engineering students' learning outcomes. *International Journal of Education in Mathematics, Science and Technology*, 10(1), 218–235.

[10] O'Connor, S., Power, J., Blom, N., et al. (2026). Problem and project-based learning (PBL) within an online engineering module: An examination of student teamwork satisfaction and attitudes. *European Journal of Engineering Education*, 51(2), 1–30. https://doi.org/10.1080/03043797.2025.2523411.

[11] Sukackė, V., Guerra, A. O. P. D. C., Ellinger, D., et al. (2022). Towards active evidence-based learning in engineering education: A systematic literature review of PBL, PjBL, and CBL. *Sustainability*, 14(21), 13955. https://doi. org/10. 3390/ su142113955.

[12] Servant-Miklos, V. F. C., & Kolmos, A. (2022). Student conceptions of problem and project-based learning in engineering education: A phenomenographic investigation. *Journal of Engineering Education*, 111(4), 792–812. https:// doi. org/ 10.1002/jee.20474.

[13] Mammadova, R. (2026). English terminology and its impact on information technology development. *Euro-Global Journal of Linguistics and Language Education*, 3(2), 34–41.

[14] Chan, J. Y. H., & Walsh, S. (2024). English learning and use in Hong Kong's bilingual education: Implications for L2 learners' development of interactional competence. *International Journal of Applied Linguistics*, 34(1), 183–205. https://doi.org/ 10. 1111/ijal.12524.

[15] Chaudhary, A. A. (2018). Enhancing academic achievement and language proficiency through bilingual education: A comprehensive study of elementary school students. *Educational Administration: Theory and Practice*, 24(4), 803–812.

[16] Ziegler, A., Kalliamvakou, E., Li, X. A., et al. (2024). Measuring GitHub Copilot's impact on productivity. *Communications of the ACM*, 67(3), 54–63. https://doi.org/ 10. 1145/ 3632208.

[17] Fan, G., Liu, D., Zhang, R., et al. (2025). The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: A comparative study with traditional pair programming and individual approaches. *International Journal of STEM Education*, 12(16), 1–17. https://doi.org/10.1186/s40596-025-00486-9.

[18] Appachikumar, A. K. (2023). Generative AI for software architecture design: Opportunities and pitfalls. *International Journal of Social Impact*, 8(4), 1–11.