

Evaluating the Integration of AI and IoT in Mobile App Education

Insights and Recommendations

Khaled M. Ben Hamed and Mohamed A. Mgheder

Mobile Computing Department
Faculty of Information Technology
University of Tripoli, Libya

{k.benhamed, m.mgheder}@uot.edu.ly

Abstract. The integration of Artificial Intelligence (AI), Internet of Things (IoT), and Ambient Computing (AC) is transforming Mobile Application Development (MAD), requiring new pedagogical approaches for undergraduate IT programs. This paper presents a 14-week intermediate MAD course for Mobile Computing (MC), Software Engineering (SE), and Web Technologies (WT) students. The course adopts a hybrid design with innovative pedagogy, evaluated for its effectiveness in fostering student engagement and real-world competency. Of 52 enrolled students (26 MC, 26 WT), 46 completed all assessments (88.5%). Using a quasi-experimental post-test-only design, analysis of six assessment types (mean = 62.89, SD = 18.78) found no significant program effect on pass/fail ($\chi^2(1, N = 46) = 0.26, p = .610$) or gender effect ($\chi^2(1, N = 46) = 0.48, p = .489$). Survey results revealed high satisfaction (94%), teamwork dominance (89%), AI adoption barriers (36%), and confidence disparities. Findings indicate the hybrid design accommodated heterogeneous preparation, with coursework predicting final exam performance ($\beta = .57, p < .001, R^2 = .40$) and early assignment performance serving as an early-warning indicator.

Keywords: Mobile App Development, AI Integration, IoT Integration, Ambient Computing, Pedagogical Strategies, Curriculum Design, Teaching Methodologies.

1 Introduction

MAD is a technology-driven field shaped by enabling technologies that are advancing rapidly. In particular, the incorporation of AI, IoT, AC has transformed mobile applications from relatively standalone tools into connected, data-rich ecosystems that improve user interaction and engagement [1]. As a result, these developments necessitate a concurrent transformation in MAD education, including updates to curriculum content, instructional approaches, and competency frameworks to align with emerging technological paradigms. Thus, educational institutions must adapt novel curricula and teaching methodologies to prepare students for a rapidly changing landscape [2].

<https://doi.org/10.5281/zenodo.23004077>

Although such adaptation is widely recognized as an important, many institutions globally especially in Arab countries, including Libya still tend to prioritize teaching activities over scientific research. This imbalance can limit the effective integration of cutting-edge technologies and practices into educational settings. Moreover, traditional pedagogical models often fail to equip learners with the interdisciplinary knowledge and practical skills needed to navigate the complexities of modern MAD [3]. Consequently, the role of educators evolves from merely disseminating knowledge to fostering innovative thinking and problem-solving abilities among students.

Despite growing enthusiasm for hybrid MAD instruction, the empirical literature has two notable weaknesses:

1. **Descriptive reporting dominates.** Many studies report percentages of students passing or achieving certain grades without statistical grounding — no chi-square, correlation, or significance testing [4], [5].
2. **Sample characteristics are often omitted.** Prior work frequently fails to state the number of registered versus completed students, demographic breakdowns, or attrition rates, making replication and generalization difficult [6].

This study addresses both gaps by providing a fully specified methodology and statistically grounded results. Specifically, we ask:

- **RQ1:** What are the sample characteristics and completion rates of students in the hybrid MAD course?
- **RQ2:** Is there a significant association between program (*MC* vs. *WT*), gender, or achievement band and pass/fail status?
- **RQ3:** What is the relationship among homework, midterm, project, assignment, final exam, and total scores?
- **RQ4:** Do *MC* and *WT* students differ significantly on any outcome?
- **RQ5:** Does coursework performance predict final exam performance, and do demographic variables add explanatory power?
- **RQ6:** How do students perceive the integration of *AI*, *IoT*, and *AC* technologies in *MAD* education, and what barriers do they report?

This paper presents a hybrid course design that integrates traditional and innovative pedagogical strategies to incorporate emerging technologies and to improve learning practices within educational settings. The course is tailored to intermediate-level undergraduate students enrolled in three Information Technology programs: *MC*, *SE*, and *WT*. Over a 14-week period, it combines theoretical foundations with practical *MAD* activities that incorporate *AI*, *IoT*, and *AC*, thereby reflecting the technologies' growing industrial relevance.

Overall, the course design was effective in integrating advanced technologies and practices into the learning environment. It also broadened students' interdisciplinary learning resources, and strengthened their competency for incorporation of *AI*, *IoT*, and *AC*, which positively supported skills development for most students, particularly those facing a steep learning curve.

2 Related Work

2.1 Evolution of Computing Education and Pedagogical Shifts

In the 1930s, Alan Turing, a foundational figure in computing, introduced the *Turing machine (TM)*, a theoretical model that defined computation and algorithms, shaping curricula focused on logic and mathematics, primarily delivered through lectures with limited hands-on experience [7]. Turing's codebreaking work aided in the development of early computers like the Colossus [8]. This real-world application began to influence curricula by integrating theoretical foundations with engineering practices, establishing key principles for teaching and curricular development in computer science and engineering.

Fig.1 illustrates how the evolution of computing since 1940 has driven major shifts in pedagogy and curriculum development. Each decade's technological and societal shifts have forced educators to transition from basic technical training to modern interdisciplinary models [9]. This dynamic transformation ensures that education keeps pace with the rapid technological environment, preparing students for future challenges.

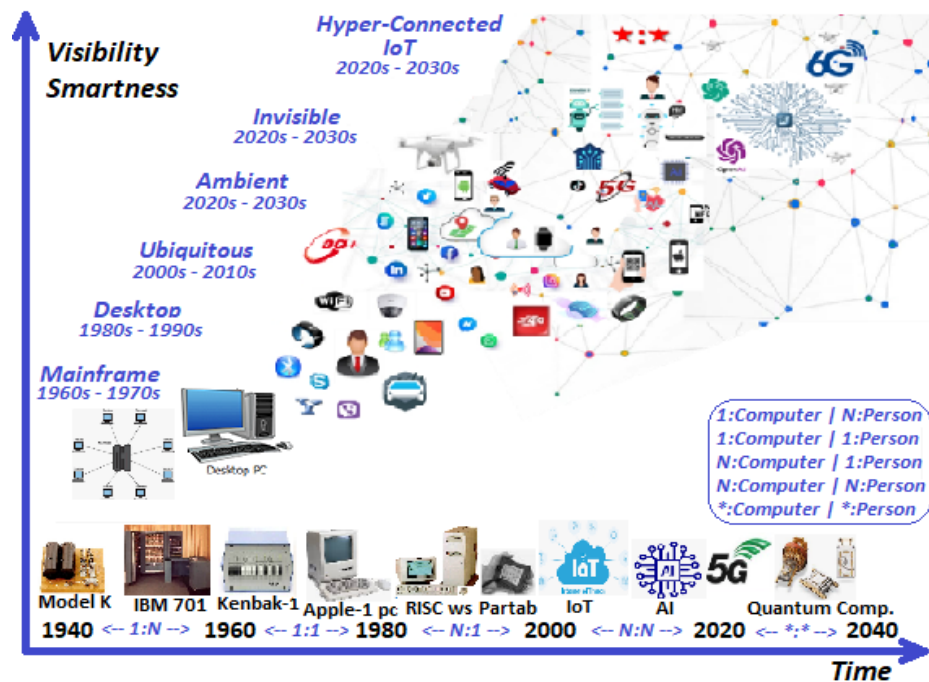


Fig. 1. Evolution of computing.

Table 1 summarizes, for each period, how pedagogy shifted from predominantly technical approaches toward more interdisciplinary approaches — that is, the direction of change in MAD education. "Technical" refers to teaching mainly how to write code and

understand computing systems; "interdisciplinary" refers to teaching students to combine computing with other fields (e.g., mathematics, engineering/UX, design, psychology, and sometimes social/ethical thinking).

Table 1. How pedagogy moved from technical toward interdisciplinary.

Period / Era	Computing and Societal Shifts	Pedagogy & Curriculum Evolution	Link to Interdisciplinary Models
1940–1960	Computing emerged as a foundational discipline; early emphasis on data processing and basic programming	Hands-on learning became central. Students engaged with computing systems through laboratory-based instruction focused on core technical skills [7]	Mostly technical at first, but practical system exposure began laying groundwork for later interdisciplinary approaches
1960–1980	Rapid technological advancement expanded the scope of computing	Curricula widened to include operating systems and networking . Instruction moved toward interactive workshops and collaborative projects . Programming was paired with mathematics for algorithms, and engineering principles were combined with design thinking for interfaces [11]	Clear transition from basic training toward integrating multiple disciplines (math + programming; engineering + UX/design)
Late 1980s–1990s	Ubiquitous computing and the conceptual foundation of pervasive, everyday technology	HCI and user-centered design gained emphasis. Curriculum increasingly merged computer science with design and psychology to support user-friendly mobile applications [12]. Collaborative, hands-on projects became essential, with added attention to social implications [13]	Strong interdisciplinary consolidation: CS + HCI + design/psychology, plus reflection on societal impact
Early 2000s	Mobile application prototypes became more feasible; IoT increased relevance	Instruction shifted toward project-based learning to solve real-world problems [14]. IoT supported rapid prototyping and iterative design cycles	Interdisciplinary emphasis strengthened via real-world problem framing (beyond “how to code” into “how to build”)
2010s	Mobile and connected technologies matured; digital learning models advanced	Flipped classrooms and blended learning gained traction, promoting active learning and critical analysis of societal impacts of mobile technologies [15]	Pedagogy became more reflective and integrative—technical learning combined with broader evaluation of technology’s effects

Today, MAD education continues to evolve, prioritizing experiential learning and interdisciplinary collaboration, with curricula increasingly focused on emerging technologies, ethical considerations, and societal impacts.

2.2 Integration of *AI*, *IoT*, and *AC* in *MAD* Education

The systematic literature review reveals a growing shift in integrating *AI*, *IoT*, and *AC* into *MAD* courses in higher education [16]. The sharp increase in publications highlights the key role these technologies play in supporting effective learning and interdisciplinary competence. Van and colleagues developed a pedagogical evaluation framework via bibliometric analysis of mobile learning literature, identifying nine effective mobile pedagogical types, including Situated Learning, Collaborative Learning, and Game-Based Learning. Their research also pinpointed two key inhibitors of effectiveness across all educational levels: the digital divide (disparities in skills, socioeconomic status, and resources) and a lack of integration, where instructional designs operate in isolation rather than being embedded into mainstream frameworks [17].

Sulemana applied the Technology Acceptance Model (*TAM*) and Technological Pedagogical Content Knowledge (*TPACK*) to highlight the critical interplay between pedagogical and technological factors in *MAD*, offering insights for effective methodologies, particularly in low- and middle-income nations [18]. Separately, Nurbekova and Aimicheva proposed a flexible hybrid curriculum architecture that integrates foundational mobile computing knowledge with programming skills across various embedded operating systems [19].

From a technology perspective, key developments include cloud-based platforms enabling real-time, global access to *MAD* learning. The integration of *AI* and machine learning enhances user engagement and personalization, with ongoing work leveraging generative *AI* tools like *ChatGPT* for context-sensitive support [20]. *IoT* has redefined mobile apps through device interoperability and real-time processing, requiring curricula to address device security and data sharing. A hands-on case study by Laszlo et al. bridges theory and practice using 3D-printed models and gesture recognition [21]. Finally, ambient computing integrates seamlessly into user environments; Corno, De Russis, and Bonino describe a project-based course on ambient intelligence that includes industry collaboration [22].

The academic consensus favors Project-Based Learning and the integration of *AI*, *IoT*, and *AC* to overcome the inflexibility of traditional teaching methods. Sales and colleagues combined Challenge-Based Learning (*CBL*), Scrum, and Lean Startup methodologies to mimic real-world environments, resulting in improved student engagement, higher quality applications, and the emergence of startups [23]. Another study by Andamari et al. merged pedagogy with engaging gameplay to bridge traditional and modern teaching [24].

2.3 Research Gap

In summary, the state of the art in *MAD* underscores the need for curricular transformation that embeds *AI*, *IoT*, and *AC* into undergraduate education. Understanding current trends and challenges allows educators to align teaching methodologies with industry demands, preparing a new generation of developers for a technology-driven *MAD*. However, as noted in Section 1, the empirical literature remains limited in two respects: statistical rigor and full sample specification. The present study addresses these gaps by

providing a fully specified methodology and statistically grounded results, complemented by survey-based perception data.

3 Hybrid Course Design

This section explains, in detail, the 14-week hybrid *MAD* course, which integrates *AI*, *IoT*, and *AC* to maximize student engagement and competency. Active learning (student role) and faculty facilitation/guidance are employed as innovative pedagogical strategies designed to bridge theory and practice. The course design evaluates how *AI*-driven, *IoT* solution, and *AC* technologies can be effectively incorporated into the curriculum through combination of pedagogical strategy, curriculum structure, and teaching methodology. The section provides insights into how the course design facilitates student engagement, knowledge acquisition, and practical skill development.

3.1 Pedagogical Approach

As shown in Fig. 2 this hybrid course design built on dual foundational elements (principals) constituting the of effective learning: Facilitation and guidance (faculty member role) and the active learning through passion and interest (student role).

3.1.1 Faculty Member's Role

The faculty member teaches the following:

- **Integrated Development Environment (IDE):** The *IDE* utilized is *Android Studio*. Students are instructed on the installation process and become familiar with its interface, including the *Gradle* build system. Through installation and exploration students gain practical experience that enhances understanding, boosts motivation, and strengthen self-efficacy in *MAD*.
- **Modern Programming Strategies:** The course integrates *DevOps* (e.g., *GitHub* for assignments), *CI/CD* for automated testing and deployment, Design Thinking for user-centric innovation, and Software Reuse via legacy code, libraries, and modular *APIs*. Agile methodologies complement these practices, enabling adaptation to evolving technologies like *AI*, *IoT*, and *AC*.
- **The Primary Application Components:** A mobile application developed in *Android* using *Java* (outlined in Table 2). The faculty member provided a comprehensive introduction to each of components, which collaboratively function to produce a cohesive and functional mobile application.

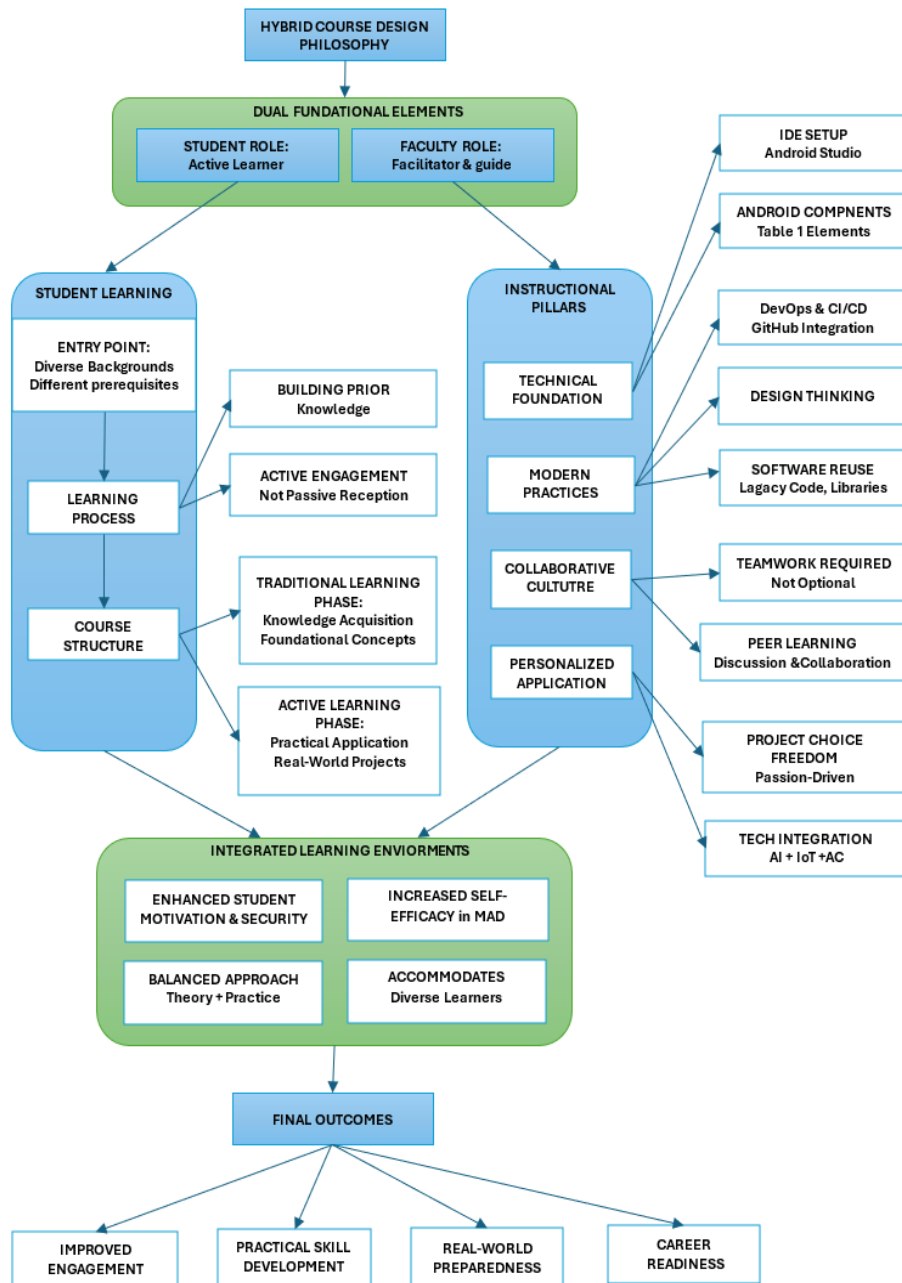


Fig. 2. Hybrid Course Design Philosophy.

Table 2. The primary application components.

No.	component	function
1	Activities	Represent a single screen with a user interface. Each activity is implemented as a subclass of the <i>Activity</i> class.
2	Intents	Messaging objects used to request an action or communicate between components (e.g., starting an <i>activity</i> or <i>service</i>).
3	Fragments	Modular sections of an activity that can be reused across multiple activities. <i>Fragments</i> help create dynamic and flexible user interfaces.
4	Services	Services are subclasses of the Service class. <i>Run</i> in the background to perform long-running operations without a user interface.
5	Content Providers:	Manage access to a central repository of data. They allow data sharing between applications and are implemented as subclasses of the <i>ContentProvider</i> class.
6	Broadcast Receivers	Respond to broadcast messages from other applications or the system. They are implemented as subclasses of the <i>BroadcastReceiver</i> class.
7	Views and ViewGroups	<i>UI</i> components that display content and structure the layout of the application. Views are individual UI elements, while <i>ViewGroups</i> are containers for organizing multiple views.
8	Resources:	Non-code assets such as strings, images, and layout files that are used to define the app's UI and behavior.

- **Teamwork:** The faculty member emphasizes collaboration, as writing all code alone is impractical given advances in *AI*, *IoT*, and *AC*. Students reuse existing code, libraries, and components from *Java*, *Android*, and third parties, while Gradle may auto-generate additional code.
- **Personalized Approach:** In the course design, the faculty member requires Students autonomously choose a project field aligned with their passions, developing a mobile app integrating *AI*, *IoT*, and *AC*. This real-world, project-based approach addresses skill gaps and produces measurable outcomes that prepare them for future courses and careers.

3.1.2 Student's Role

Students' roles are not passive receivers of knowledge; but instead, they are active learners whose behaviors, motivations, and interactions significantly influence learning outcomes. Students taking this course come from different related computing programs that result to different prerequisites courses. The hybrid design enables and accommodate students to build on their prior knowledge and experiences. Recognizing and accommodating diverse knowledge and experiences is accomplished by permitting teamwork and learning from peers through discussion and collaboration. For these reasons,

the course split into two parts, traditional and active learning. These two parts are intended to foster a more effective, diverse, and enriching learning environment.

3.2 IEEE/ACM Curriculum Outline

The curriculum is organized into distinct sections aligned with IEEE/ACM guidelines.

3.2.1 Foundational Programming Skills (Traditional Learning Phase)

This section adheres to the *IEEE/ACM* Curriculum 2013 guidelines for Android Java development, emphasizing two key knowledge areas (*KAs*): foundational concepts of Object-Oriented Programming (*OOP*) and an understanding of Android Application Architecture (*AAA*), both of which are essential for the development of robust mobile applications [26].

- ***KA in Core OOP Skills***: Focuses on the fundamental building blocks of *OOP*, including classes, interfaces, and abstract classes. Additionally, it covers essential *Java* processes pertinent to Android development, such as *Threads* and *Handlers*. These concepts are critical for managing background processing and updating the main activity. Upon completion, students can reaffirm their proficiency in *Java* programming.
- ***KA in Core AAA Skills***: Introduces the principles of *AAA*, encompassing the components listed in Table 2, which constitute the structural framework of Android applications. While *AAA* is addressed at a foundational level, students engage with it through ten assignments that provide hands-on programming experience with each of the main components, as well as utilizing *GitHub* for assignment submissions.

3.2.2 Assessment of Foundational Programming Skills

The assessment strategy evaluates students' knowledge of basic Android development skills, specifically their ability to decompose complex tasks into manageable activities, including writing code, utilizing an IDE, and debugging with GitHub. This assessment comprises ten programming assignments contributing 10% to the final grade, in addition to a mid-term examination worth 15%. The mid-term exam assesses students' understanding of *OOP* concepts within the Android development environment and their knowledge of *AAA*. Furthermore, this assessment measures students' acquisition of a solid foundation in the traditional learning phase, indicating their preparedness for more advanced topics.

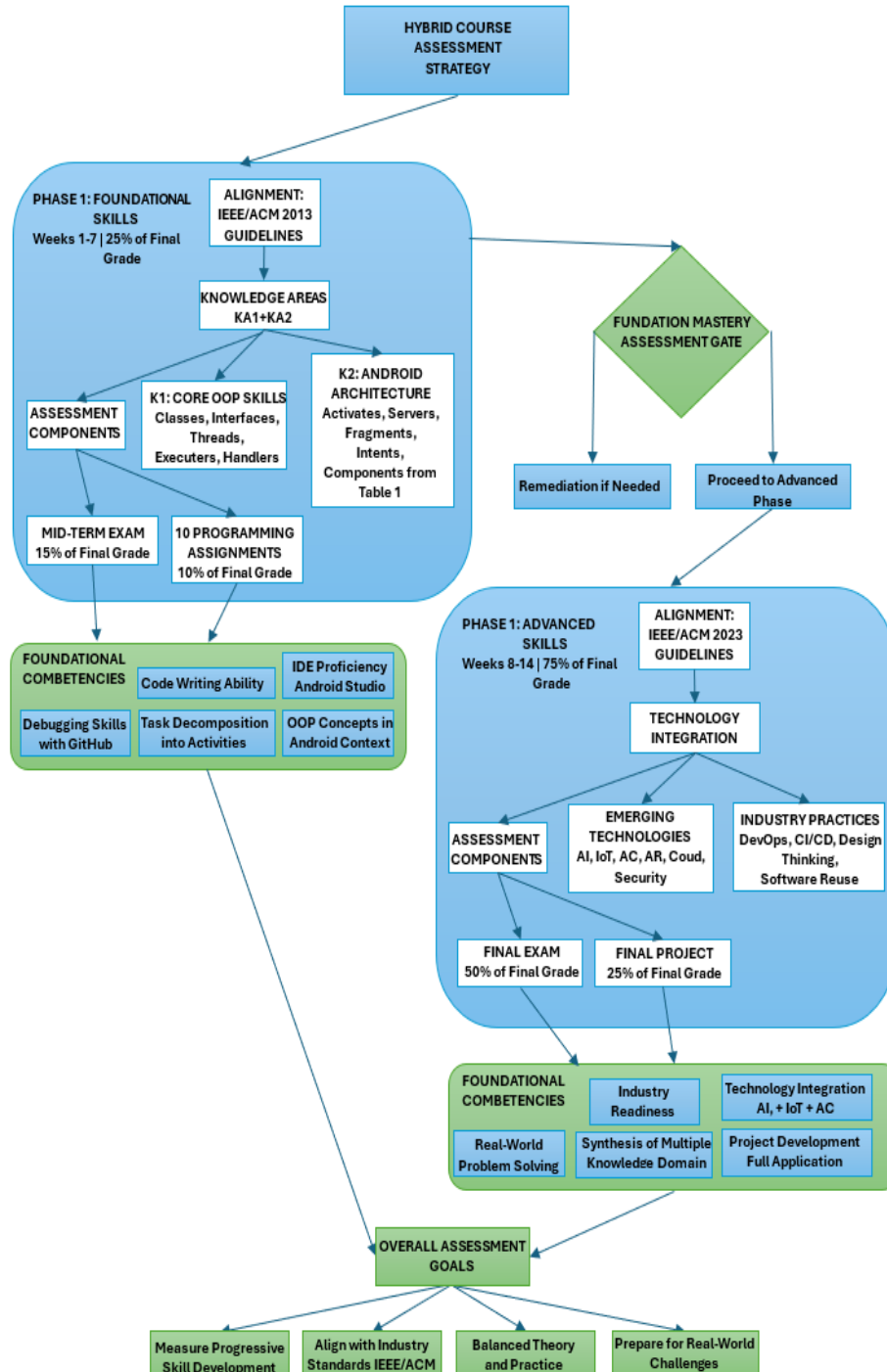


Fig. 3. Hybrid Course Assessment Strategy

3.2.3 Advance Programming Skills (Active Learning Phase)

This section aligns with the *IEEE/ACM Curriculum 2023* guidelines for *Android Java* development, highlighting the necessity for students to remain informed about advancements in *AI*, *IoT*, and *AC* [27]. It emphasizes the building on the knowledge gained from the previous phase of the course to integrate it with emerging technologies from various domains, reflecting the interconnectivity inherent in new technological applications. Students engage in hands-on course projects, either collaboratively in teams or independently. *Android* application projects focus on various emerging technologies across multiple domains, including *AI*, networking, cloud, augmented reality (*AR*), *IoT*, Security, *AC*, and *UI/UX*, utilizing packages of *Android*, *Java*, and third packages.

3.2.4 Assessment Advance Programming Skills:

The active learning phase assessment consists of a project worth 25% of the final grade, which focuses on developing a mobile app using *Java* and *Android Studio*, and a final exam worth 50%.

4 Methodology

4.1 Research Design

This study employed a quasi-experimental, post-test-only design with a correlational component to evaluate student learning outcomes in the *MAD* course delivered at the Faculty of Information Technology, University of Tripoli, during Spring 2024. The pedagogical model, curriculum phases, and instructional roles are presented in Section 4 and are not repeated here. This section reports only the empirical procedures used to collect, measure, and analyze data.

4.2 Context and Participants

The study was conducted within a 14-week undergraduate *MAD* course at the Faculty of Information Technology, University of Tripoli, in Spring 2024. The course targets intermediate-level undergraduate students. In this offering, no students from the *SE* program were enrolled; the sample therefore comprises students from two programs: *MC* and *WT*.

A total of 52 students registered at the start of the course (26 *MC*, 26 *WT*). Of these, 46 students completed all course assessments and were included in the final analysis (25 *MC*, 21 *WT*), yielding an overall completion rate of 88.5%.

4.3 Sample Characteristics

Sample characteristics for the 46 completers are reported in Table 3.

Table 3. Sample characteristics of completers (N = 46).

Characteristic	Category	MC (n = 25)	WT (n = 21)	Total (N = 46)
Gender	Male	15 (60.0%)	13 (61.9%)	28 (60.9%)
	Female	10 (40.0%)	8 (38.1%)	18 (39.1%)
Age (years)	Mean (SD)	21.72 (0.94)	23.10 (0.94)	22.35 (1.16)
	Range	21–24	21–25	21–25
Program	MC	25 (100%)	—	25 (54.3%)
	WT	—	21 (100%)	21 (45.7%)
	SE	0	0	0 (0%)
Characteristic	Category	MC (n = 25)	WT (n = 21)	Total (N = 46)
Gender	Male	15 (60.0%)	13 (61.9%)	28 (60.9%)

Table 4. Registration, completion, and attrition by program (Spring 2024).

Program	Registered	Completed	Retained	Attrition
MC	26	25	96.2%	3.8%
WT	26	21	80.8%	19.2%
Total	52	46	88.5%	11.5%

Note. Three MC records appeared in the completion list but not the registration roster; these were treated as late registrants and excluded from the matched analysis, leaving 25 matched MC completers. All 21 WT completers were matched to the WT registration roster.

Inclusion criteria: (a) intermediate-level undergraduate status; (b) enrollment in the MC or WT program; (c) completion of all course assessments (homework, midterm exam, project, final exam). Students who withdrew or failed to complete any assessment were excluded.

4.4 Variables and Measures

- **Independent variable:** participation in the 14-week hybrid *MAD* course. Between-subjects factor: program (*MC* vs. *WT*).
- **Dependent variables — continuous (0–100 scale):** Homework score (10%), Midterm exam score (15%), Project score (25%), Coursework score (50%) = homework + midterm + project, Final exam score (50%), Total course score (100%)
- **Dependent variables — categorical (for association testing):** Pass/fail (threshold = 50%) and achievement band (Excellent ≥ 85 , Very Good 75–84, Good 65–74, Satisfactory 50–64, Fail < 50).
- **Covariates:** program (*MC*, *WT*), gender, and age.

- **Instruments.** Homework was aggregated from ten programming assignments submitted via *GitHub*. The midterm and final exams were written and practical exams covering *OOP*, Android Application Architecture, and emerging-technology integration. The project was assessed with a departmental rubric covering functionality, *UI/UX*, code quality, and integration of *AI*, *IoT*, *AC*.

4.5 Data Collection Procedure

Data were collected at two points during the Spring 2024 semester:

- **Week 1 (pre-course):** Registration rosters were exported from the faculty's Student Information System, including student ID, name, age, gender, and program.
- **Week 14 (post-course):** Homework, midterm, project, assignments, final exam, and total scores were extracted from the official gradebook at the Faculty of Information Technology, University of Tripoli.

No pre-test or diagnostic scores were available; therefore, a within-subject pre-post comparison was not possible.

4.6 Data Analysis

All analyses were conducted in SPSS v.29, with significance set at $p < 0.05$. The following procedures were applied:

1. **Descriptive statistics** — means, standard deviations, frequencies, and percentages for all variables (Tables 1 and 2).
2. **Normality checks** — Shapiro-Wilk test and skewness/kurtosis to select parametric vs. non-parametric tests.
3. **Association testing (categorical outcomes):**
 - Chi-square test of independence for program $\times$ pass/fail, gender $\times$ pass/fail, and program $\times$ achievement band.
 - Cramér's V as the effect size for chi-square results.
4. **Correlation analysis:** Pearson correlation among homework, midterm, project, assignments, final exam, and total score.
5. **Group comparisons:** independent-samples t-test to compare MC vs. WT on each continuous outcome. Effect size: Cohen's d.
6. **Regression:** multiple linear regression predicting total course score from assignments, final exam, age, gender, and program.

4.7 Ethical Considerations

The study used anonymized, routinely collected course assessment data from the Faculty of Information Technology, University of Tripoli (Spring 2024). No personally identifiable information was disclosed, and all data were reported in aggregate. The study complied with the institution's data protection and research ethics guidelines.

5 Results

5.1 Descriptive Statistics

Table 5 reports descriptive statistics for all continuous outcomes by program.

Table 5. Descriptive statistics by program (N = 46).

Variable	MC (n = 25) M(SD)	WT (n = 21) M(SD)	Total (N = 46) M(SD)
Homework (/10)	8.16 (1.46)	7.86 (1.62)	8.02 (1.53)
Midterm (/15)	8.28 (3.96)	9.76 (3.22)	8.96 (3.70)
Project (/25)	18.68 (4.41)	17.90 (5.25)	18.33 (4.79)
Coursework (/50)	35.12 (8.02)	35.52 (8.80)	35.30 (8.33)
Final Exam (/50)	26.28 (13.79)	29.14 (11.37)	27.59 (12.74)
Total (/100)	61.40 (19.83)	64.67 (17.62)	62.89 (18.78)

5.2 Chi-Square Analyses

Table 6. Program × Pass/Fail

	Pass	Fail	Total	Total (N = 46)
MC	20 (80.0%)	5 (20.0%)	25	28 (60.9%)
WT	18 (85.7%)	3 (14.3%)	21	18 (39.1%)
Total	38	8	46	22.35 (1.16)

$\chi^2(1, N = 46) = 0.26, p = .610$, Cramér's V = .08 (not significant)

Table 7. Gender × Pass/Fail

	Pass	Fail	Total	
Male	24 (85.7%)	4 (14.3%)	28	Male
Female	14 (77.8%)	4 (22.2%)	18	Female
Total	38	8	46	Total

$\chi^2(1, N = 46) = 0.48, p = .489$, Cramér's V = .10 (not significant)

Table 8. Program × Achievement Band

Band	MC	WT	Total
Excellent (≥ 85)	4	2	6
Very Good (75–84)	3	5	8
Good (65–74)	5	6	11
Satisfactory (50–64)	8	5	13
Fail (< 50)	5	3	8
Total	25	21	46

$\chi^2(4, N = 46) = 1.31, p = .860$, Cramér's V = .17 (not significant)

5.3 Correlation Analysis

Table 6 reports Pearson correlations among all continuous outcomes.

Table 9. Pearson correlation matrix (N = 46).

	HW	MT	PJ	AS	FE
Homework (HW)	1.00				
Midterm (MT)	.22	1.00			
Project (PJ)	.39**	.37*	1.00		
Coursework (AS)	.58***	.76***	.78***	1.00	
Final Exam (FE)	.26	.43**	.48**	.52***	1.00
Total (TOT)	.46**	.67***	.73***	.84***	.89***

Note. * $p < .05$, ** $p < .01$, *** $p < .001$.

Key finding. The final exam showed a moderate, significant positive correlation with Coursework ($r = .52$, $p < .001$), and the total score correlated strongly with both Coursework ($r = .84$, $p < .001$) and final exam ($r = .89$, $p < .001$).

5.4 Independent-Samples t-Tests: MC vs. WT

Table 10. Independent-samples t-tests comparing MC and WT students.

Variable	MC M (SD)	WT M (SD)	t(44)	p	Cohen's d
Homework	8.16 (1.46)	7.86 (1.62)	0.66	.513	0.20
Midterm	8.28 (3.96)	9.76 (3.22)	-1.36	.181	0.40
Project	18.68 (4.41)	17.90 (5.25)	0.54	.590	0.16
Coursework	35.12 (8.02)	35.52 (8.80)	-0.16	.873	0.05
Final Exam	26.28 (13.79)	29.14 (11.37)	-0.76	.450	0.23
Total	61.40 (19.83)	64.67 (17.62)	-0.59	.558	0.18

No statistically significant differences were found between MC and WT students on any outcome. All effect sizes were small.

5.5 Predicting Final Exam Performance

To test whether continuous assessment predicts later performance, a multiple linear regression was conducted with final exam score as the outcome and coursework, homework, midterm, age, and gender as predictors (Table 11). Program (*MC* vs. *WT*) was excluded because preliminary analyses (Section 5.4) found no significant between-program differences. This specification avoids circularity because the outcome is not a component of any predictor.

Table 11. Multiple linear regression predicting final exam score (N = 46).

Constant	22.05	25.82	—	0.85	.398
Coursework	0.87	0.22	.57	3.99	< .001
Homework	−1.05	1.24	−.13	−0.85	.401
Midterm	−0.08	0.53	−.02	−0.15	.879
Age	−0.56	1.55	−.05	−0.36	.719
Gender (Female)	−0.31	3.52	−.01	−0.09	.931

$R^2 = .396$, Adjusted $R^2 = .321$, $F(5, 40) = 5.25$, $p = .001$. All *VIF* values < 5, indicating no multicollinearity.

The model was statistically significant, $F(5, 40) = 5.25$, $p = .001$, explaining 39.6% of variance in final exam performance. Coursework was the only significant predictor ($\beta = .57$, $p < .001$), indicating that students who performed well on continuous assessments also performed well on the final exam. Homework, midterm, age, and gender were non-significant (*all* $p > .39$). These results support the use of early coursework performance as an early-warning indicator, as students at risk of failing the final exam can be identified well before the examination.

5.6 Survey Results: Satisfaction, Learning Outcomes, and Challenges

This section presents the quantitative and qualitative findings from the course evaluation survey, analyzed to assess student satisfaction, learning outcomes, challenges, and the impact of integration of cutting-edge technologies such as *AI*-driven, *IoT* solution, and *AC* technologies in *MAD* education. **Satisfaction and perceived learning outcomes.** As shown in Fig.4, **94% of respondents reported satisfaction** (58% very satisfied, 36% somewhat satisfied), while no students expressed dissatisfaction. This aligns with the finding that **75% of students confirmed achieving their learning outcomes**, though **25% remained uncertain** (*Maybe*). The pedagogical effectiveness of lectures and assignments of the traditional learning phase was notable, with **86% attributing their understanding of mobile app components to these structured activities** (Table 2).

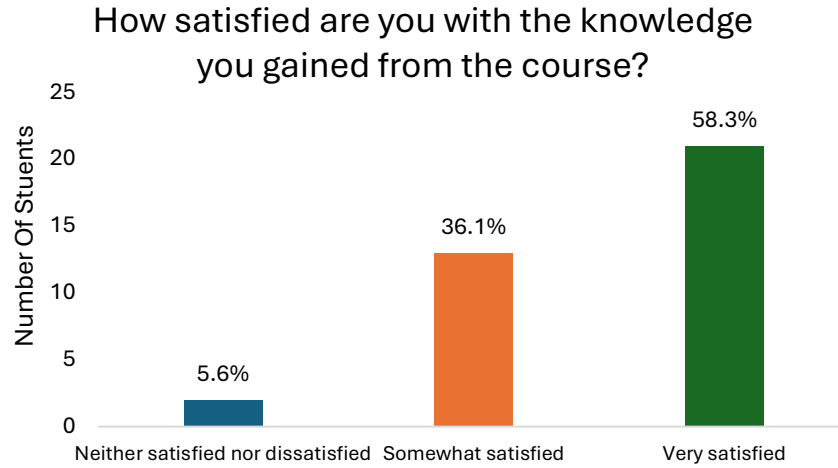


Fig. 4. Student satisfaction distribution.

Post-course readiness. Post-course readiness varied across respondents. Specifically, **50% reported confidence in developing smart mobile applications**, 47% indicated uncertainty, and 3% reported an inability to do so. Regarding future intentions, 31% planned to reuse the *Android* packages, 28% indicated that they would not, and 42% remained undecided.

Skills mastery and improvement. Regarding skills mastery via the programming pattern, 47% of respondents reported complete mastery, whereas 53% indicated partial mastery. None reported failure to learn. With respect to improvement in programming skills, 81% of respondents reported improvement, 19% were unsure, and none disagreed.

Team-based learning. Team-based learning was predominant, with **89% of students working in teams** for the course project (**Fig. 5**). Furthermore, **83% of *Android***

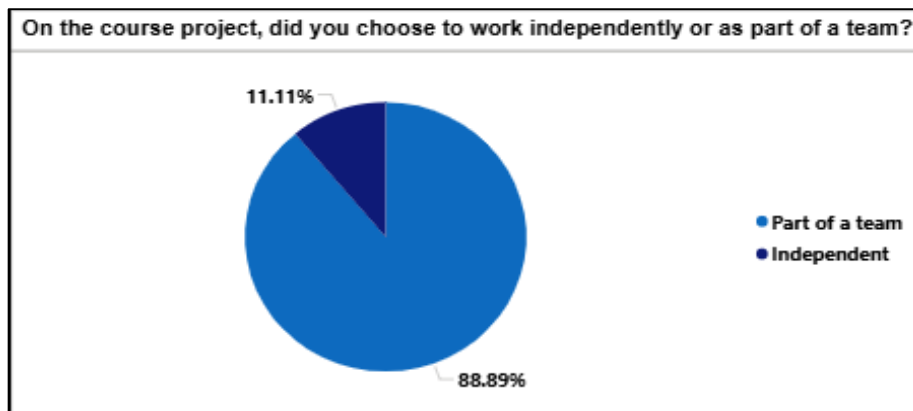


Fig. 5. Team vs. independent work preferences.

package selections were team-driven, while only 11% of students opted for independent work.

Regarding package selection, only 17% of respondents chose the *Android* package individually, whereas 83% reported adopting it through a team decision.

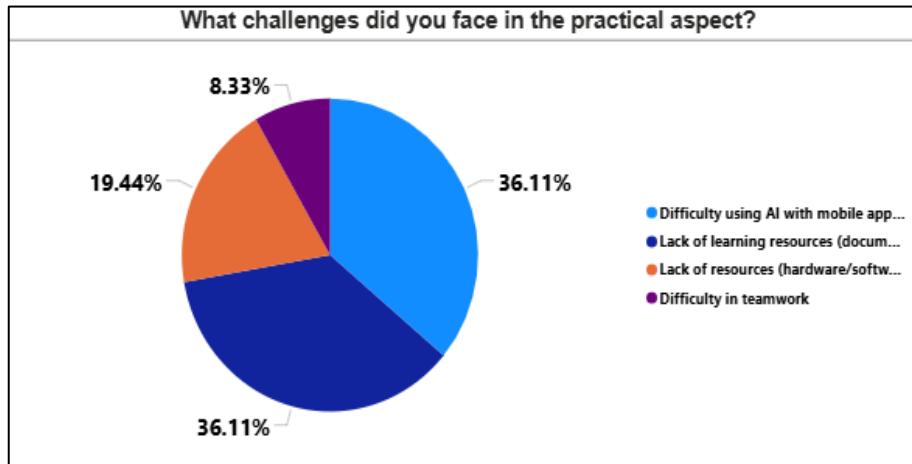


Fig. 6. Distribution of reported practical challenges.

Challenges encountered. The most common challenges were difficulty integrating *AI* (36%) and lack of interdisciplinary resources (36%). Other issues included hardware/software constraints (19%) and teamwork issues (8%) (Fig. 6).

AI integration challenges were primarily attributed to complexity (36%) and lack of documentation (22%) (Fig. 7).

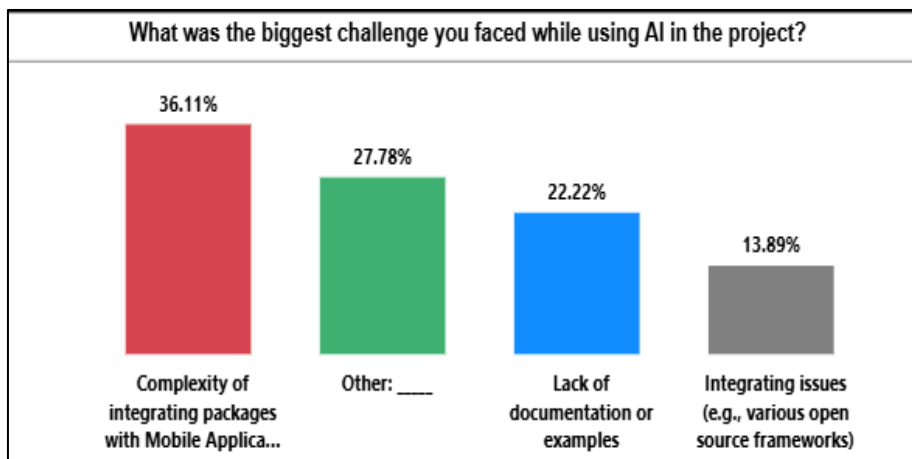


Fig. 7. Challenges in *AI* integration.

Impact of AI on project quality and skills. Regarding the of *AI*'s impact on project quality (Active Learning Phase), 78% of respondents reported a significant improvement in quality and efficiency, 19% reported a moderate improvement, and only 3% indicated no impact.

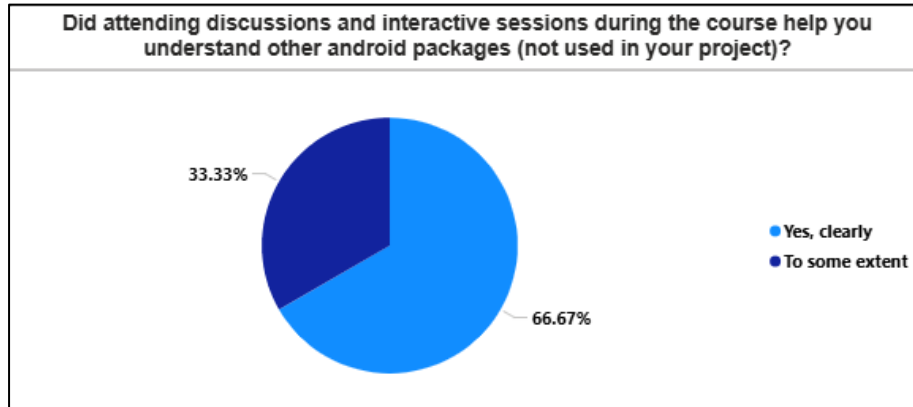


Fig. 8. Impact of interactive sessions.

These results suggest that *AI* plays a key role in enhancing overall project performance. With respect to *AI*'s impact on *MAD* skills, 89% reported improvement, while 6% were unsure and 6% disagreed. For **programming pattern understanding**, 75% of respondents agreed that *AI* improved their understanding of programming concepts, while 25% expressed uncertainty and none disagreed.

Interactive discussions. Interactive discussions were highly effective, with 67% of students reporting clear understanding of non-project Android packages, and 33% benefiting partially (Fig. 8).

Most respondents (86%) agreed that lectures and assignments effectively enhanced their knowledge, while 14% were uncertain and none negatively.

5.7 Summary of Results

Key Findings:

1. **High satisfaction (94%)** with knowledge gain, though **25% uncertainty** on learning outcomes.
2. **Teamwork dominance (89%)**, with limited individual assessment.
3. **AI enhances skills** but faces adoption barriers (36% integration challenges).
4. **Moderate learning curves** necessitate targeted support.
5. **Confidence disparities** suggest the need for advanced training opportunities.

6 Discussion

6.1 Summary of Principal Findings

This study evaluated student learning outcomes in a 14-week hybrid MAD course at the University of Tripoli, integrating both performance-based statistics and survey-based perceptions. Four principal findings emerged, organized by research question.

Completion and performance (RQ1). The hybrid course achieved a high completion rate: 88.5% overall (96.2% MC, 80.8% WT), with a mean total score of 62.89 (SD = 18.78) across 46 completers. Grade distribution showed that 26.1% of students achieved Very Good or Excellent, while 17.4% failed.

Program and gender equivalence (RQ2, RQ4). No statistically significant differences were found between MC and WT students on any outcome — homework, midterm, project, coursework, final exam, or total score — despite the two programs having different prerequisite coursework and a mean age gap of approximately 1.4 years. All effect sizes were small (Cohen's d ranged from 0.05 to 0.40). Chi-square tests similarly found no significant association between program and pass/fail ($p = .610$), gender and pass/fail ($p = .489$), or program and achievement band ($p = .860$).

Assessment relationships and predictors (RQ3, RQ5). Coursework correlated strongly with total score ($r = .84, p < .001$) and moderately with final exam ($r = .52, p < .001$), supporting the construct validity of the continuous-assessment scheme. A non-circular regression predicting final exam performance from coursework and covariates was statistically significant, $F(5, 40) = 5.25, p = .001$, explaining 39.6% of variance. Coursework was the sole significant predictor ($\beta = .57, p < .001$), while homework, midterm, age, and gender were non-significant. This convergence of objective and perceived measures strengthens confidence in the assessment design and supports the practical recommendation that early coursework performance serve as an early-warning indicator.

Student perceptions (RQ6). Survey results revealed that AI was widely recognized as transformative (89% reported MAD skill enhancement; 78% reported quality and efficiency gains), yet 36% encountered integration barriers due to complexity and insufficient documentation. Teamwork dominated project work (89%), with only 11% opting for independent work. Confidence disparities (50% confident vs. 47% uncertain about developing smart applications) and moderate learning curves point to the value of tiered support structures.

6.2 Interpretation in Light of the Hybrid Design

The absence of program differences supports the design principle described in Section 3.1.2: allowing students to autonomously choose a project domain aligned with their passions may compensate for differences in formal preparation. The strong correlation between assignment performance and final exam performance further supports the alignment of the assessment strategy with the hybrid design — because coursework spanned both the traditional and active learning phases, the assignment score functioned as an integrated measure of foundational and advanced skills. The survey data enrich

this interpretation: students attributed their understanding of mobile app components primarily to structured lectures and assignments (86%), while crediting AI tools with enhancing programming quality (78%) and skill development (89%). This suggests the hybrid model successfully bridged traditional instruction and technology-enhanced active learning.

6.3 Comparison with Prior Work

The finding of no program differences aligns with Van et al. [14], who reported that well-integrated mobile learning designs reduce performance gaps across educational levels, and with Nurbekova and Aimicheva [16], whose flexible hybrid curriculum architecture was designed precisely to accommodate heterogeneous preparation. However, it contrasts with studies that found persistent program differences in computing education, suggesting that the outcome may depend on the degree of scaffolding provided.

The *AI* adoption barriers observed here (36%) mirror the "lack of integration" inhibitor identified by Van et al. [14] and echo Sulemana's [15] emphasis on the interplay between pedagogical and technological factors. The finding that teamwork dominated individual assessment (89% vs. 11%) is consistent with Sales et al. [20], who reported improved engagement through CBL and Scrum but also noted the challenge of ensuring individual accountability in team-based projects.

The strong predictive role of continuous assessment supports the constructivist and experiential-learning orientation advocated by Sánchez-García and Reyes-de-Cózar [11] for project-based learning. Finally, the moderate learning curves reported around IDE setup, Gradle configuration, and debugging suggest that the "digital divide" inhibitor identified by Van et al. [14] may manifest even within well-resourced cohorts as a skills-access gap.

6.4 Practical Implications

1. **Hybrid designs can serve heterogeneous cohorts.** Institutions with multiple IT programs can adopt a single hybrid MAD course without disadvantaging students from any program.
2. **Continuous assessment predicts final performance.** Early assignment performance can be used as an early-warning indicator to identify students at risk of failing.
3. **Gender and age were not significant predictors,** supporting the equity of the hybrid design across demographic subgroups.
4. **AI adoption barriers require structured support.** With 36% of students reporting integration challenges, dedicated AI onboarding labs, expanded documentation, and industry partnerships are warranted.
5. **Balance team and individual work.** Given teamwork dominance (89%) and limited individual assessment, complementary individual tasks should be introduced to ensure full competency evaluation.

6. **Address confidence disparities.** Tiered learning paths and mentorship programs can support the 47% of students reporting uncertainty about developing smart applications.

6.5 Methodological Note on Prediction

An earlier specification regressed total course score on coursework and final exam. Because total score is the weighted sum of these two components, that model was circular and its high R^2 reflected arithmetic identity rather than predictive power. The revised model reported in Section 5.5 addresses this by predicting final exam score — an outcome not contained in any predictor. Coursework explained 39.6% of variance in final exam performance ($\beta = .57$, $p < .001$), a moderate and interpretable effect that supports the practical recommendation that early assignment performance serve as an early-warning indicator. Establishing whether these scores forecast longer-term outcomes, such as subsequent course performance or industry readiness, remains a direction for future longitudinal work.

7 Limitations and Future Work

7.1 Limitations

1. **No pre-test data.** Improvement claims must be made cautiously; findings describe post-course performance, not gains attributable to the intervention. *Impact:* The non-significant program difference may reflect ceiling effects rather than design effectiveness. *Future work:* A pre-post design with a control group would isolate the intervention effect.
2. **No SE students enrolled.** Generalizability is limited to two programs (MC and WT). *Impact:* Conclusions about the hybrid design's capacity to accommodate three programs remain untested. *Future work:* Replication in a semester with SE enrollment.
3. **Single institution, single semester.** Results may not generalize to other contexts. *Impact:* External validity is constrained. *Future work:* Multi-institutional replication across diverse educational settings.
4. **Excluded MC records.** Three MC records were excluded due to inconsistent registration data, reducing the MC sample from 28 to 25. *Impact:* A small proportion of the MC cohort is unrepresented; the exclusion was conservative. *Future work:* Improved data linkage between registration and gradebook systems.
5. **Self-reported survey data.** Survey responses may be subject to response bias, and the instrument is not fully described in terms of item count, scale, or reliability. *Impact:* Survey findings should be treated as indicative rather than definitive. *Future work:* Validation of the survey instrument with reported Cronbach's α and factor structure.

7.2 Future Work

Future iterations of hybrid *MAD* courses should provide structured AI integration guides, expanded *IoT* and *AC* resource libraries, and tiered learning pathways that accommodate varying levels of prior preparation. Courses should also balance collaborative and individual assessments — for example, by complementing the existing project weighting with individual coding assessments or oral defenses — and strengthen industry collaboration through guest lectures, internship pipelines, and industry-sponsored capstone projects. Longitudinal tracking of graduates into industry roles would further test the design's real-world effectiveness.

8 Conclusion

MAD pedagogical strategies and curricula must shift toward adaptability, collaboration, and real-world relevance, with sustained emphasis on *AI*, *IoT*, and *AC* delivered through project-based learning. This shift is essential not only to enhance student engagement and competency for real-world *MAD* practice, but also to foster innovation and align computing education with evolving industry needs.

The hybrid *MAD* course examined in this study responds directly to this imperative by integrating traditional instruction with active, project-driven learning aligned with the IEEE/ACM 2013 and 2023 curricular guidelines. The course achieved a high completion rate of 88.5% (96.2% *MC*, 80.8% *WT*), and statistical analysis showed no significant program or gender differences on any outcome — homework, midterm, project, assignments, final exam, or total score (all $p > .05$; Cohen's $d = 0.05$ – 0.40). The continuous assessment scheme demonstrated strong internal validity: assignment scores correlated strongly with total scores, $r(44) = .84$, $p < .001$, and a non-circular regression showed that coursework predicted final exam performance, explaining 39.6% of variance, $F(5, 40) = 5.25$, $p = .001$. These results confirm that the hybrid design effectively prepared students across heterogeneous backgrounds for advanced work in mobile application development.

At the same time, the study identified three challenges that qualify these positive findings. First, uncertainty around outcomes and confidence gaps remain — although aggregate performance was satisfactory, many students reported uncertainty about their ability to independently develop *MAD* solutions after the course. Second, *AI* tool adoption was uneven: while *AI* tools enhanced skill development for some students, many struggled to adopt them effectively, suggesting that tool availability alone is insufficient. Third, teamwork dominated individual assessment, and the heavy weighting of collaborative project work reduced opportunities for individual accountability, calling for a more balanced evaluation strategy. Moderate learning curves — particularly around *IDE* setup, Gradle configuration, and debugging — further indicate that targeted scaffolding is needed during the foundational phase.

By implementing structured *AI* guides, expanded *IoT/AC* resources, tiered pathways, balanced assessments, and industry partnerships, *MAD* educators can better prepare students to develop innovative mobile applications that tackle complex real-world chal-

lenges. The statistical evidence — chi-square, correlation, t-tests, and regression — supports the conclusion that the hybrid design effectively prepared students across heterogeneous backgrounds, while the survey evidence identifies concrete gaps that must be addressed in future iterations.

References

1. S. Rana, L. Gopal, and N. Gupta, "Smart City Concepts, Features and the Role of Internet of Things: A Review," 2021 International Conference on Computational Performance Evaluation (ComPE), Shillong, India, 2021, pp. 585-591, doi: 10.1109/ComPE53109.2021.9752040.
2. Z. Liang, K. Morimura, and YA Hernandez. "W-01 Beyond Writing Code: A New Pedagogical Model for Teaching App Development." In JSEE Annual Conference International Session Proceedings 2018 JSEE Annual Conference, Japanese Society for Engineering Education, 2018, pp. 10-15, doi.org/10.20549/jseer.2018.0_10.
3. F. McCown, and S. Gibson, "Using interdisciplinary teams in a mobile application development course," Journal of Computing Sciences in Colleges, 2013, 28. 85-93.
4. Field, "Discovering statistics using IBM SPSS statistics," Vol. 5. London: sage, 2018.
5. V. Braun and V. Clarke, "Using thematic analysis in psychology," Qualitative research in psychology 3.2, pp. 77-101, 2006, <http://dx.doi.org/10.1191/1478088706qp063oa>.
6. J. Bishop, and M. A. Verleger. "The flipped classroom: A survey of the research," 2013 ASEE annual conference & exposition, 2013, <https://doi.org/10.18260/1-2--22585>.
7. Turing, "On Computable Numbers, with an Application to the Entscheidungs problem." Proceedings of the London Mathematical Society, Series 2, 42 (1), 230–265. A follow-up correction was published in 1937: Proceedings of the London Mathematical Society, Series 2, 43 (1), 544–546.DOI: 10.1112/plms/s2-42.1.230.
8. BJ. Copeland, "Colossus: The secrets of Bletchley Park's code-breaking computers," Oxford, 2006, <https://doi.org/10.1093/oso/9780192840554.002.0003>.
9. S. Venkateswaran. 2022. "Reflections on the Evolution of Computer Science Education". ACM SIGSOFT Software Engineering Notes, Volume 47, Issue 3, 2022, pp.11–13. <https://doi.org/10.1145/3539814.3539817>.
10. J. F. Brennan. *The IBM Watson Laboratory at Columbia University: A History*. IBM, 1971.
11. Academic Programs in Computer Science. "Curriculum 68: Recommendations for academic programs in computer science", Communications of the ACM, Volume 11, Issue 3, pp. 151 – 197, 1968, <https://doi.org/10.1145/362929.362976>.
12. Y. Lester, "Advancing the Multidisciplinary Nature of Human Computer Interaction in a Newly Developed Undergraduate Course," First International Conference on Advances in Computer-Human Interaction, Sainte Luce, Martinique, France, 2008, pp. 177-182, doi: 10.1109/ACHI.2008.35.
13. X. Che, B. Ip and K. Bolka, "Collaborative Learning and Information Dissemination Based on Social and Mobile Computing," 2023 International Conference on Information Network and Computer Communications (INCC), Beijing, China, 2023, pp. 1-7, doi: 10.1109/INCC58754.2023.00008.
14. R. Sánchez-García, and S. Reyes-de-Cózar, "Enhancing Project-Based Learning: A Framework for Optimizing Structural Design and Implementation—A Systematic Review with a Sustainable Focus" *Sustainability* 17, 2025, no. 11: 4978, <https://doi.org/10.3390/su17114978>.

15. Frohber, C. Göth, G. Schwabe, "Mobile Learning Projects - a critical analysis of the state of the art. J. Comp. Assisted Learning," 2009, 25, pp.307-331, doi:10.5167/uzh-25270.
16. S. Hijazi, "AI-and IoT-Driven Mobile Learning Environments: A Framework for Educational Transformation," Workforce Development, and Digital Inclusion. International Journal of Interactive Mobile Technologies (IJIM), 19, 2025, doi:10.3991/ijim.v19i24.58959.
17. N. Van, S. Yaghmour, T. Son, S. Trinh, N. Hoa, " Pedagogical Design Patterns in Educational Mobile Apps: A Comparative Analysis of Learning Effectiveness across K-12, Higher Education, and Professional Training, " International Journal of Interactive Mobile Technologies," International Journal of Interactive Mobile Technologies (IJIM), 2025, 19. 4-21, doi:10.3991/ijim.v19i22.58511.
18. Sulemana, "The use of Handheld Devices to Enhance Teaching and Learning in Nursing and Midwifery Education: Research-in-Progress Paper," 2024 3rd Zimbabwe Conference of Information and Communication Technologies (ZCICT), Bulawayo, Zimbabwe, 2024, pp. 1-3, doi: 10.1109/ZCICT63770.2024.10958312.
19. Z. Nurbekova and G. Aimicheva, "Teaching Mobile Application Development: from the Idea to the Result," 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, Bosnia and Herzegovina, 2018, pp. 666-669, doi: 10.1109/UBMK.2018.8566488.
20. P. Lauren and P. Watta, "Work-in-Progress: Integrating Generative AI with Evidence-based Learning Strategies in Computer Science and Engineering Education," 2023 IEEE Frontiers in Education Conference (FIE), College Station, TX, USA, 2023, pp. 1-5, doi: 10.1109/FIE58773.2023.10342970.
21. R. Laszlo, R. Holonec, L. Grindei, B. Adrian and N. András, "Integrating IoT and AI Technologies in Engineering Education: A Smart Home Control Case Study," 2025 34th Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEIE), Cluj-Napoca, Romania, 2025, pp. 1-6, doi: 10.1109/EAEIE65428.2025.11136225.
22. F. Corno, L. De Russis and D. Bonino, "Educating Internet of Things Professionals: The Ambient Intelligence Course," in IT Professional, vol. 18, no. 6, pp. 50-57, Nov.-Dec. 2016, doi: 10.1109/MITP.2016.100.
23. Sales, N. Nascimento, R. Chanin and A. de Campos, "Agile Mobile Application Development: A Multidisciplinary Approach with CBL, Scrum and Lean Startup," 2025 IEEE Frontiers in Education Conference (FIE), Nashville, TN, USA, 2025, pp. 1-9, doi: 10.1109/FIE63693.2025.11328200.
24. S. A. Andamari, B. Yong, R. Y. Jie Lee and Y. Yanfi, "Mobile Educational Game "Imuno" to Teach Human Immune System," 2023 6th International Conference of Computer and Informatics Engineering (IC2IE), Lombok, Indonesia, 2023, pp. 250-255, doi: 10.1109/IC2IE60547.2023.10331286.
25. Lunt, et al. "Curriculum guidelines for undergraduate degree programs in information technology," ACM, 2008, doi: <https://doi.org/10.1145/2593311>.
26. M. Sahami and S. Roach, "Computer science curricula 2013," Commun. ACM 57, 6, USA, June 2014, <https://doi.org/10.1145/2610445>
27. N. Amruth, et al, "Computer Science Curricula 2023," ACM, New York, NY, USA, 2024, <https://doi.org/10.1145/3664191>

تقييم تكامل تقنيات الذكاء الاصطناعي وإنترنت الأشياء في تعليم تطوير تطبيقات الأجهزة المتنقلة: رؤى وتوصيات

خالد محمد بن حامد ومحمد احمد مغيدر

كلية تقنية المعلومات، جامعة طرابلس، طرابلس، ليبيا

{k.benhamed, m.mgheder}@uot.edu.ly

المستخلص: يشهد مجال تطوير تطبيقات الأجهزة المتنقلة (Mobile Application Development - MAD) تحولاً متسارعاً نتيجة التكامل المتزايد بين تقنيات الذكاء الاصطناعي (Artificial Intelligence - AI)، وإنترنت الأشياء (Internet of Things - IoT)، والحوسبة المحيطة (Ambient Computing - AC)، الأمر الذي يفرض الحاجة إلى تطوير المناهج الدراسية واعتماد ممارسات تربوية وأساليب تدريسية حديثة في برامج تقنية المعلومات على مستوى التعليم الجامعي. وفي هذا السياق، تقدم هذه الدراسة مقرر دراسياً متوسط المستوى في تطوير تطبيقات الأجهزة المتنقلة يمتد على مدى أربعة عشر أسبوعاً، ويستهدف طلبة الحوسبة المتنقلة وهندسة البرمجيات وتقنيات الويب.

يعتمد المقرر على تصميم تعليمي هجين يجمع بين أساليب التعلم التقليدية والتعلم النشط، مدعوماً باستراتيجيات تدريس مبتكرة تهدف إلى تعزيز مشاركة الطلبة وتنمية كفاءاتهم المهنية والتقنية بما يتوافق مع متطلبات تطوير تطبيقات الأجهزة المتنقلة في البيئات الواقعية. وقد شملت الدراسة 52 طالباً مسجلاً في المقرر، منهم 26 طالباً من تخصص الحوسبة المتنقلة و26 طالباً من تخصص تقنيات الويب، حيث أكمل 46 طالباً جميع متطلبات التقييم بنسبة إنجاز بلغت 88.5%. واعتمدت الدراسة منهجاً شبه تجريبي قائماً على الاختبار البعدي فقط، وتم تحليل نتائج ستة أنواع من أدوات التقييم، بمتوسط عام بلغ 62.89 وانحراف معياري قدره 18.78. وأظهرت النتائج عدم وجود فروق ذات دلالة إحصائية بين التخصصات فيما يتعلق بحالتي النجاح والرسوب $(\chi^2(1, N = 46) = 0.610, p = 0.489)$ ، كما لم تظهر فروق ذات دلالة إحصائية تعزى إلى الجنس $(\chi^2(1, N = 46) = 0.489, p = 0.489)$.

كما كشفت نتائج الاستبانة عن مستويات مرتفعة من رضا الطلبة عن المقرر بلغت 94%، وهيمنة واضحة للعمل الجماعي في تنفيذ المشروعات بنسبة 89%، إضافة إلى وجود تحديات مرتبطة بتبني تقنيات الذكاء الاصطناعي بنسبة 36%. كما أظهرت النتائج تبايناً في مستويات الثقة لدى الطلبة فيما يتعلق بقدرتهم على توظيف التقنيات الحديثة في تطوير التطبيقات. وتشير النتائج إلى أن التصميم الهجين للمقرر نجح في استيعاب التفاوت في الخلفيات الأكاديمية ومستويات الإعداد السابقة للطلبة، كما تبين أن أداء الطلبة في أعمال المقرر يُعد مؤشراً مهماً للتنبؤ بأدائهم في الاختبار النهائي $(\beta = 0.57, p < 0.001)$. كذلك $R^2 = 0.40$ أظهرت الدراسة أن الأداء المبكر في الواجبات والأنشطة التطبيقية يمكن أن يُستخدم كمؤشر إنذار مبكر لتحديد الطلبة المعرضين لخطر تدني التحصيل الأكاديمي. وتؤكد هذه النتائج فاعلية التصميم التعليمي الهجين في تعزيز تعلم تطوير تطبيقات الأجهزة المتنقلة وتهيئة الطلبة للتعامل مع التقنيات الناشئة ومتطلبات سوق العمل الحديثة.

الكلمات الدلالية: تطوير تطبيقات الأجهزة المتنقلة، الذكاء الاصطناعي، إنترنت الأشياء، الحوسبة المحيطة، التعليم الهجين، تصميم المناهج، الاستراتيجيات التربوية.