

Y-Learn: Design and Initial Implementation of an OBE-Based Learning Platform for Group Work Monitoring and Peer Review Assessment

Yerik Afrianto Singgalen¹, Efendi², Jodie Giovanna Abrahamsz³

^{1,3}Tourism Study Program, School of Business and Social Innovation, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia

²Management Study Program, School of Business and Social Innovation, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia

Received:

May 17, 2026

Revised:

July 25, 2026

Accepted:

September 2, 2026

Published:

October 24, 2026

Corresponding Author:

Author Name*:

Yerik Afrianto Singgalen

Email*:

yerik.afrianto@atmajaya.ac.id

DOI:

10.63158/journalisi.v8i5.1808

© 2026 Journal of Information Systems and Informatics. This open access article is distributed under a (CC-BY License)



Abstract. Group assignments often preserve the submitted product more clearly than individual participation. Y-Learn was developed to retain member-attributed contributions alongside peer records, lecturer decisions, and Outcome-Based Education (OBE) information. Using Design Science Research, requirements were derived from course outcomes, Sub-CPMK/Sub-CLO structures, assessment documents, development records, and issues observed during teaching use. The native Android artifact uses Kotlin, Jetpack Compose, Firebase Authentication, and Cloud Firestore. Evaluation combined repository inspection, automated technical checks, production-console evidence, three course analytics exports, and a later black-box cycle. Version 3.3 recorded 24 passing unit tests and 10 passing governance-rule scenarios; version 3.4 was subsequently released and examined through an additional black-box evaluation. PAR 102, SFT 106, and SFT 210 contained 39, 42, and 39 normalised OBE evidence records, with assessment coverage of 48.7%, 100%, and 100%. Peer-review activity varied substantially, and unresolved records remained visible. No formal usability study, peer-score reliability analysis, or learning-outcome impact evaluation was conducted. The contribution is an implemented artifact and a traceability mechanism connecting activities, member contributions, peer evidence, and lecturer-controlled assessment.

Keywords: Collaborative Assessment, Design Science Research, Learning Outcome Traceability, Outcome-Based Education, Peer Review

1. INTRODUCTION

Course outcomes become difficult to examine once the link to the activity and assessed work is lost. In Y-Learn, the course learning outcome serves as the general point of reference; Sub-CPMK or Sub-CLO and the rubric identify the more specific performance to be examined. Romanowski et al. discuss this relationship as constructive alignment, which depends on consistency among intended outcomes, teaching activities, and assessment [1]. The same problem remains in blended participation, as reported by Simon et al. for online and face-to-face settings [2]. The application therefore keeps the outcome record connected to the relevant activity, submission, and lecturer assessment. This arrangement allows the assessment trail to be followed without treating OBE labels as stand-alone documentation.

Group assessment revealed a different documentation problem. The report or presentation captured the collective result, but did not routinely show which member handled a task, when the work was recorded, or whether participation was understood in the same way across the group. For example, when several students submit one group report, the lecturer may see the finished product without a contemporaneous record of who prepared a section, responded to feedback, or completed a supporting task. This gap prompted the development of Y-Learn; the application was not intended to displace lecturer judgement of the submitted product. Adesina et al. associate peer assessment with engagement, while also noting that the effect depends on the organisation of participation and assessment [3]. Fleckney et al. likewise emphasise the need for a deliberately structured peer-feedback process [4]. Y-Learn consequently retains contribution histories and peer records beside lecturer assessment. Peer scores may support interpretation, but are not treated as a complete account of individual participation. Research on cooperative learning also links structured collaboration with engagement, although the relationship remains dependent on how participation is organised [5].

Most learning management systems already cover core tasks such as distributing materials, receiving assignments, supporting communication, and recording grades. What lecturers obtain from those functions depends heavily on local configuration and on the way the system is incorporated into teaching [6], [7]. Moodle, for example, can be

configured for authentic learning activities [8]. Y-Learn addresses a narrower requirement. It places member-attributable contribution records, peer ratings or comments, lecturer evaluation, and outcome-achievement information in one Android-based course workflow. The comparison made here concerns that arrangement of records; the study did not test whether Y-Learn outperforms an established LMS.

The inquiry was organised around three research questions. RQ1 identifies the functional and non-functional requirements relevant to collaborative learning in an OBE setting. RQ2 considers how contribution records and peer evidence may be read alongside lecturer assessment and learning-outcome information. RQ3 asks how those relationships can be retained so that the basis of a group assessment remains traceable after the activity has ended. Together, the questions informed the user roles, data relations, screen flows, and assessment records described in the following sections. The article follows the artifact through design, implementation, release, and operational use; that use is not presented as an experiment on educational effectiveness.

The contribution made here is an implemented application and a documented account of the reasoning behind its workflow, rather than a new formulation of OBE. Levrai argues that collaborative assessment may need to address two objects at once: the submitted product and the process through which the group produced it [9]. Little et al. further note that students require workable conditions for interpreting criteria and composing useful peer feedback [10]. Y-Learn translates these concerns into a record structure in which outcomes, assessment criteria, contribution histories, peer input, and lecturer review can be examined together. Its practical relevance lies in widening the evidence available to the lecturer. The study did not measure whether this arrangement improves fairness, learning, or assessment effectiveness.

Lecturers and students use the learning functions of Y-Learn, while administrators monitor accounts and selected operational records. The July evaluation examined version 3.3 with versionCode 33 at full Google Play rollout; version 3.4 with versionCode 34 subsequently entered the production track. Beyond the collaborative functions discussed here, the application contains course and OBE structures, assignments, quizzes or examinations, presentations, discussions, contribution moderation, clarification records, achievement monitoring, and an OBE Traceability Navigator with CSV and PDF exports.

Course-level exports from PAR 102, SFT 106, and SFT 210 provide documentary evidence that these records were populated during teaching. Y-Learn was developed and examined within one institutional context and can organise more than one course, but broader programme- or institution-level adoption has not been evaluated. The Android-first implementation was selected to keep the course and collaborative-assessment workflow within a single mobile application rather than depend on an LMS-specific extension architecture; no comparative development study against plugin alternatives was conducted. The production version does not provide a separate institutional quality-assurance dashboard and has not been integrated with an LMS, student information system, single sign-on service, or LTI environment. Studies of continued e-learning use and information-system success show that technical availability does not settle questions of system quality, user experience, perceived value, or continued adoption [11], [12]. Production release and course records establish operational use; neither establishes sustained adoption, usability, or improved educational outcomes. Related studies of higher-education digitalisation describe a broader move towards automated and paperless academic processes [13], yet technology-related educational outcomes remain dependent on teaching and assessment conditions rather than delivery technology alone [29]. Evidence on service quality and student perception likewise shows why deployment records cannot substitute for user-based evaluation [14].

2. METHODS

Design Science Research (DSR) was adopted because the study centred on the development and examination of an information-system artifact within an educational setting. Carstensen and Bernhard describe DSR as an approach in which the design process and the resulting artifact are examined together, allowing practical development to remain connected with systematic inquiry [15]. Artifact evaluation is equally important within this approach; Venable et al. position evaluation as a means of examining whether a designed artifact addresses its intended purpose while also informing subsequent refinement [16]. For Y-Learn, the research process encompassed problem identification, requirement specification, Android development, Firebase integration, refinement of contribution governance and OBE traceability, technical verification, production release, and examination of application-generated course records. Evaluation consequently drew on evidence produced at different points in the artifact lifecycle rather than on a single

test procedure. The study did not measure perceived usability, assessment fairness, peer-score reliability, or causal changes in learning outcomes.

The requirements were assembled from teaching practice and course documentation rather than collected from a respondent sample. The review covered course outcomes, Sub-CPMK or Sub-CLO structures, learning activities, assignment instructions, rubrics, group-work procedures, contribution and peer records, lecturer assessment needs, development notes, and release requirements. No interviews, surveys, workshops, or formal stakeholder panel were conducted, so participant demographics are not reported. Lecturer, student, and administrator needs were expressed as role requirements drawn from the working course process and refined during development and operational use. Experience from operational use informed application changes, although the experience was not recorded or analysed as a user-study dataset. Participatory prototyping and stakeholder-preference studies offer alternative ways to involve users in educational system design [17], [18]; those procedures were not undertaken in the present study.

The development was organised into eight connected activities. Work began with the difficulty of seeing individual participation behind a collective submission and then defined objectives for traceability, contribution visibility, peer evidence, and lecturer control. Functional and non-functional requirements were separated before roles, process sequences, record relationships, and outcome-to-evidence links were modelled. The native Android implementation was subsequently extended with contribution moderation, student clarification, workspace attention summaries, administrative activity records, and an OBE Traceability Navigator. Verification combined repository inspection with automated unit and Firestore rules tests, lint, build checks, and console evidence. Production deployment and the generation of course-level analytics reports completed the sequence reported here.

Rather than examining Firestore collection by collection, the analysis reconstructed the path taken by an assessment record. For each selected event, the review identified the responsible actor, the action performed, the criterion or outcome involved, and the evidence retained by Y-Learn. The sequence was then followed from lecturer configuration to student work, peer response, contribution moderation, clarification where needed, lecturer judgement, and the resulting achievement summary. Val and

Quintas note that indicators from an educational platform have little interpretive value when their relationship to course design and academic activity is unclear [19]. Heil and Ifenthaler similarly distinguish the functions of teacher, peer, and self-feedback in online assessment [20]. The resulting map was used to examine whether the academic activity, assessment purpose, and stored record remained coherent. It was not treated as evidence of pedagogical validity or user acceptance.

Production deployment was examined from the Android client towards the supporting services used by the application. The review did not identify a separately administered application server. Application behaviour and the role-specific interface are implemented with Kotlin and Jetpack Compose, with ViewModels and Hilt used for state and dependency management. User identity is handled through Firebase Authentication, while the Android client accesses course, OBE, group, contribution, clarification, submission, assessment, discussion, analytics, and monitoring records in Cloud Firestore through the official SDK. The repository also contains callable Cloud Functions code for selected privileged operations. Although this code passed syntax inspection, its deployment could not be verified because the Firebase project remained on the Spark plan; it was therefore excluded from the confirmed production services. A Firebase Storage dependency was also present in the project, but no active production upload route was identified during the review. Within these limits, the verified production environment comprised the Android client, Firebase Authentication, Cloud Firestore, deployed Security Rules, and distribution through Google Play.

Access control operates at several points in this environment. Authentication was available through Email/Password and Google sign-in, after which account status and role routing determined whether a session could proceed. Accounts that were missing, inactive, or deleted were not processed in the same manner as active accounts. At the data layer, Firestore Security Rules evaluated requests against both the authenticated identity and the relationships associated with the requested record. Ten selected Emulator scenarios were executed successfully, covering contribution moderation, clarification, denial of self-escalation, denial of audit-log writes, and bounded operational records. The Android manifest additionally disabled cleartext traffic and application backup, while App Signing and Automatic Protection were provided through Google Play. These controls describe the security measures verified in the current implementation

rather than an independently validated security posture. Rule coverage remains partial, Firebase App Check and the application-level Play Integrity API were inactive, and neither penetration testing nor a privacy-impact assessment formed part of the evaluation.

The first technical interpretation was restricted to evidence available on 30 July 2026 and therefore corresponds to version 3.3. Repository inspection covered application configuration, authentication, navigation, data models, contribution governance, OBE analytics, Firestore rules, audit records, and release settings. Verification activities included Android unit testing, Gradle checks, lint, debug and release assembly, the Firestore Emulator rules suite, and a Node syntax check of functions/index.js. Rather than treating all available evidence as equivalent, each result was classified according to what had actually been verified: source verified, automated-test verified, console verified, blocked, optional hardening, or data gap. This distinction was retained so that the presence of source code, an unsuccessful execution attempt, or an optional security control would not be interpreted as evidence of completed production testing or observed user behaviour.

A second evaluation cycle was conducted after version 3.4 (versionCode 34) entered Google Play production. Manual black-box testing began on 14 August 2026 using a version 3.4 debug APK and observable interface behaviour as the criterion for PASS. The plan contained 78 scenarios spanning authentication, role access, course operations, assessment workflows, analytics, administration, and error handling. Direct UI execution or attempted execution with evidence was possible for 13 cases in the baseline cycle. A full-chain retest on 18 August 2026 used authenticated student and lecturer sessions, a version 3.4 debug rebuild after fixes, and the isolated YL-BBX-2026-0818 dataset. The observed sequence covered OBE/Sub-CPMK mapping, course creation and publication, enrollment, session/activity creation, student evidence submission, lecturer moderation, student revision or clarification, lecturer assessment, traceability analytics, CSV/PDF export, and administrator course monitoring. Peer evidence was not executed because the isolated workflow did not include a second peer student or presentation activity, so no PASS was assigned to peer review. The black-box execution, repository inspection, and internal automated testing were developer-led; no independent third-party tester or expert panel was involved. Firebase and Google Play console records supplied external

service-state evidence, but interpretation of those records remained the responsibility of the author.

Operational evidence came from OBE Analytics reports generated by Y-Learn for PAR 102, SFT 106, and SFT 210 on 30 July 2026. Each export applied the application's normalized rule: for each student and meeting, only the latest quiz or examination evidence and the latest assignment or examination evidence were counted. The document analysis recorded students represented in the evidence, total and assessed records, pending review, mapped Sub-CPMK, activity and session links, discussion participation, presentations, and peer-review entries. Student identities were removed from the manuscript. The reports were treated as evidence of operational record generation and traceability, not as a respondent sample, SUS dataset, or causal classroom experiment.

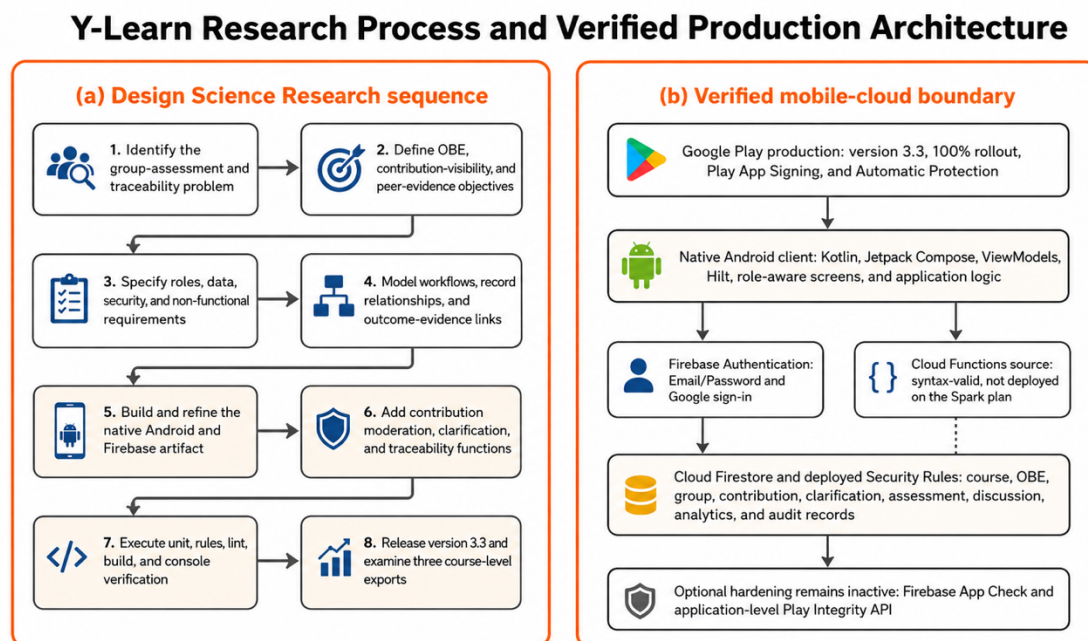


Figure 1. Verified development sequence and production architecture of Y-Learn

Figure 1 relates the project history to the production arrangement confirmed on 30 July 2026. Panel (a) moves from the assessment problem and OBE modelling through implementation, governance changes, automated checks, release, and review of three course exports. Panel (b) shows which services were observed in production and which remained at source level or as optional controls. Version 3.3 reached users through Google Play, with the Android client connected to Firebase Authentication, Cloud

Firestore, and deployed Security Rules. Callable Cloud Functions code passed syntax inspection but had not been deployed under the Spark plan. Firebase App Check and application-level Play Integrity were also inactive. The boundary drawn in the figure therefore reflects confirmed operation rather than every dependency listed in the project.

Table 1. Development work aligned with the Design Science Research stages

DSR stage	Work carried out	Resulting record or output
Problem identification	Review the assessment problem created when a group product obscures individual contribution and peer views	Problem statement and study limits
Objective definition	Set the intended OBE links, contribution visibility, peer evidence, and lecturer oversight	Artifact objectives
Requirement analysis	Organise role-based needs together with assessment, privacy, security, usability, reliability, device, scalability, and interoperability concerns	Requirement specification
Design	Translate the requirements into roles, learning and assessment paths, data relations, Android screens, Firebase services, and outcome-evidence links	Functional design and architecture
Development	Build the Kotlin and Compose application and connect authentication, Cloud Firestore, deployed Security Rules, and role routing	Operational Android artifact
Demonstration	Use lecturer and student workflows in teaching and examine selected administrator operations	Record of operational workflows
Technical review	Execute Android unit tests, selected Firestore Emulator rules tests, Gradle verification, lint, debug/release builds, and console inspection	Verified technical results and bounded test limits

DSR stage	Work carried out	Resulting record or output
Communication and release	Publish version 3.3 through Google Play and examine OBE Analytics exports from three courses	Production release, manuscript, supplementary reports, and course-level evidence

Table 1 places the completed work within the DSR sequence while keeping deployment separate from comprehensive validation. Android development and Firebase configuration were refined together because identity, data access, contribution governance, and OBE traceability were parts of the same course workflow. Demonstration relied on lecturer and student paths used during teaching. Evaluation used 24 Android unit tests, 10 selected Firestore Emulator scenarios, lint and build checks, console inspection, and analysis of three course exports. Google Play publication established operational distribution. The evidence supports claims about the implemented artifact, selected access boundaries, and generated academic records, but not about usability, certified security, or causal improvement in learning. Multi-layered evaluation of educational interventions also illustrates why implementation, engagement, and outcome evidence need to be treated as distinct analytical levels [21].

Table 2. Requirement traceability from teaching problem to implemented feature

Source/problem	Derived requirement	Implemented feature or record	Evidence boundary
Course outcomes, Sub-CPMK/Sub-CLO, and rubrics	Retain the link between intended outcome, activity, assessment, and reviewed evidence	Course/OBE records, activity links, achievement summaries, and OBE Traceability Navigator	Traceability of stored records; not pedagogical validity
	Collective submissions can obscure member participation	Retain member-attributable contribution evidence with group, activity, status, and time context	Contribution records and working-group context
			Documents recorded participation; not proof of authorship or effort

Source/problem	Derived requirement	Implemented feature or record	Evidence boundary
Lecturer review of disputed or incomplete contribution evidence	Keep academic judgement with the lecturer and preserve revision/clarification exchanges	Contribution moderation, rejection/revision reason, student clarification, lecturer response	Workflow governance; not fairness validation
Peer input may add context but should not automatically determine grades	Store activity-specific peer evidence separately from official assessment	Peer-rating/comment records and lecturer-controlled assessment	No validated automatic grade conversion or peer-score reliability claim
Role and privacy requirements in academic records	Restrict functions and data access by authenticated identity, role, enrolment, and record relationship	Firestore Authentication, role routing, account-state checks, deployed Firestore Security Rules	Selected rule paths tested; no certification or complete collection coverage
Need for review outside the live interface	Provide filtered summaries for lecturer follow-up and documentary review	CSV/PDF OBE exports and administrative monitoring	Generated records demonstrate availability; not usability or institutional adoption

Table 2 links the documentary and teaching problems used in requirement analysis with the corresponding application response. The mapping is a traceability device for the design process rather than evidence that each requirement has been independently validated. Requirements concerning usability, accessibility, privacy, reliability, scalability, maintainability, and interoperability are evaluated separately against the evidence and limits reported below.

3. RESULTS AND DISCUSSION

3.1. Results

Y-Learn version 3.3 was at full Google Play rollout when the July technical and course-document evaluation closed. Repository and console records confirmed the implemented roles, Firestore relationships, contribution moderation, clarification handling, OBE traceability, exports, and administrative monitoring. Version 3.4 (versionCode 34) subsequently entered Google Play production and was used as the version basis for an additional black-box evaluation; manual execution itself used the corresponding debug APK on Android emulators. The later interface-level evidence comprises the initial black-box baseline and the 18 August full-chain retest using the isolated test course. The analysis therefore draws on three evidence layers: the earlier automated and console verification, course-level operational exports, and the later interface-level black-box evidence. Each layer answers a different question. A successful technical check says nothing about user perception, an analytics export cannot establish that Y-Learn caused the recorded achievement level, and a black-box PASS is limited to the observed behaviour under the tested build, device, account, and data conditions.

Table 3. Functional scope, actors, and retained records in Y-Learn

Functional area	Main role	Operation in the application	Record retained
Authentication and account access	All roles	Email/Password or Google sign-in, session restore, account-status check, and role routing	Identity, profile, session, role, and account status
Course and OBE	Lecturer	Maintain course details, participants, sessions, materials, CPL, CPMK/CLO, and Sub-CPMK/Sub-CLO links	Course, outcome, participant, schedule, and resource data
Learning and assessment	Lecturer / student	Create or complete assignments, quizzes or examinations, submissions, feedback, revisions, and results	Activity, submission, score, feedback, and revision data
Working groups	Lecturer / student	Arrange membership, leadership or participation context, and the collaborative workspace	Group and membership data

Functional area	Main role	Operation in the application	Record retained
Contribution records	Student / lecturer	Submit contribution evidence; verify, request revision, reject, revise, clarify, and review permitted records	Contribution, moderation status, reason, clarification, response, identity, context, and time data
Peer review or rating	Student / lecturer	Submit ratings or comments and expose permitted records for lecturer review	Reviewer, reviewee, activity, rating, comment, and timestamp
Presentation and discussion	Lecturer / student	Retain presentation participation and academic discussion in the course workflow	Presentation, discussion, participant, content, and status data
Achievement monitoring	Lecturer / student	Read filtered progress, outcome, activity, student, and graded-evidence information; export traceability summaries	Outcome-linked achievement and CSV/PDF traceability records
Administrative monitoring	Administrator	Review users, account states, grade-audit records, and Group Workspace operational activity	Account, audit, and bounded operational records

Table 3 shows how an operation in Y-Learn leaves a record that can later be inspected. Students generate contribution and peer-review evidence, but the authority to verify, moderate, and finalise academic assessment remains with the lecturer. Accordingly, peer ratings and comments are read together with submissions, contribution histories, presentation or discussion records, and the relevant rubric. Peer-review results remain separate from the lecturer's final grading decision.

Table 4. Role permissions and access boundaries in the production application

Role	Application permissions	Firestore/access basis	Boundary
Administrator	Review users and account states, inspect grade-audit and Group Workspace activity, support account continuity, and open permitted lecturer-mode functions.	Authenticated administrator role plus rules/record relationships for permitted operational records. Selected Emulator tests covered bounded logs and denial paths.	No default authority to alter lecturer-controlled academic grades; callable function source was not active under the earlier Spark-plan boundary.
Lecturer	Manage courses, outcomes, activities, assessments, groups, contribution moderation, clarification responses, peer records, feedback, analytics, and achievement summaries.	Authenticated lecturer identity combined with course ownership or permitted course relationships evaluated by Firestore Security Rules.	Peer evidence is interpreted by the lecturer; no validated automatic peer-to-grade conversion is used.
Student	Open authorised enrolled-course activities, submit work and contribution evidence, complete assigned peer review, revise returned evidence, request clarification, and view authorised feedback.	Authenticated student identity, enrolment, group/activity relationship, and record ownership determine permitted reads/writes.	Lecturer assessment records remain read-only to students and administrative functions are unavailable.
Class leader (contextual	Participate in the same enrolled-course	Uses the student identity and applicable	No administrative privilege and no

Role	Application permissions	Firestore/access basis	Boundary
student status)	and group workflows, with group-lead context where the activity uses it.	course/group relationships; class-leader status does not create a separate administrator role in Firestore.	default authority over academic grades or other groups.

Table 4 expresses each role through both the functions made available and the limits attached to them. Administrators monitor accounts, grade-audit information, and selected workspace activity without receiving default authority to change academic grades. Lecturers manage the course and OBE structure, moderate contribution evidence, answer clarification requests, inspect peer records, and record assessment decisions. Student participation is limited to enrolled activities and includes submitting work and contribution records, completing assigned peer reviews, revising returned material, and requesting clarification when needed. Class-leader status applies only within the relevant group context and does not confer administrative authority.

Table 5. Conceptual records and workflow relationships in Y-Learn

Logical entity	Representative fields	Place in the workflow
User	Identity and access: userId, name, email, institutional identifier, role, accountStatus, isActive; metadata: timestamps	Links an authenticated person to course participation, group work, contribution, assessment, and monitoring records
Course	Course identity: courseId, lecturerId, title; context: description, class or cohort, schedule, status	Provides the container for outcomes, participants, learning activities, assessments, and collaborative work
Learning outcome	outcomeId, courseId, outcome type, statement, indicator, status or weight	Provides the intended result to which activities and evidence are related

Logical entity	Representative fields	Place in the workflow
Learning activity or assignment	activityId, courseId, outcomeId, title, instructions, due date, criteria, status	Defines the learning or assessment context
Working group	groupId, courseId, assignmentId, name, memberIds, leaderId or status	Associates students with a collaborative task Connects member-
Contribution record	contributionId, groupId, activityId, studentId, description or reference, status, moderation reason, timestamps	attributable evidence to its group and activity and records lecturer moderation
Peer-review record	reviewId, reviewerId, revieweeId, group or activity reference, score or rating, comment, timestamp	Stores peer input for authorized review
Submission or evidence	submissionId, activityId, student or group reference, content or reference, submittedAt, revision status	Keeps student or group work with the relevant assessment
Lecturer assessment	assessmentId, activity or submission reference, lecturerId, score, feedback, timestamp	Retains lecturer-controlled judgement and feedback
Discussion or presentation record	recordId, course or activity reference, participant reference, content or status, timestamp	Records academic interaction and participation
Achievement summary	studentId and outcomeId, with evidence references, recorded or calculated result, and status	Combines the records used to review progress against a stated outcome
Clarification record	clarificationId linked to contributionId or activityId, studentId, request, lecturer response, status, and timestamps	Documents the exchange that follows a revision request or rejection

Table 5 is a conceptual map of the records needed by the workflow; its labels should not be read as literal names for every Firestore collection. The current implementation adds moderation information to contribution records and links revision or rejection to a clarification record containing the student request, lecturer response, status, and time metadata. Through these relationships, a lecturer can move from the course and outcome

to the activity, group, submission, contribution, peer review, clarification, assessment, interaction record, and achievement summary. No single record is treated as sufficient evidence of individual achievement.

Table 6. Non-Functional requirements, implemented support, and evaluation boundaries

Non-Functional requirement	Implemented or observed support	Evaluation boundary
Usability	Role-specific screens, state feedback, confirmation dialogs, attention summaries, and an operational user manual	No SUS, UEQ, TAM, or sampled task-completion study
Accessibility and device adaptation	Jetpack Compose layouts, adaptive navigation, and phone/tablet refinements	Instrumented execution was blocked; no accessibility-conformance audit
Security and privacy	Authentication, active-account checks, deployed Firestore Rules, 10 selected Emulator tests, audit records, disabled cleartext/backup, Play signing and protection	No penetration test, complete collection coverage, threat model, privacy-impact assessment, or certification
Reliability	Twenty-four Android unit tests passed; Gradle check, lint, debug, and release builds completed	No longitudinal uptime, fault-injection, recovery-time, or healthy-device instrumented result
Performance	Memory optimisation, R8 package-size reduction, and release stability work	No controlled startup, latency, memory, battery, or network benchmark
Scalability	Managed Firebase services avoid a separately maintained application server	No application-specific load, concurrency, or cost test
Maintainability	Kotlin, Jetpack Compose, ViewModels, Hilt, structured records, and modular role flows	No external maintainability index or code-quality certification
Interoperability	Focused Android and Firebase implementation with CSV/PDF traceability exports	No completed LMS, SIS, SSO, LTI, or institutional data-warehouse integration

Table 6 sets the engineering evidence beside the properties that remain untested. Unit tests, lint, builds, and selected Firestore rule checks were completed. Firebase Console records showed that authentication and rules were deployed, while Google Play confirmed production distribution and store-side protection. This evidence is more substantial than screenshots alone, but it remains confined to the checks actually performed. No sampled usability score, accessibility-conformance result, penetration-test finding, longitudinal reliability measure, controlled performance benchmark, load test, or verified institutional integration is available.

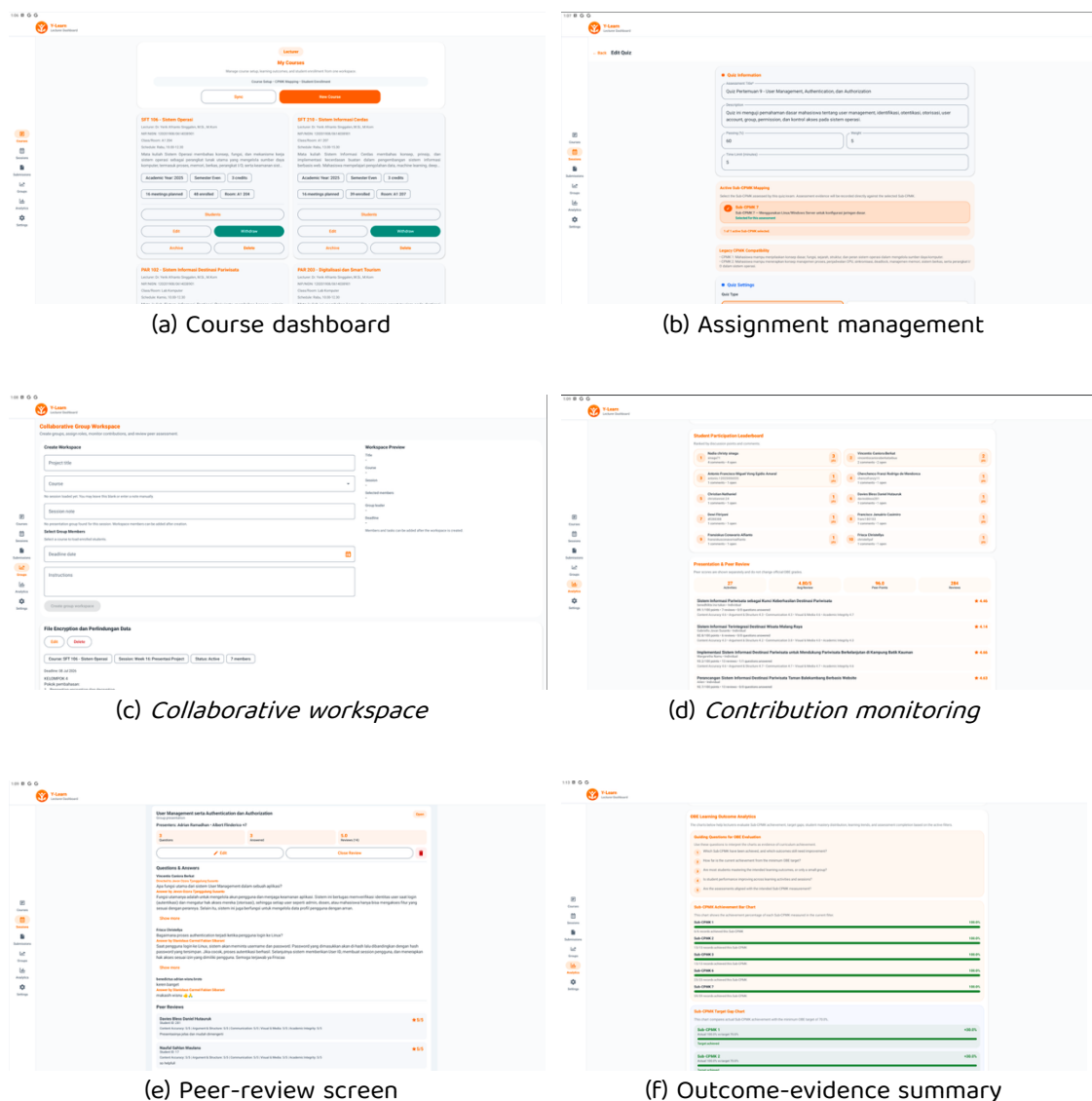


Figure 2. Interfaces documenting the Y-Learn course and assessment workflow

Figure 2 assembles six interfaces drawn from lecturer and student activity: the course overview, assignment setup, collaborative workspace, contribution view, peer-review screen, and outcome-evidence summary. The panels are separated to keep the main labels and controls readable. The six interfaces document a connected sequence through which collaborative evidence is recorded and reviewed. The screenshots alone do not establish usability, accessibility, or educational effectiveness. Evidence for contribution moderation, student clarification, attention summaries, and OBE traceability exports is provided separately through source inspection, testing, and generated reports.

Table 7. Record links supporting OBE traceability

OBE element	Relationship in the design	Record in Y-Learn	Role of the record
Course learning outcome	Defines the expected course-level performance	Outcome record associated with the course	Reference point for activities and evidence
Sub-CPMK or Sub-CLO	Translates the course outcome into a more specific target	Sub-outcome linked to the course outcome	More precise traceability point
Learning activity	Provides the setting in which the target is developed	Session, assignment, quiz, examination, presentation, or discussion record	Connects planned learning with observable activity
Assessment activity	Specifies instructions, criteria, deadline, and assessment context	Assignment, quiz, examination, or other assessment record	Defines where assessable evidence is produced
Submission or group evidence	Retains work produced by a student or group	Submission, group, and membership records	Documents the assessed product and collaborative setting
Individual contribution	Associates a recorded activity or	Contribution record linked to course, activity, group,	Makes recorded participation visible

OBE element	Relationship in the design	Record in Y-Learn	Role of the record
	evidence item with a student	description/reference, and time	
Peer evidence	Retains student perspectives on participation	Peer record with reviewer, reviewee, context, rating/comment, and timestamp	Adds context for lecturer interpretation
Lecturer assessment	Records lecturer judgement and feedback	Assessment, score, feedback, and revision records	Authoritative academic evaluation
Achievement summary	Assembles permitted evidence against an outcome	Student-outcome summary with evidence references and status	Supports progress and outcome monitoring
Traceability export	Preserves a filtered outcome-to-evidence view for review outside the application	CSV/PDF summary of outcomes, activities, sessions, students, assessed evidence, and status	Supports documentary review and follow-up

Table 7 follows the record chain from the intended course outcome to the evidence available for lecturer review. The course outcome and Sub-CPMK establish the target; learning and assessment activities define where evidence is produced; submissions, group information, contribution histories, peer records, and lecturer assessment add different forms of evidence; and the achievement summary brings relevant records together. A contribution count or peer rating is therefore contextual evidence, not proof that a learning outcome has been achieved.

A concrete course-export example is available from PAR 102. The report linked "*Quiz - Pengenalan SIG dan Komponen Sistem Informasi Spasial*" to Session 1, "*Konsep Dasar SIG & Komponen Sistem Informasi Spasial*," and identified Sub-CPMK 1 as the measured outcome. Six of six records for that mapped activity had been assessed, with an average score of 100.0% and reported Sub-CPMK achievement of 100.0%. This example verifies the outcome-to-activity-to-session-to-assessed-evidence path retained in the export. Contribution and peer-review records for the same quiz were not presented as part of

that specific export chain, so the example is not used to claim an end-to-end link between peer evidence and the official grade.



Figure 3. Contribution moderation, clarification, and peer-evidence workflow

Figure 3 traces the current workflow from outcome and activity configuration to contribution evidence, lecturer moderation, student clarification, peer input, assessment, and OBE traceability. Contribution records remain associated with the authenticated student, course, activity, group, status, and time metadata. A lecturer may verify the evidence, request revision, or reject it with a reason; a student may then revise the record or submit clarification, after which the lecturer records a response. Peer ratings or comments remain separate from official grades and are reviewed with the submission, discussion, presentation, contribution history, and assessment data. The OBE Traceability Navigator and its CSV/PDF exports provide a filtered outcome-to-evidence summary. The workflow improves documentary visibility, but does not by itself validate authorship, fairness, peer reliability, or learning attainment.

Table 8. Descriptive peer-review criteria for later controlled use

Criterion	Description for controlled use	Scale
Contribution completion	Extent to which agreed tasks were completed and supported by observable evidence	1-5
Quality of contribution	Relevance, accuracy, and usefulness of the work contributed to the group	1-5
Collaboration and communication	Responsiveness, constructive interaction, peer support, and participation in discussion	1-5
Timeliness and responsibility	Reliability in meeting agreed deadlines and following through on assigned work	1-5

Table 8 presents contribution-focused criteria that may support later controlled validation. The normalized expression places ratings on a 0-100 reporting scale, but no validated policy currently uses that result to determine a final course grade. The course exports examined in this study summarize activity-specific presentation rubrics on a 1-5 scale and retain those peer results separately from official OBE grades. Table 8 is therefore an analytical specification rather than the source of the reported course-level peer-review counts.

Table 9. Verified technical results, operational course evidence, and black-box testing

Activity or evidence source	Verified result	Interpretive boundary
Production release	Version 3.3/code 33 was at full Google Play rollout for the July evaluation; version 3.4/code 34 subsequently entered production.	Establishes distribution and version chronology, not sustained adoption or educational impact.
Android unit tests	24 tests passed; 0 failures; 0 errors	Verifies selected JVM logic and policy boundaries, not user experience
Gradle, lint, and builds	Gradle check, lint, debug assembly, and release assembly completed; release APK produced	Build and static verification, not a controlled performance benchmark

Activity or evidence source	Verified result	Interpretive boundary
Firestore governance	10/10 selected Emulator tests passed for moderation, clarification, denials, and bounded logs	Covers selected paths, not every collection or authorization boundary
Firebase and Play Console	Authentication providers enabled; rules deployed; audit records present; Play signing and protections active	Console evidence, not penetration testing or certification
Instrumented Android test	Execution blocked because the emulator exposed no Android services to ADB	No device-level pass or fail result
PAR 102 export	20 students with evidence; 39 records; 19 assessed; 20 pending; 48.7% coverage; 3 mapped Sub-CPMK; 44 peer reviews	Descriptive course export under the normalized latest-evidence rule
SFT 106 export	42 students with evidence; 42 assessed records; 100% coverage; 1 mapped Sub-CPMK; 1,114 peer reviews	Descriptive course export; not a usability or causal dataset
SFT 210 export	39 students with evidence; 39 assessed records; 100% coverage; 2 mapped Sub-CPMK; 1,755 peer reviews	Descriptive course export; not a usability or causal dataset
Black-box evaluation of version 3.4	Version 3.4 debug testing began with the 78-scenario baseline and continued with the consolidated full-chain retest on 18 August 2026. The final 17-scenario full-chain set contained 15 PASS, one PARTIAL, and one BLOCKED result. The observable non-peer chain covered OBE/Sub-CPMK configuration, course creation and	Interface-level evidence under the tested accounts, devices, build, and isolated dataset; it does not establish usability, accessibility, educational effectiveness, assessment fairness, peer-score reliability, privacy compliance, security

Activity or evidence source	Verified result	Interpretive boundary
	publication, enrollment, session/activity creation, student evidence submission, lecturer moderation, student revision/clarification, lecturer assessment, traceability analytics, CSV/PDF export, and administrator course monitoring; peer review remained blocked by the isolated test setup.	certification, scalability, or long-term reliability.

Table 9 reports technical checks and course exports together, but does not treat them as equivalent evidence. The Android suite completed 24 tests without failure or error, all ten selected Firestore governance scenarios passed, and Gradle verification, lint, debug assembly, and release assembly were completed. Device-level instrumented testing could not run because the available emulator exposed no Android services to ADB. The Cloud Functions source passed syntax checking, although deployment was not verified. This reporting prevents a blocked execution or source-only component from being counted as a completed production test.

Course coverage was not uniform. PAR 102 contained 39 normalised OBE evidence records representing 20 students in the export; 19 records had been assessed and 20 were still awaiting review, giving 48.7% coverage across three mapped Sub-CPMK. SFT 106 contained 42 assessed records with complete coverage for one mapped Sub-CPMK, whereas SFT 210 contained 39 assessed records with complete coverage across two mapped Sub-CPMK. Peer-review activity also varied, with totals of 44, 1,114, and 1,755 records. Because the figures represent separate course-level records, combining them would not produce a deduplicated participant count.

The exports also documented unresolved work. PAR 102 retained 20 unassessed evidence items; SFT 106 recorded 17 unanswered discussion comments; and SFT 210 recorded 235 unanswered comments together with an 8.6% presentation-question response rate.

Average assessed scores ranged from 97.9% to 99.5%, and the mapped Sub-CPMK indicators were reported at 100% against a 70% target. Those figures describe the filtered records produced by the application and its latest-evidence rule. Their value here lies in showing that Y-Learn can reveal completed and incomplete processes for lecturer follow-up, not in supporting a causal claim about learning effectiveness.

Table 10. Black-box testing results for major workflows

Workflow	Test case(s)	Observed behaviour	Status
Authentication validation	BB-AUTH-005; BB-AUTH-006	Blank login remained unavailable and an invalid email produced a visible format error.	PASS
Rejected credentials	BB-AUTH-004	The login screen remained unauthenticated, but no visible rejection message appeared in the baseline test.	FAIL
Role/access restriction	BB-ROLE-003; BB-ROLE-004	Administrator and lecturer-mode behaviour were observed; a complete non-admin denial-path retest was not part of the final full-chain sequence.	PARTIAL / BLOCKED
Course, OBE, and session/activity setup	BB-COURSE-001; BB-P2-OBE-001; BB-P2-SESSION-001; BB-ASG-001	The dedicated test course was created and published, Sub-CPMK mapping was visible in the course/session/activity path, and the collaborative activity became available to the student.	PASS
Enrollment	BB-ENR-001; BB-ENR-002; BB-ENR-004	The student submitted an enrollment request, the lecturer approved it, and the course then appeared to the student with ENROLLED status and an Open action.	PASS
Contribution submission	BB-REV-001	The student submitted link evidence and the record status changed to Submitted.	PASS

Workflow	Test case(s)	Observed behaviour	Status
Moderation and clarification	BB-REV-002; BB-REV-003; BB-REV-004	The lecturer opened the submitted evidence and requested revision; the student saw the lecturer feedback and submitted revised evidence, after which the status changed to Revision Submitted.	PASS
Lecturer assessment	BB-P2-ASSESS-001	The lecturer finalised the test assessment, and the resulting grade was subsequently visible in the authorised student grade summary.	PASS
OBE traceability	BB-TRACE-001	The populated YL-BBX-2026-0818 analytics filter returned one record and all non-peer links in the configured workflow were observed; peer evidence remained absent.	PARTIAL
CSV/PDF export	BB-ANL-003; BB-P2-EXPORT-PDF-001	With non-zero traceability data, the Android share sheet displayed generated CSV and PDF files. The earlier zero-record CSV Feedback issue remains a separate baseline condition.	PASS (non-zero data)
Administrator monitoring	BB-ADM-002; BB-ADM-004	Administrator monitoring was accessible and Course Monitoring displayed the dedicated YL-BBX-2026-0818 test course.	PASS
Peer review	BB-PRS-001	A legitimate peer-review record was not executed because the isolated workflow did not include a second peer student or presentation activity.	BLOCKED

Table 10 shows the 14 August black-box baseline retained the original 78-scenario plan and documented the cases that could be executed without inferring results from source code. The 18 August full-chain retest then used the isolated YL-BBX-2026-0818 dataset and authenticated student and lecturer sessions after targeted fixes. The final consolidated full-chain set comprised 17 scenarios: 15 PASS, one PARTIAL, and one BLOCKED. BB-TRACE-001 remained PARTIAL because all non-peer links were observed but peer evidence was unavailable, while BB-PRS-001 remained BLOCKED because the test environment did not include a second peer student or a valid peer/presentation workflow. The observable chain extended from OBE/Sub-CPMK configuration through course and enrollment, session/activity, student evidence submission, lecturer moderation, student revision or clarification, lecturer assessment, traceability analytics, CSV/PDF export, and administrator course monitoring. During retesting, BUG-YL-003 through BUG-YL-006 were fixed and passed the relevant retests; BUG-YL-001 remained open, while BUG-YL-002 remained partial because the non-zero CSV export path passed but zero-record handling still required explicit UX treatment. These results extend interface-level evidence but do not constitute usability, accessibility, educational-effectiveness, fairness, peer-reliability, privacy, or independent security validation.

3.2. Discussion

Y-Learn addresses a more specific problem than a general-purpose LMS. Moodle, Canvas, and Google Classroom already provide course organisation, assignments, grading, groups, rubrics, outcomes, and forms of peer feedback, although the available workflow depends on platform version and local configuration. The contribution of Y-Learn therefore does not rest on the presence of familiar learning-platform functions. Its emphasis is the continuity among member-attributable contribution evidence, lecturer moderation, student clarification, peer records, academic assessment, and outcome traceability within one Android course workflow. Because no comparative experiment was conducted, Table 11 identifies differences in design emphasis rather than evidence that Y-Learn performs better than established systems.

Table 11 is intended to locate Y-Learn within existing learning-platform practice, not to rank the products. The functions of Moodle, Canvas, and Google Classroom vary with platform version, institutional configuration, extensions, and local policy. For that reason, the comparison is limited to differences in workflow emphasis that are relevant to the

design problem addressed in this article. No conclusion can be drawn from the table about superior usability, reduced lecturer workload, stronger learning outcomes, or overall product quality.

Table 11. Workflow comparison with common LMS platforms

Dimension	Moodle	Canvas	Google Classroom	Y-Learn
Primary scope	Extensible institutional LMS	Institutional LMS and course environment	Classroom assignment and feedback environment	Native Android platform centred on course-level OBE and collaborative-assessment records
Course and assignment management	Core capability	Core capability	Core capability	Included within the broader course and OBE workflow
Group work	Available through course and group configuration	Available through group assignments and discussions	Usually arranged through assignments and shared documents	Working-group records and member context form a central part of the workflow
Peer assessment	Can be configured through activities such as Workshop	Can be configured through peer-review assignments or discussions	No dedicated group peer-review workflow is assumed here	Peer ratings or comments are retained as supporting evidence
Outcome or rubric linkage	Outcomes, competencies, and assessment configuration are available	Outcomes and rubrics are available	Rubrics and learning goals may be attached to assignments	Outcome, Sub-CPMK, activity, assessment, and achievement records are linked; the peer rubric remains descriptive
Individual contribution evidence	Depends on activity design,	Depends on assignments, discussions,	Submission history and comments	Member-attributable records retain group, activity,

Dimension	Moodle	Canvas	Google Classroom	Y-Learn
	plugins, and local use	analytics, and local configuration	are available, but group-contribution tracing is not the main design focus	description/reference, and time context
Current maturity	Established platform	Established platform	Established platform	Production version 3.4; earlier three-course operational exports and a full-chain black-box retest verified the non-peer workflow end-to-end, while peer review remained blocked by the isolated test setup; no comparative, usability, accessibility, peer-reliability, or causal-impact evaluation.

In Y-Learn, a contribution record is tied to a named student, a course activity, and a working group. Version 3.3 introduced a moderation step in which the lecturer may verify the entry, return it for revision, or reject it with a reason. The student can respond by revising the evidence or submitting a clarification, so the record retains the exchange rather than only the initial claim. Structured online peer assessment can support evaluative judgement when criteria and review activities are made explicit [23]. Riegler and Guest, however, note that collusion can weaken peer-assessment arrangements in group work [22]. Moderation strengthens documentary traceability, but conclusions about effort, authorship, quality, fairness, or individual achievement still require lecturer judgement and other records.

The operational exports show that peer-review records were generated at markedly different volumes across the three courses: 44 in PAR 102, 1,114 in SFT 106, and 1,755 in

SFT 210. Average ratings were also high, ranging from 4.41 to 4.96 on a five-point scale. Volume does not establish review quality, and a high average does not demonstrate agreement among reviewers or validity for summative grading. Structured criteria may support evaluative judgement [23], whereas comparisons of peer and instructor grading caution against treating the sources as interchangeable [24]. Friendship bias, retaliation, collusion, punishment of free riders, and score inflation remain plausible sources of distortion [25], [26]. The moderation and clarification workflow may also add lecturer work because disputed or incomplete contribution records require review rather than automatic resolution. Anonymity safeguards, anti-collusion controls, outlier procedures, inter-rater reliability, workload effects, and a formal institutional appeal policy have not been validated. Evidence from experiential peer-assessment settings further suggests that validity and learning benefits depend on the design and context of use [27].

Traceability in Y-Learn is not represented by a single score. The OBE Traceability Navigator groups filtered records by outcome or Sub-CPMK and connects them with activities, sessions, students, and graded evidence; CSV and PDF exports preserve the same summary for review. The three course reports demonstrate that these links were populated in operational settings, including 39 normalised evidence records in PAR 102, 42 in SFT 106, and 39 in SFT 210. The reports also expose incomplete processes, such as pending assessment or unanswered interaction records. These exports are generated by the same application under study and depend on the active filters and latest-evidence rule; consequently, the records demonstrate persistence and retrievability rather than independent observation of teaching quality. Motivation and other conditions remain relevant in digitally mediated learning [28], while technology integration has been associated with student engagement in higher-education settings [29]. Predictive-analytics research has also examined engagement patterns in e-learning environments [30]. Y-Learn does not use the available exports to predict student behaviour, and the recorded achievement values cannot be attributed to the application independently of teaching, assessment design, and moderation.

Privacy and access control are embedded in the academic workflow because Y-Learn stores contribution evidence, peer comments, clarification exchanges, and other performance-related records. The production arrangement verified in this study used Firebase Authentication, account-state checks, role-aware navigation, deployed Firestore

Security Rules, administrative audit collections, disabled cleartext traffic and backup, and Google Play signing and protection. Ten Emulator cases covered selected contribution-governance and denial paths, while the later black-box work added observable checks for authentication validation and administrator monitoring as well as course creation, enrollment, session/activity setup, student evidence submission, lecturer moderation, revision/clarification, lecturer assessment, traceability analytics, and CSV/PDF export. Peer review was not executed in the isolated full-chain workflow because a second peer student or valid peer/presentation setup was unavailable. Other paths, including parts of the user, grade, course, audit, and private-record boundary, were not comprehensively exercised. Records written by the client are therefore not described as immutable server-side logs, and callable Cloud Functions were not active under the earlier Spark-plan boundary. Firebase App Check and application-level Play Integrity were also inactive during the July technical evaluation. Wider institutional use would require explicit data-retention and deletion schedules, periodic access review, administrator-handover procedures, documented institutional ownership of academic records, and clearer governance for audit-log preservation. The current evidence supports a bounded implementation claim; it does not amount to security certification, privacy compliance, or institutional data-governance validation.

The study has several interpretive limits. Requirements were derived from practitioner experience and course documents, not from a formally sampled stakeholder study. Automated tests, Firestore Emulator checks, lint, builds, and console inspection were completed for the July evaluation, while the August black-box work remained developer-led. The final full-chain retest verified the non-peer workflow across separate student and lecturer sessions and an isolated test course, but peer-review evidence could not be executed because no second peer student or presentation activity was available. The three course reports were generated by the application and do not constitute a deduplicated participant dataset; their high scores and achievement indicators cannot be attributed causally to Y-Learn. No sampled SUS, UEQ, or TAM study, accessibility audit, peer-score reliability analysis, controlled classroom trial, assessment-fairness validation, formal threat model, penetration test, or privacy-impact assessment was conducted. The immediate technical agenda is therefore narrower than in the baseline cycle: complete the peer-review black-box path with a valid second-peer/presentation setup, broaden device and authorisation coverage, document authoritative grade paths, and establish

privacy, retention, ownership, and administrator-handover procedures. Subsequent research should examine student acceptance, lecturer workload, peer-score reliability, accessibility, comparison with LMS-supported workflows, and longitudinal OBE use with designs appropriate to each question.

4. CONCLUSION

Y-Learn has moved beyond a design proposal and is available as a native Android learning platform, with version 3.4 (versionCode 34) representing the current production release. The July technical evidence remains tied to version 3.3 and includes 24 passing Android unit tests, ten passing selected Firestore governance scenarios, completed lint and debug/release builds, production-console records, and three course exports containing 39, 42, and 39 normalised OBE evidence records. The subsequent version 3.4 black-box work progressed from a bounded baseline to a full-chain retest in which all non-peer links from OBE/Sub-CPMK setup through contribution, moderation, clarification, lecturer assessment, analytics, export, and administrator course monitoring were observed; peer review remained unexecuted because a valid second-peer/presentation setup was unavailable. Across these evidence sources, the central mechanism remains the continuity of records connecting outcomes, activities, students, groups, submitted evidence, moderation decisions, assessment, and time. The available records document use of the workflow in teaching settings, but do not establish an improvement in learning outcomes attributable to Y-Learn.

For lecturers, the practical value lies in having a mobile record trail that can inform review while leaving final academic judgement with the lecturer rather than an automatically calculated peer mark. Moderation and clarification make the treatment of submitted contribution evidence visible, and the traceability exports reveal both completed records and items awaiting attention. The current evaluation remains limited by practitioner-derived requirements, developer-led testing, the unexecuted peer-review path in the isolated full-chain retest, blocked device instrumentation in the earlier cycle, incomplete Firestore coverage, and the absence of formal usability, accessibility, peer-review reliability, assessment-fairness, classroom-impact, and independent security evaluation. Near-term work should complete the peer-review black-box path with an appropriate multi-student activity, broaden authorisation and physical-device testing,

document grade authority, and establish privacy, retention, administrator-handover, and institutional ownership procedures. Later studies should include SUS/UEQ/TAM or comparable acceptance measures, accessibility auditing, peer-score reliability analysis, lecturer-workload evaluation, controlled or longitudinal classroom research, comparison with LMS-supported workflows, and LMS/SIS/SSO integration where institutionally appropriate.

ACKNOWLEDGMENT

The author would like to thank Atma Jaya Catholic University of Indonesia for supporting the development and initial use of Y-Learn as an Outcome-Based Education learning platform.

CLARIFICATION AND ATTRIBUTION

Preparation of the manuscript followed Rector Decree of Atma Jaya Catholic University of Indonesia Number 4575/II/SK-OT.10.05/10/2025 on Academic Integrity in Producing Scientific Works and Rector Decree Number 4576/II/SK-OT.10.05/11/2025 on Guidelines for the Use of Artificial Intelligence (AI). The author retains responsibility for the research design, software development, production deployment, technical evaluation, analysis of course documents, interpretation, and approval of the final manuscript. AI-assisted tools were used only for editorial review of language, organisation, and internal consistency. Figure 1 was prepared from the DSR sequence and the verified production boundary; Figure 2 reproduces interfaces from the application; and Figure 3 depicts the implemented moderation, clarification, and peer-evidence workflow. Student names contained in the source course exports were omitted. External concepts, methods, and comparisons are acknowledged in the reference list. Responsibility for the accuracy, originality, interpretation, and integrity of the article remains with the author.

SOFTWARE AVAILABILITY

Y-Learn version 3.4 (versionCode 34) is the current production release distributed through Google Play under package com.yerik.y_learn. The application is used in learning and assessment activities, and its public listing is available at

https://play.google.com/store/apps/details?id=com.yerik.y_learn. The July 2026 technical verification and three course-level exports correspond to version 3.3, while the version 3.4 black-box evaluation progressed from the 14 August baseline to the 18 August full-chain retest using the isolated YL-BBX-2026-0818 dataset and a debug rebuild after fixes. The supplementary technical appendix, verified testing and evaluation report, course analytics exports, and black-box execution materials document the technology stack, production boundary, role and access model, deployed rules, automated checks, release evidence, contribution-governance workflow, OBE traceability, course-level evidence, and current validation limits.

REFERENCES

- [1] M. H. Romanowski, H. Alkhateeb, and Y. Chaaban, "Problematizing constructive alignment in higher education in gulf cooperation council countries," *Teach. High. Educ.*, vol. 29, no. 8, pp. 2042–2060, 2024, doi: 10.1080/13562517.2023.2180729.
- [2] P. D. Simon, J. Jiang, and L. K. Fryer, "Assessment of class participation in online and offline learning environments: a qualitative investigation of teachers' best practices and challenges," *High. Educ. Res. Dev.*, vol. 44, no. 5, pp. 1191–1208, 2025, doi: 10.1080/07294360.2025.2462024.
- [3] O. O. Adesina, O. A. Adesina, I. Adelopo, and G. A. Afrifa, "Managing group work: the impact of peer assessment on student engagement," *Account. Educ.*, vol. 32, no. 1, pp. 90–113, Jan. 2023, doi: 10.1080/09639284.2022.2034023.
- [4] P. Fleckney, J. Thompson, and P. Vaz-Serra, "Designing effective peer assessment processes in higher education: a systematic review," *High. Educ. Res. Dev.*, vol. 44, no. 2, pp. 386–401, 2025, doi: 10.1080/07294360.2024.2407083.
- [5] T. Ngoc Tuong Nguyen and D. Thi Kim Oanh, "Cooperative learning and its influences on student engagement," *Cogent Educ.*, vol. 12, no. 1, pp. 1–19, 2025, doi: 10.1080/2331186X.2025.2513414.
- [6] B. A. McKenna, J. B. Wehr, and P. M. Kopittke, "Quantifying online engagement at three levels of undergraduate study," *Cogent Educ.*, vol. 11, no. 1, pp. 1–16, 2024, doi: 10.1080/2331186X.2024.2345939.

- [7] M. J. Bonilla-Priego, M. Fernández-Giordano, and M. R. Pacheco-Olivares, "Evaluating the success of a learning management system for supplemental learning in accounting," *Account. Educ.*, vol. 35, no. 2, pp. 198–223, 2024, doi: 10.1080/09639284.2024.2424269.
- [8] V. Maphosa, "Enhancing authentic learning in a rural university: exploring student perceptions of Moodle as a technology-enabled platform," *Cogent Educ.*, vol. 11, no. 1, pp. 1–13, 2024, doi: 10.1080/2331186X.2024.2410096.
- [9] P. Levrai, "What to assess when assessing collaborative assignments: the practitioner perspective," *Assess. Eval. High. Educ.*, vol. 0, no. 0, pp. 1–18, 2025, doi: 10.1080/02602938.2025.2571219.
- [10] T. Little, P. Dawson, D. Boud, and J. Tai, "What does it take to provide effective peer feedback?," *Assess. Eval. High. Educ.*, vol. 50, no. 5, pp. 775–786, 2025, doi: 10.1080/02602938.2025.2475059.
- [11] E. Dwi Lestari and R. Riatun, "Unveiling key factors for the continuation of E-learning adoption in blended learning environments within Indonesian higher education during the era of the 'new normal,'" *Cogent Educ.*, vol. 11, no. 1, pp. 1–17, 2024, doi: 10.1080/2331186X.2024.2428871.
- [12] Rulinawaty *et al.*, "Investigating the influence of the updated DeLone and McLean information system (IS) success model on the effectiveness of learning management system (LMS) implementation," *Cogent Educ.*, vol. 11, no. 1, p. 2365611, Dec. 2024, doi: 10.1080/2331186X.2024.2365611.
- [13] W. Kayanja, M. Kyambade, and T. Kiggundu, "Exploring digital transformation in higher education setting: the shift to fully automated and paperless systems," *Cogent Educ.*, vol. 12, no. 1, pp. 1–15, 2025, doi: 10.1080/2331186X.2025.2489800.
- [14] R. Ismailova *et al.*, "A comparative analysis of service quality and student perceptions in Kyrgyzstan's higher education institutions," *Cogent Educ.*, vol. 12, no. 1, pp. 1–17, 2025, doi: 10.1080/2331186X.2025.2502202.
- [15] A. K. Carstensen and J. Bernhard, "Design science research—a powerful tool for improving methods in engineering education research," *Eur. J. Eng. Educ.*, vol. 44, no. 1–2, pp. 85–102, 2019, doi: 10.1080/03043797.2018.1498459.
- [16] J. Venable, J. Pries-Heje, and R. Baskerville, "FEDS: A Framework for Evaluation in Design Science Research," *Eur. J. Inf. Syst.*, vol. 25, no. 1, pp. 77–89, 2016, doi: 10.1057/ejis.2014.36.

- [17] A. Palmquist, "Stakeholders' design preferences for instructional gamification," *Behav. Inf. Technol.*, vol. 43, no. 12, pp. 2717–2749, 2024, doi: 10.1080/0144929X.2023.2255905.
- [18] J. Edwards *et al.*, "A Wizard of Oz methodology for designing future learning spaces," *High. Educ. Res. Dev.*, vol. 44, no. 1, pp. 147–162, 2025, doi: 10.1080/07294360.2024.2429433.
- [19] S. Val and A. Quintas, "Key performance indicators for optimizing academic performance and course design in online educational platforms," *Cogent Educ.*, vol. 12, no. 1, pp. 1–24, 2025, doi: 10.1080/2331186X.2025.2529420.
- [20] J. Heil and D. Ifenthaler, "Effects of teacher, peer and self-feedback on student improvement in online assessment: the role of individuals' presumptions and feedback literacy," *Assess. Eval. High. Educ.*, vol. 51, no. 2, pp. 281–300, 2025, doi: 10.1080/02602938.2025.2530452.
- [21] S. de Freitas, G. J. Curtis, and S. M. Ritchie, "A multi-layered evaluation of university-wide education interventions to improve student engagement and retention," *J. Furth. High. Educ.*, vol. 49, no. 10, pp. 1348–1362, 2025, doi: 10.1080/0309877X.2025.2560999.
- [22] R. Riegler and J. Guest, "Does widespread collusion undermine the case for using peer-assessment schemes with assessed group work?," *Stud. High. Educ.*, vol. 51, no. 2, pp. 295–308, 2025, doi: 10.1080/03075079.2025.2465687.
- [23] E. Cano García, L. Halbaut Bellowa, L. Martins Gironelli, and L. Lluch Molins, "Online peer assessment in Galenic Pharmacy: enhancing evaluative judgement in higher education," *Med. Educ. Online*, vol. 29, no. 1, pp. 1–11, 2024, doi: 10.1080/10872981.2024.2409487.
- [24] M. Usher and M. Faraon, "Who grades best? Comparing ChatGPT, peer, and instructor evaluations across varying levels of student project quality," *Assess. Eval. High. Educ.*, vol. 0, no. 0, pp. 1–20, 2025, doi: 10.1080/02602938.2025.2588682.
- [25] A. Rasooli, J. Turner, T. Varga-Atkins, E. Pitt, S. Asgari, and W. Moindrot, "Students' perceptions of fairness in groupwork assessment: validity evidence for peer assessment fairness instrument," *Assess. Eval. High. Educ.*, vol. 50, no. 1, pp. 111–126, 2025, doi: 10.1080/02602938.2024.2354510.

- [26] E. Panadero and M. Alqassab, "An empirical review of anonymity effects in peer assessment, peer feedback, peer review, peer evaluation and peer grading," *Assess. Eval. High. Educ.*, vol. 44, no. 8, pp. 1253–1278, 2019, doi: 10.1080/02602938.2019.1600186.
- [27] J. Käpylä and M. Palvalin, "Integrating peer assessment into experiential learning: the validity and learning benefits of peer assessment in reading circle discussions," *Assess. Eval. High. Educ.*, vol. 50, no. 8, pp. 1253–1276, 2025, doi: 10.1080/02602938.2025.2566659.
- [28] I. B. P. Puja, "The mediating role of online learning motivation in the influence of service quality, social media usage, and pedagogical teaching competence of teachers on student learning satisfaction," *Cogent Soc. Sci.*, vol. 10, no. 1, pp. 1–29, 2024, doi: 10.1080/23311886.2024.2396934.
- [29] J. R. Cabalbag, "Technology integration: impact to students' learning engagement in higher education institutions," *Cogent Educ.*, vol. 12, no. 1, pp. 1–13, 2025, doi: 10.1080/2331186X.2025.2581410.
- [30] V. Mikić, G. Keković, K. Mangaroska, M. Ilić, L. Kopanja, and B. Vesin, "Exploring Learner Engagement in E-Learning Environments: A Predictive Analytics Perspective," *Int. J. Hum. Comput. Interact.*, vol. 42, no. 6, pp. 3896–3919, 2025, doi: 10.1080/10447318.2025.2535530.