International Journal of Graphic Design (IJGD)
Volume 3 No. 1 May 2025
E-ISSN: 2987-9434; P-ISSN: 2988-0343, Page 175-195
DOI: https://doi.org/10.51903/ijgd.v3i1.3618
Available online at: https://journal.stekom.ac.id/index.php/ijgd

Adaptive User Interface Design for Volleyball Learning Apps: Empirical Evidence from Google Play Reviews and Mobile Screen Analysis

Jubin Zhang[*1]
[1]Department of Physical Education, North China Institute of Aerospace Engineering, Langfang 065000, China
Email Address: jz0801@outlook.com

Abstract. *Adaptive sports-learning apps should not expose the same interface to a first-session learner and to a self-directed advanced player. This paper develops an empirical design basis for a volleyball learning app that adapts information density, feedback granularity, and interaction complexity to learner level. We combined two public datasets: a Google Play review corpus and the MASC mobile UI dataset. From 63,340 reviews we constructed a learning-and-training subset of 7,328 reviews from 1,702 apps and benchmarked five sentiment models using grouped 3-fold cross-validation. The best review classifier was a Linear SVM (accuracy = 0.786, macro-F1 = 0.548). We then compared LDA, NMF, and KMeans topic models on 1,527 negative reviews; LDA with four topics produced the best coherence. The resulting themes were core reliability and login/access friction (34.1%), control quality and monetization friction (28.0%), update/privacy/state-persistence failures (23.1%), and support, billing, and account recovery (14.7%). Novice-coded complaints were 3.74 times more likely than advanced-coded complaints to focus on support and recovery, whereas advanced-coded complaints over-indexed on control and monetization. On MASC, a fusion Linear SVM combining keywords and numeric UI features achieved a macro-F1 of 0.938 for 10-class screen prediction. Complexity clustering produced two stable screen regimes: a low-complexity cluster dominated by Welcome, Login, and Home screens and a high-complexity cluster dominated by List, Menu, and Search screens. Based on these results, the paper specifies a two-mode adaptive volleyball UI blueprint: a guided beginner mode and a dense expert mode. The study shows how reproducible review mining and screen analytics can directly inform adaptive UI decisions for skill-learning applications.*

Keywords: *Adaptive User Interface, Volleyball Learning App, Learner Modeling, Mobile Screen Classification, Human-Computer Interaction*

INTRODUCTION

Mobile skill-learning applications now deliver demonstration videos, drill schedules, progress trackers, and self-assessment tools to learners outside formal coaching environments. In volleyball, this promise is especially attractive because serving, passing, setting, blocking, and approach mechanics require repeated observation, focused attention, and timely corrective feedback. However, a static mobile interface is rarely optimal for all learners. A novice usually needs a highly guided flow, sparse information, and clear recovery from mistakes, whereas an advanced learner expects faster navigation, denser analytics, and fine-grained control over drill composition. Feedback research and motor-learning theory show that guidance is most effective when it is timely, interpretable, and matched to the learner's current state rather than delivered as a uniform layer to everyone (Hattie & Timperley, 2007; Shute, 2008; Wulf & Lewthwaite, 2016).

This challenge aligns with a long-standing problem in human-computer interaction: how to personalize an interface without sacrificing predictability. Adaptive systems can lower interaction costs and direct attention toward relevant content, yet they also risk disorienting users

Received: March 2026; Revised: March 2025; Accepted: April 2026; Published: May 2025
*Corresponding author, jz0801@outlook.com

if the adaptation is opaque or unstable (Jameson, 2003; Findlater & McGrenere, 2004; Gajos et al., 2008). For learning applications, the problem is sharper because interface choices influence not only efficiency but also the quality of practice, the pacing of feedback, and the amount of cognitive load imposed during skill acquisition (Sweller, 1988; Mayer, 2009). A volleyball learning app therefore needs more than a content library. It needs an adaptive presentation layer that changes what is shown, how much is shown, and how directly corrective feedback is expressed.

In practice, designers often make these adaptation decisions from intuition, small focus groups, or benchmark screenshots. Those inputs are useful, but they do not provide a broad empirical map of what learners complain about, which interaction patterns create friction, or how screen structure varies across simple and complex mobile interfaces. App store reviews are a scalable source of design evidence because they capture unsolicited user feedback at the point of breakdown or satisfaction (Pagano & Maalej, 2013; Fu et al., 2013; Chen et al., 2014). Public UI datasets are another scalable source because they expose recurring mobile screen structures that can be measured, classified, and linked to information density or control complexity (Deka et al., 2017; Zaki & Abdallah, 2023).

This paper treats adaptive UI design for a volleyball learning app as an evidence-synthesis problem. Rather than claiming to have already deployed a volleyball app, the study first asks what empirical signals are available in adjacent mobile-learning and training apps and then translates those signals into a concrete adaptive design specification. The study addressed three questions: what learner-facing themes and pain points appear in learning-and-training app reviews; which mobile screen patterns distinguish lower from higher interaction complexity; and how can those two evidence streams be merged into a novice mode and an advanced mode for a volleyball learning interface?

The contributions are fourfold. First, the paper constructs a reproducible learning-and-training review subset from a public Google Play corpus and benchmarks multiple text models instead of relying on illustrative review anecdotes. Second, it identifies stable complaint themes and measures how novice-coded and advanced-coded users differ in the problems they emphasize. Third, it benchmarks multiple screen-classification models on MASC and derives empirically grounded low- and high-complexity screen regimes. Fourth, it combines those results into a concise adaptive UI blueprint for volleyball learning that specifies changes in density, feedback granularity, interaction complexity, and microcopy.

LITERATURE REVIEW

Research on adaptive interfaces has consistently shown that personalization is most useful when the system models user differences in a way that is both behaviorally meaningful and predictable to the user (Dinata et al., 2025; Mendez & Okafor, 2026; Petrova & Watanabe, 2025). Foundational work on adaptive hypermedia and adaptive educational systems framed learner modeling as the core mechanism for deciding what content, sequence, or support to show next (Brusilovsky & Millán, 2007). HCI research later examined specific interface components such as menus and graphical controls, showing that adaptation can lower navigation effort but must preserve an intelligible mapping between user goals and system behavior (Findlater & McGrenere, 2004; Gajos et al., 2006; Gajos et al., 2008). These results are directly relevant to a volleyball learning app because progression from novice to advanced use is not merely a content issue; it is also an interface issue about how much structure, choice, and explanation the learner sees at each stage.

Learner modeling in educational technology further clarifies how adaptation should be timed. Knowledge tracing and later data-mining work treat learner state as something inferable from observable behavior rather than something designers must guess in advance (Corbett & Anderson, 1995; Koedinger et al., 2015). Once learner state is estimated, feedback design becomes crucial. (Hattie & Timperley, 2007) distinguish feedback about task performance, process, self-regulation, and self-concept, while (Shute, 2008) argues that effective formative feedback is specific, timely, and tied to current performance. In motor-learning settings, overloading early learners with too many corrections can hinder performance, whereas well-timed attentional cues can improve skill acquisition (Wulf & Lewthwaite, 2016). These findings support the idea that a volleyball app should vary not only the amount of information but also the grain of corrective feedback, for example by presenting one cue at a time to novices while surfacing session-level metrics for advanced learners.

A second strand of literature concerns mobile app reviews as a design-evidence source (Kuhn et al., 2024). User reviews have been studied as a large-scale repository of complaints, feature requests, and usability issues that developers can mine for actionable intelligence (Pagano & Maalej, 2013; Fu et al., 2013; Chen et al., 2014; Martin et al., 2017). Requirements-engineering work has gone further by showing that review mining can systematically support backlog refinement and product decisions, particularly when the analysis distinguishes bug reports, experience complaints, and requests for missing functionality (Maalej et al., 2016). For adaptive UI design, review mining is valuable because it reveals not only what functionality users want but also which parts of the interface become brittle under real-world conditions, such as sign-in

failures, confusing control flows, and disruptive monetization. Those signals are important when the target application, here a volleyball learning app, must remain usable during repeated practice sessions rather than one-off exploration.

A third strand concerns how mobile interfaces themselves are represented and studied. Rico made large-scale data-driven UI analysis possible by providing a broad corpus of mobile screenshots and view hierarchies (Deka et al., 2017). More recent datasets extend this direction through manual screen categorization or higher-level semantic annotations, including MASC for screen-type classification and UEyes for attention-grounded UI analysis (Zaki & Abdallah, 2023; Jiang et al., 2024). Attention-oriented resources and methods, such as BubbleView, show that visual prominence and gaze allocation can be approximated or measured at scale (Kim et al., 2017). For the present study, attention-grounded work is conceptually important because adaptive UIs often succeed by guiding scarce visual attention, yet a structured screen dataset is more directly useful for modeling information density and interaction complexity. The empirical screen analysis (Chen & Chan, 2023) in this paper therefore uses MASC rather than an eye-tracking corpus because the design target is adaptive layout and control configuration, not fixation prediction.

Finally, explanation and intelligibility research highlights that adaptation is more acceptable when users understand why the interface changed. Toolkits for intelligibility and work on explanatory debugging show that systems gain trust when they expose a concise rationale, recovery path, or override mechanism instead of silently changing behavior (Lim & Dey, 2010; Kulesza et al., 2015). More recent HCI discussions around explainable and accountable systems make the same point at a broader scale: personalization should be visible enough to preserve user agency (Abdul et al., 2018). This literature matters for sports-learning apps because beginners need reassurance when the system simplifies the interface, and advanced users need to know why additional controls or analytics have become available. The gap in prior work is that review mining, screen-structure analysis, and adaptive-learning theory are often studied separately. The present paper closes that gap by combining them in one reproducible design pipeline.

METHODS

The study used two public datasets. The first was the Google Play Store review corpus released on Zenodo by (Marrero, 2019). The merged review file contained 63,340 rows with review text, current star rating, app identifier, a numeric category code, and app-level metadata. The second was MASC, a structured mobile screen dataset derived from Rico and organized into 10 screen classes with numeric UI descriptors and keywords for each screen (Deka et al., 2017;

Zaki & Abdallah, 2023). Table 1 summarizes the raw inputs and their role in the study, while Figure 1 shows the overall workflow.

Table 1. Dataset Summary

Dataset component	Rows	Columns	Modalities	Role in study
Google Play review corpus (full)	63340	12	App reviews + app metadata	Original merged review file from Zenodo
Learning-and-training review subset	7328	16	Filtered review text, ratings, app IDs, category code, and installs	Heuristic subset for adaptive-learning design analysis
MASC features	7065	11	Numeric UI descriptors + keywords	Structured mobile screen features
MASC labels	7065	2	Screen class labels	10 screen categories

A learning-and-training subset was constructed from the Google Play corpus in order to focus the analysis on apps that more closely resemble a volleyball learning scenario. The filtering rule retained reviews that belonged to category codes 3 or 4 or that contained at least one token from a learning-and-training lexicon: beginner, tutorial, guide, learn, lesson, easy, simple, help, start, step, explain, practice, advanced, pro, expert, detail, analytics, stat, custom, customize, shortcut, precise, fine, control, settings, training, coach, video, progress, feedback, or drill. Reviews shorter than five tokens were removed, and exact duplicates by app identifier and normalized text were dropped. The final subset contained 7,328 reviews from 1,702 distinct apps. Manual inspection of the most frequent app identifiers in category codes 3 and 4 showed that those codes were dominated by wellness/training and educational apps, which made them a practical proxy for adjacent learning contexts even though the original file did not expose human-readable category labels.

Table 2. Review Subset Preprocessing and Distribution Statistics

Statistic	Value
Filtered subset size	7328
Distinct apps	1702
Average tokens per review	17.177
Positive reviews	4988
Neutral reviews	624
Negative reviews	1716
Novice-cue reviews	3106
Advanced-cue reviews	959
Mixed-cue reviews	189
Neither cue	3074

The review analysis operationalized sentiment directly from star ratings so that every observation had a reproducible label: one or two stars were negative, three stars were neutral, and four or five stars were positive. This mapping is common in review mining because it avoids manual relabeling while preserving the user's explicit evaluative signal (Fu et al., 2013; Maalej

et al., 2016). To study learner-level differences, the review text was also tagged with a novice-cue lexicon (beginner, tutorial, guide, learn, lesson, easy, simple, help, start, step, explain, practice) and an advanced-cue lexicon (advanced, pro, expert, detail, analytics, stat, custom, customize, shortcut, precise, fine, control, settings). Each review was then assigned to one of four segments: novice, advanced, mixed, or neither.

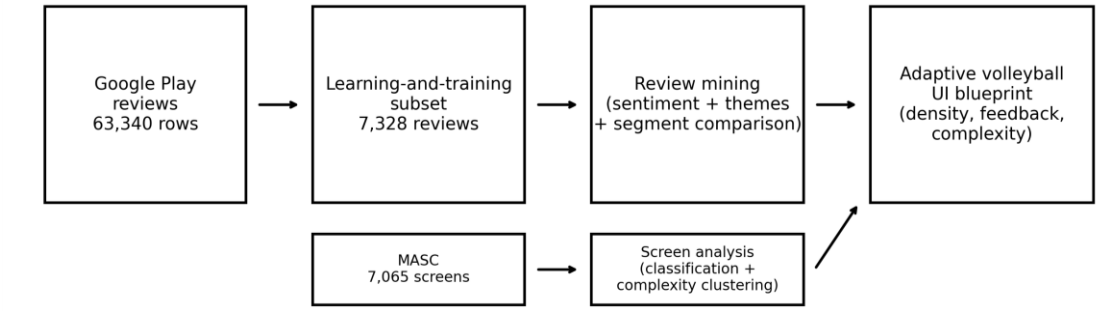

Figure 1. Empirical Workflow from Raw Datasets to the Adaptive Volleyball UI Blueprint

Review sentiment models were benchmarked with grouped 3-fold cross-validation in which the grouping variable was app identifier. Grouping by app prevented near-duplicate reviews from the same app appearing in both train and test folds. Text was vectorized with TF-IDF unigrams and bigrams using min_df = 3, max_df = 0.9, and max_features = 12,000. Five models were compared: a majority baseline, Multinomial Naive Bayes, logistic regression, a linear support vector machine, and a linear SGD classifier. Accuracy, macro-F1, and weighted F1 were recorded on every fold, and mean and standard deviation were reported. A fixed random seed of 42 was used throughout.

Issue theme mining focused on the 1,527 negative reviews that still contained at least five alphabetic tokens after tokenization and stop-word removal. Three families of unsupervised models were compared across k = 4 to 8 topics: LDA on count vectors, NMF on TF-IDF vectors, and KMeans on TF-IDF vectors. The comparison used u_mass coherence and topic diversity because the objective was not merely compressive clustering but interpretable theme discovery. After the best configuration was selected, every negative review was assigned to its highest-membership topic. Topic labels were then written after inspecting the top ten terms and the highest-membership reviews for each topic. This labeling step did not change any topic assignment; it only converted fixed topic IDs into readable design themes.

The segment comparison used only negative reviews that were tagged as novice or advanced. For each discovered theme, a 2 by 2 chi-square test compared the count of theme mentions versus all other themes across the two segments. The analysis reported raw counts, within-segment shares, the novice-to-advanced share ratio, and the chi-square p-value. A ratio

larger than 1.0 indicated that the theme was relatively more common among novice-coded complaints, while a ratio smaller than 1.0 indicated greater concentration among advanced-coded complaints. A separate n-gram analysis extracted frequent bi-grams and tri-grams from these segment-specific complaints and translated them into concise microcopy revisions.

The MASC analysis used the public feature table and label file for 7,065 screens. The numeric feature set was expanded with three derived totals: total clickable elements, total text fields, and total swipeable elements. Screen-type prediction used stratified 3-fold cross-validation with five models: a majority baseline, numeric logistic regression, numeric random forest, keyword-only linear SVM, and a fusion linear SVM that concatenated standardized numeric features with TF-IDF keyword vectors. This comparison distinguished between semantic signals carried by the keyword field and structural signals carried by UI density features.

Complexity analysis treated the numeric screen features as indicators of interaction load. The numeric features were standardized, and the first principal component was used as a continuous complexity score, oriented so that larger values corresponded to more interactive elements. KMeans clustering was then evaluated for k = 2 to 6 with sampled silhouette scores. The selected solution was labeled by ordering clusters from the lowest to the highest mean complexity score. The final design synthesis did not invent new evidence. It mapped the empirically observed novice and advanced complaint patterns onto the low- and high-complexity screen regimes to create two adaptive UI modes for a volleyball learning app. The replication package included the raw datasets, derived tables, figures, and all analysis scripts.

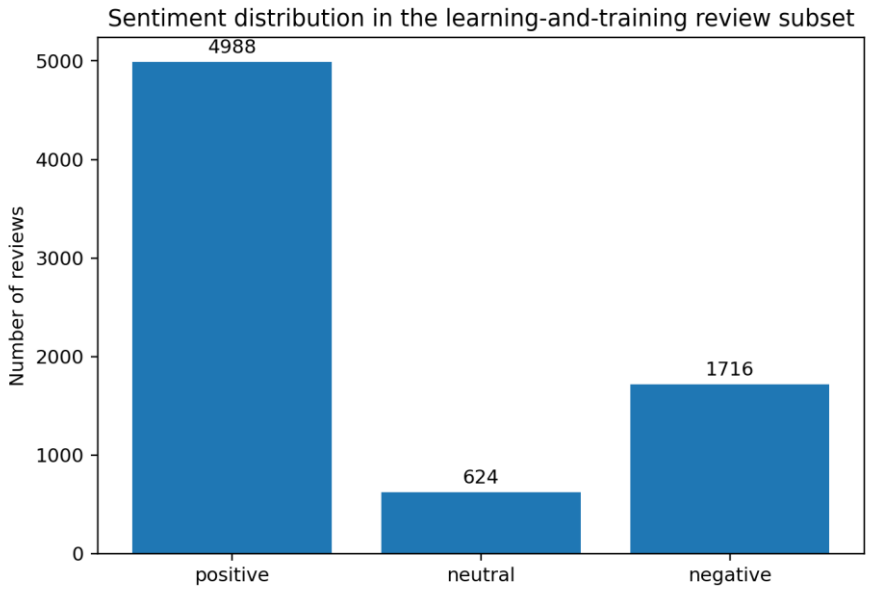

Figure 2. Sentiment Distribution in the Learning-and-Training Review Subset

RESULTS

The filtered review corpus remained broad enough for comparative analysis while being specific enough to speak to learning-oriented app behavior. Table 2 shows that the subset contained 7,328 reviews from 1,702 apps, with an average of 17.18 tokens per review. The distribution was positive-skewed: 4,988 reviews (68.1%) were positive, 624 (8.5%) were neutral, and 1,716 (23.4%) were negative. Segment tagging also produced a substantial learner-oriented split, with 3,106 novice-coded reviews and 959 advanced-coded reviews. Figure 2 visualizes the sentiment distribution.

Table 3. Review Sentiment Classification Results (Grouped 3-Fold Cross-Validation)

Model	accuracy_mean	accuracy_sd	macro_f1_mean	macro_f1_sd	weighted_f1_mean	weighted_f1_sd
Linear SVM	0.786	0.002	0.548	0.004	0.764	0.002
Logistic Regression	0.766	0.002	0.548	0.013	0.755	0.004
SGD Linear SVM	0.748	0.002	0.542	0.014	0.742	0.002
Multinomial NB	0.791	0.006	0.508	0.008	0.748	0.007
Majority	0.681	0	0.27	0	0.551	0

Table 3 reports the review-sentiment benchmarks. The Linear SVM delivered the best macro-F1 (0.548 ± 0.004) together with a mean accuracy of 0.786 ± 0.002. Logistic regression reached the same rounded macro-F1 (0.548) but at lower mean accuracy (0.766). Multinomial Naive Bayes produced the highest overall accuracy (0.791) but a clearly lower macro-F1 (0.508), which indicates stronger bias toward the majority positive class. The majority baseline reached only 0.270 macro-F1, so every trained model improved substantially over a trivial classifier.

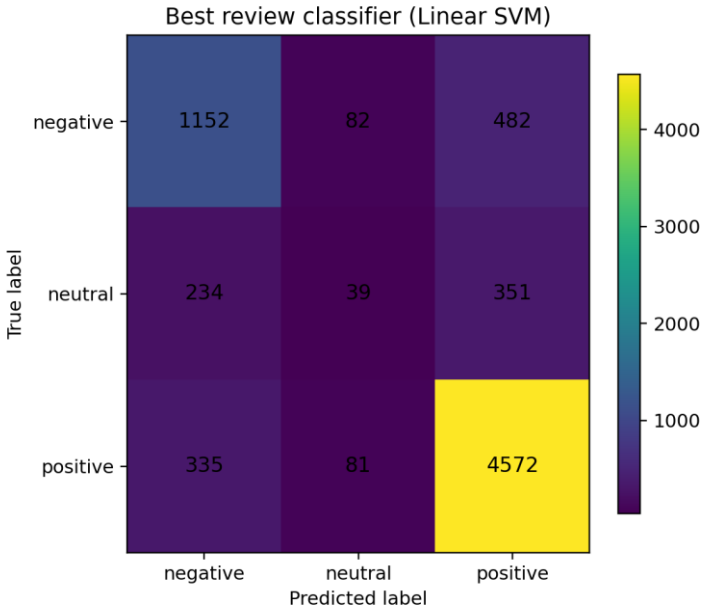

Figure 3. Confusion Matrix for the Best Review Sentiment Classifier (Linear SVM)

Figure 3 clarifies the error pattern of the best review model. The confusion matrix showed that the best model correctly identified 1,152 negative, 39 neutral, and 4,572 positive reviews. The hardest class was neutral, for which 351 of 624 instances were predicted as positive. The strong asymmetry of the confusion matrix shows that the model captured polarized praise and criticism more reliably than the middle category. That behavior is consistent with app-store feedback patterns, where three-star reviews often mix praise with disappointment in ways that blur a single sentiment label. The result is still useful for design because the adaptive UI problem in this paper is driven primarily by reliably detected complaints and positive confirmations rather than by fine distinctions inside mild sentiment.

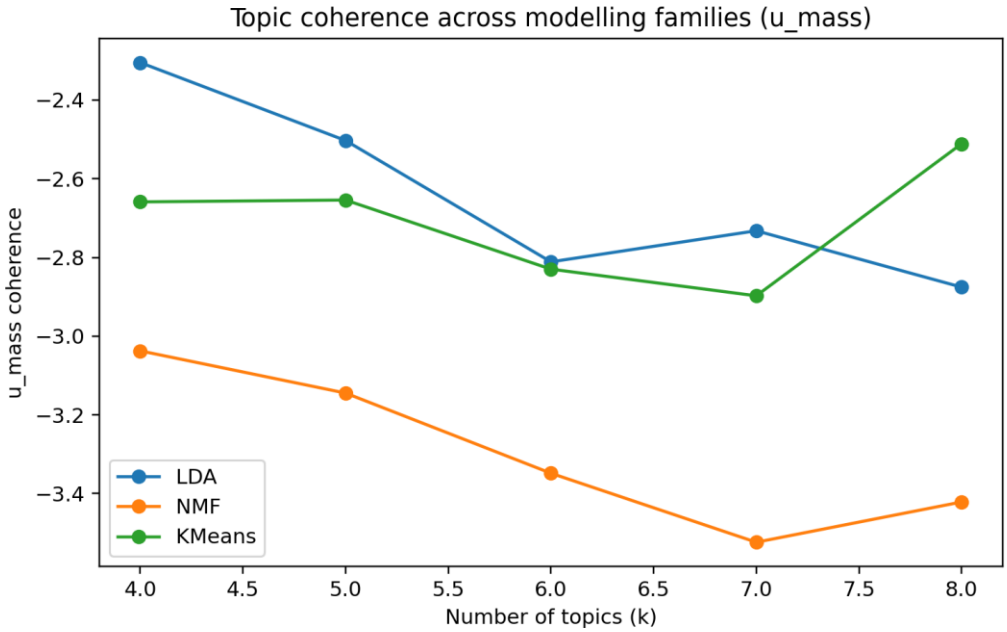

Figure 4. Coherence Comparison for LDA, NMF, and KMeans Across Topic Counts

Table 4. Topic-Model Comparison on Negative Reviews

Family	k	Coherence	Diversity	fit_Stat
LDA	4	-2.305	0.775	550
LDA	5	-2.503	0.72	572
KMeans	8	-2.511	0.625	1421
KMeans	5	-2.654	0.72	1440
KMeans	4	-2.659	0.675	1448
LDA	7	-2.732	0.629	607
LDA	6	-2.811	0.717	596
KMeans	6	-2.83	0.65	1433
LDA	8	-2.876	0.612	630
KMeans	7	-2.898	0.7	1428
NMF	4	-3.037	0.95	37.804
NMF	5	-3.145	0.92	37.653
NMF	6	-3.349	0.883	37.512
NMF	8	-3.422	0.85	37.249
NMF	7	-3.524	0.871	37.379

Theme discovery also produced a clear experimental winner. Table 4 and Figure 4 show that LDA with four topics achieved the strongest coherence (-2.305) together with high topic diversity (0.775). The best NMF configuration reached only -3.037 coherence, so the selected LDA model outperformed it by 0.732 coherence units. The KMeans variants were stronger than NMF but still weaker than the selected LDA solution. Because the evaluation covered three model families and five topic counts, the final themes were based on measured rather than illustrative model choice.

Table 5. Final Complaint Themes from the Best Topic Model

topic_id	Theme	Reviews	Share	mean_rating	mean_tokens	top_terms	design_implication
1	Core reability	521	0.341	1.29	20.39	app, work, time, won't, doesn't, fine, don't, use, fix, let	Reduce sign-in friction, preserve state, and use plain-language error recovery.
2	Control quality and monetizati on friction	428	0.28	1.308	23.521	game, control, play, start, like, time, make, good, don't, way	Sequence advanced controls gradually and keep monetisation out of the primary practice flow.
0	Update, privacy, and state-persistence failures	353	0.231	1.3	26.249	app, update, video, version, want, fix, use, like, dont, crash	Show explicit retry/download states, clarify permission requests, and preserve session progress.
3	Support, billing, and account recovery	225	0.147	1.307	24.516	app, help, account, phone, time, ive, google, email, tried, data	Expose recovery, refund, and support actions where failure occurs instead of hiding them in settings.

Table 5 presents the four final complaint themes. The largest theme was core reliability and login/access friction, which accounted for 521 reviews (34.1%). This theme grouped failures such as sessions not loading, repeated sign-in problems, blank screens, and broken core functions. The second theme was control quality and monetization friction with 428 reviews (28.0%). These reviews emphasized awkward control schemes, obstructive monetization, and frustration with having to watch ads or pay before basic progress felt possible. The third theme, update, privacy, and state-persistence failures, contained 353 reviews (23.1%) and concentrated on regressions after updates, suspicious permissions, and lost state. The fourth theme, support, billing, and account recovery, contained 225 reviews (14.7%) and focused on refunds, account transfer, hidden recovery paths, and unresponsive support. The design implications in Table 5 translate

each theme into a specific UI action instead of leaving the findings at the level of complaint categories.

The novice-versus-advanced comparison sharpened the design interpretation. Table 6 shows that support, billing, and account recovery was far more concentrated among novice-coded complaints: the novice share was 19.1% versus only 5.1% for advanced-coded complaints, a ratio of 3.74 with $p < .001$. In contrast, control quality and monetization friction was more concentrated among advanced-coded complaints: the novice share was 34.8% whereas the advanced share was 46.4% (ratio = 0.749, p = 0.002). Core reliability and login/access friction and update/privacy/state-persistence failures did not differ significantly between the two segments, which means they are universal design concerns rather than level-specific ones.

Table 6. Novice-Versus-Advanced Comparison for Negative Complaint Themes

Theme	novice _count	Novice _share	Advanced _count	Advanced _share	novice_to _advanced _ratio	chi2_p _value
billing, and account recovery	88	0.191	15	0.0512	3.737	8.97e-08
Update, privacy, and state-persistence failures	100	0.217	58	0.198	1.098	0.584
Core reliability and login/access friction	112	0.243	84	0.287	0.849	0.218
Control quality and monetization friction	160	0.348	136	0.464	0.749	0.00187

Table 7 operationalizes the complaint themes into microcopy revisions. The observed phrases were extracted from recurring n-grams rather than invented from memory. For example, the phrase “customer service” mapped to a recovery-oriented message that places restore, support, and billing actions on the same screen; “play game” mapped to a message that explicitly delays expert controls and premium surfaces until after an initial practice success. These revisions are not A/B-tested claims of performance improvement. They are concise, evidence-traceable copy proposals linked to the measured complaint structure.

Table 7. Microcopy Revisions Derived from Recurrent Complaint Phrases

Theme	**observed_phrase**	**revised_microcopy**
Core reliability and login/access friction	work fine	Sign in once to sync drills and progress. We saved your last lesson and will reopen it automatically.
Control quality and monetization friction	play game	Start with simple controls. Advanced settings and premium features appear only after the first completed drill.
Update, privacy, and state-persistence failures	pro version	Video quality adjusted for your connection. Retry now or download this drill for offline practice.
Support, billing, and account recovery	customer service	Need help? Restore progress, request support, or review billing from this screen without leaving practice.

The screen-structure analysis on MASC produced equally strong empirical separation. Table 8 shows large descriptive differences across screen classes. Welcome screens had the smallest mean number of clickable elements (2.469) and general elements (2.802), which makes them structurally suitable for novice entry points. Menu screens were much denser, with 9.736 clickable elements and 15.679 general elements on average, while List screens were notably swipe-heavy (3.598). Search screens also carried more interaction load than Welcome or Login screens, particularly through a combination of clickable elements and swipeable content.

Table 8. MASC Screen-Class Distribution and Mean Structural Features

screen_class	count	total_clickable	total_textfields	general_elements	total_swipeable	has_nav_drawer
Welcome	1084	2.469	0.043	2.802	0.164	0.034
List	960	5.543	0.317	10.217	3.598	0.17
Login	889	3.916	2.099	6.06	0.029	0.045
Home	866	4.423	0.077	5.493	0.717	0.229
Search	725	5.032	0.739	7.234	1.181	0.254
Setting	629	3.509	0.407	7.576	3.041	0.107
Menu	557	9.736	0.25	15.679	4.652	0.106
Profile	526	5.196	1.057	7.106	0.622	0.114
Map	500	5.686	0.238	6.248	0.226	0.272
Chat	329	5.587	0.362	7.69	1.328	0.158

Table 9 and Figure 5 report the screen-classification benchmarks. The fusion Linear SVM achieved the best mean macro-F1 (0.938) and mean accuracy (0.929). The keyword-only Linear SVM was nearly identical (0.938 macro-F1), which shows that the semantic keyword channel carried most of the information needed for exact screen-type prediction. Numeric-only models were much weaker: the random forest reached 0.422 macro-F1, and numeric logistic regression reached 0.351. The gap between the best fusion model and the best numeric-only model was therefore 0.516 macro-F1 points. This result is important because it separates two roles of UI analytics: semantic screen recognition is driven strongly by textual or symbolic cues, while numeric structure is more useful for measuring complexity.

Table 9. MASC Screen-Class Prediction Results (3-Fold Cross-Validation)

Model	accuracy_mean	accuracy_sd	macro_f1_mean	macro_f1_sd	Weighted f1_mean	Weighted f1_sd
Fusion Linear SVM	0.929	0.005	0.938	0.004	0.93	0.004
Keyword Linear SVM	0.926	0.005	0.938	0.004	0.929	0.004
Numeric Random Forest	0.465	0.015	0.422	0.018	0.45	0.014
Numeric Logistic Regression	0.423	0.003	0.351	0.006	0.388	0.004
Majority	0.153	0	0.027	0	0.041	0

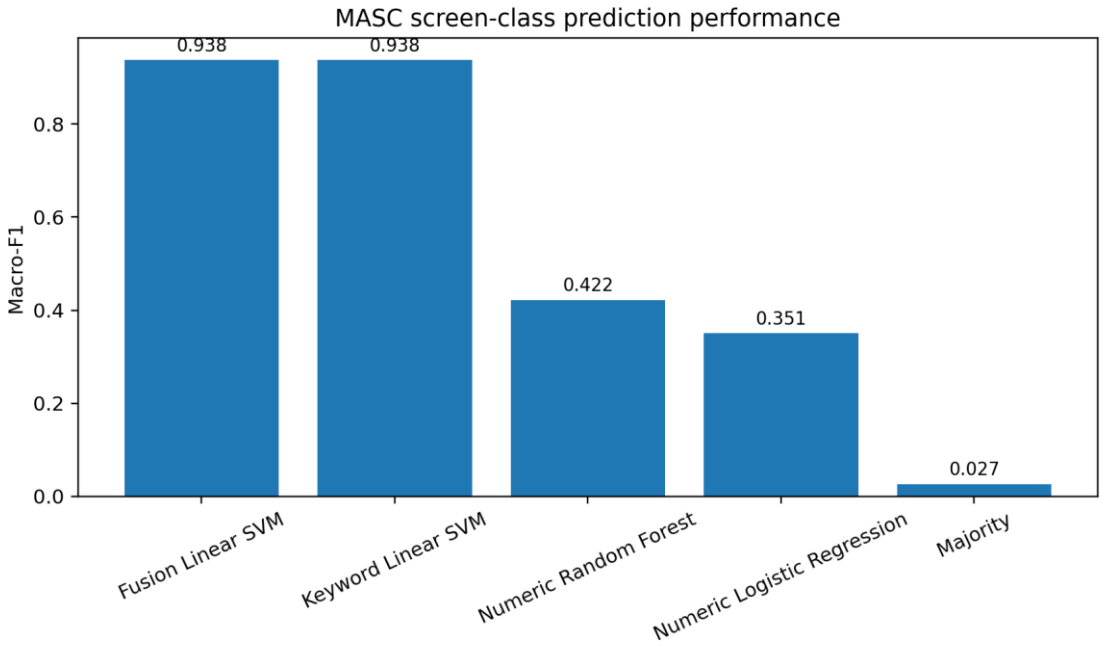

Figure 5. Macro-F1 Comparison for MASC Screen-Class Prediction Models

Figure 6 selected the number of complexity clusters. The silhouette score peaked at 0.345 for k = 2, compared with 0.323 for k = 3, 0.300 for k = 4, 0.307 for k = 5, and 0.321 for k = 6. Table 10 then shows the two resulting regimes. The low-complexity cluster contained 5489 screens (77.7% of MASC) and was dominated by Welcome, Login, and Home screens. Its mean complexity score was -0.759, with only 3.212 clickable elements and 4.760 general elements on average. The high-complexity cluster contained 1576 screens (22.3% of MASC), had a mean complexity score of 2.643, and was dominated by List, Menu, and Search screens. It averaged 10.388 clickable elements, 4.218 swipeable elements, and 15.861 general elements. Figure 7 visualizes the same pattern continuously: Menu had the highest median complexity, Welcome the lowest, and List occupied the upper part of the distribution. Figure 8 confirms that low-complexity screens are concentrated in Welcome and Login, whereas high-complexity screens are concentrated in List and Menu.

Table 10. Low- and High-Complexity Screen Clusters from MASC

complexity_cluster	complexity_cluster_name	screens	mean_complexity	total_clickable	total_textfields	total_swipeable	general_elements	nav_drawer_rate	dominant_classes
0	Low complexity	5489	-0.759	3.212	0.611	0.705	4.76	0.111	Welcome, Login, Home
1	High complexity	1576	2.643	10.388	0.414	4.218	15.861	0.244	List, Menu, Search

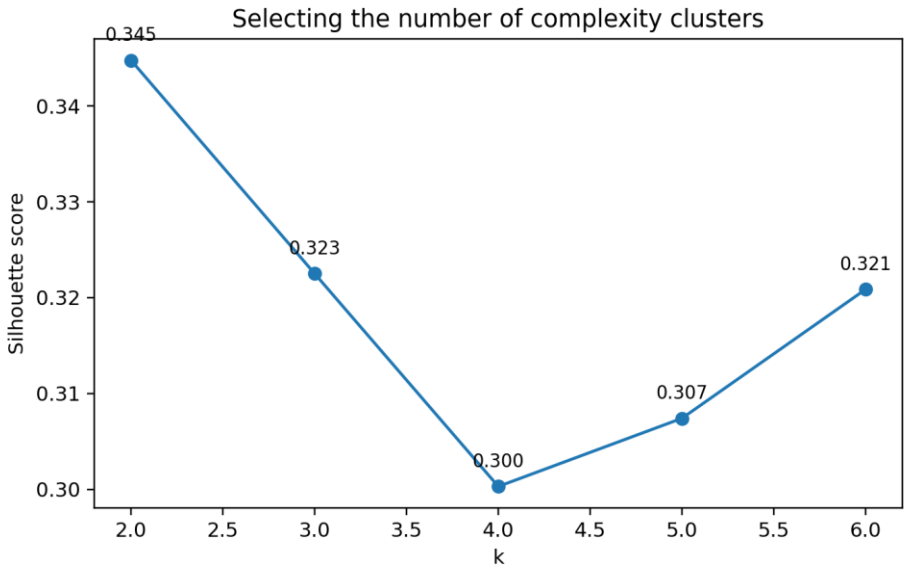

Figure 6. Sampled Silhouette Scores Used to Select the Number of Complexity Clusters

The final synthesis is reported in Table 11 and Figure 9. Because the complaint analysis produced distinct novice and advanced needs, and the structural analysis produced distinct low- and high-complexity screen regimes, the adaptive blueprint maps directly to two interface modes. The novice mode uses low-density screens, one-step cues, visible recovery options, and a single primary action path. The advanced mode uses denser list-and-panel layouts, comparative metrics, shortcuts, and expert toggles. Figure 9 translates this synthesis into a concrete volleyball-learning wireframe: the beginner side foregrounds one drill card at a time, while the advanced side packs session summary, metrics, filters, and custom drill composition into a compact layout.

Table 11. Adaptive UI synthesis for novice and advanced volleyball learners.

Learner state	Evidence trigger	Information density	Feedback granularity	Interaction complexity	Example UI copy
Novice	Low complexity cluster + novice-coded complaints that over-indexed on login/ recovery and guidance breakdowns	Low	One-step cues, exemplar clips, and immediate success confirmation af ter each drill	Single primary action per screen, capped options, persistent back path, and visible recovery links	Beginner mode is on. Follow one cue at a time and we will adapt the next drill automatically.
Advanced	High complexity cluster + advanced-coded complaints that emphasized control depth and monetization friction	High	Compact analytics, comparative feedback, and drill-level diagnostics with expandable rationale	Dense dashboards, shortcuts, custom drill composition, and detailed settings hidden behind expert toggles	Advanced controls are enabled. Tune metrics, compare sessions, and create custom practice blocks.

DISCUSSION

The results establish that adaptive UI for a volleyball learning app should respond to more than performance level in a narrow pedagogical sense. The review analysis showed that learner level is intertwined with very different tolerance thresholds for friction. Novice-coded users did not only need easier content; they also needed clearer recovery, visible support, and reassurance that progress would not disappear. Advanced-coded users, by contrast, concentrated more heavily on control quality and monetization friction. In design terms, this means that adaptation should change the surrounding interface contract as well as the lesson itself. Beginner adaptation must simplify, reassure, and recover. Advanced adaptation must condense, accelerate, and remove interruptions.

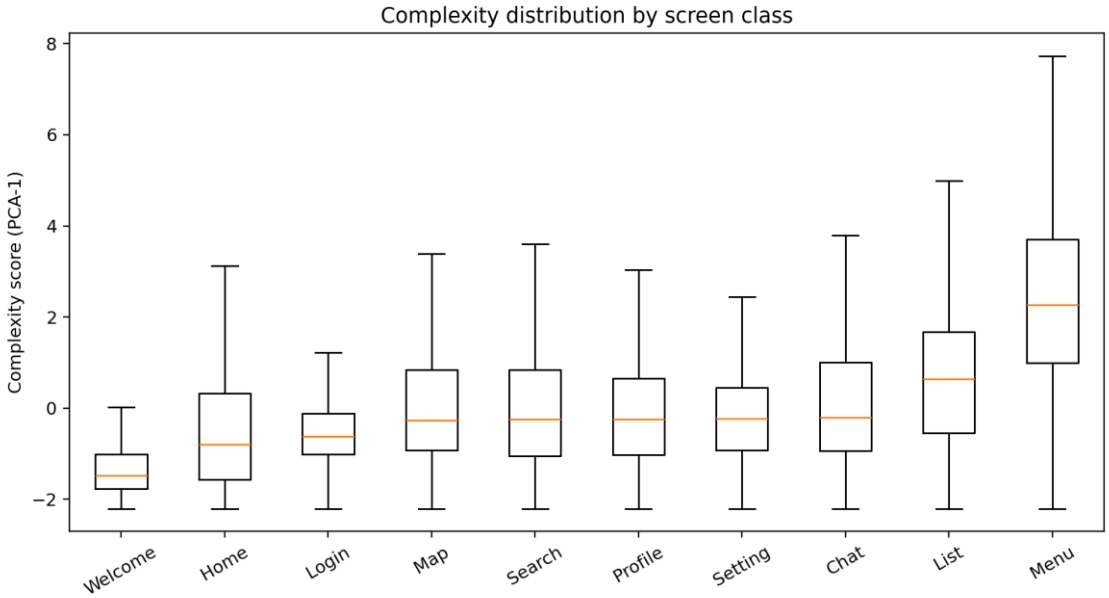

Figure 7. Distribution of the Continuous Complexity Score Across Screen Classes

This two-mode interpretation also fits the screen analytics. The low-complexity cluster was dominated by Welcome, Login, and Home screens, which are structurally suited to guided onboarding and narrow choice sets. The high-complexity cluster was dominated by List, Menu, and Search screens, which support broader exploration, rapid selection, and denser control surfaces. For a volleyball learning app, the implication is direct: early sessions should anchor learners in a home-like guided route with one focal cue, one demonstration, and one obvious next step. Later sessions should shift toward list and menu structures that let learners search drills, compare sessions, and assemble custom practice blocks.

The screen-classification comparison adds a useful nuance. Keyword semantics alone almost matched the fusion model, whereas numeric features alone were much weaker for exact

screen-type recognition. This means that screen meaning is encoded heavily in textual and symbolic cues such as the screen's keywords, labels, and obvious intent. However, the numeric features were still indispensable for the complexity analysis. In practice, a volleyball app therefore needs two concurrent adaptive logics: a semantic logic that decides what kind of screen the learner needs next, and a structural logic that decides how dense and interactive that screen should be. Treating those two decisions as identical would miss an important part of the design problem.

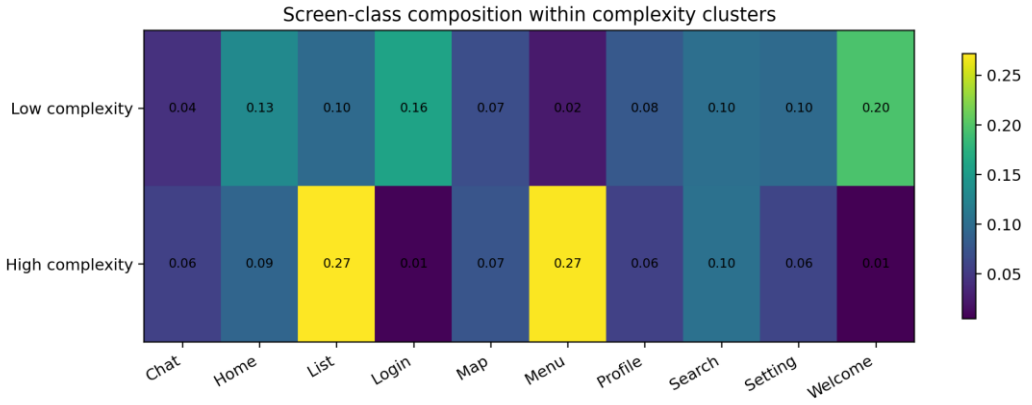

Figure 8. Screen-Class Composition within the Low- and High-Complexity Clusters

The complaint themes also show why explanation UI matters. If the system silently hides advanced controls from a beginner, that reduction can look like missing functionality. If the system suddenly exposes dense analytics to an advanced learner, the change can feel arbitrary unless the rationale is clear. Prior work on intelligibility and explanatory debugging argues that users accept adaptation more readily when they are given a brief reason and a visible path to override or recover (Lim & Dey, 2010; Kulesza et al., 2015; Abdul et al., 2018). That insight is reflected in the blueprint here. The novice microcopy explicitly states that beginner mode is on and that the next drill will adapt automatically. The advanced microcopy explicitly states that expert controls are enabled. These messages are small, but they help keep the adaptation legible.

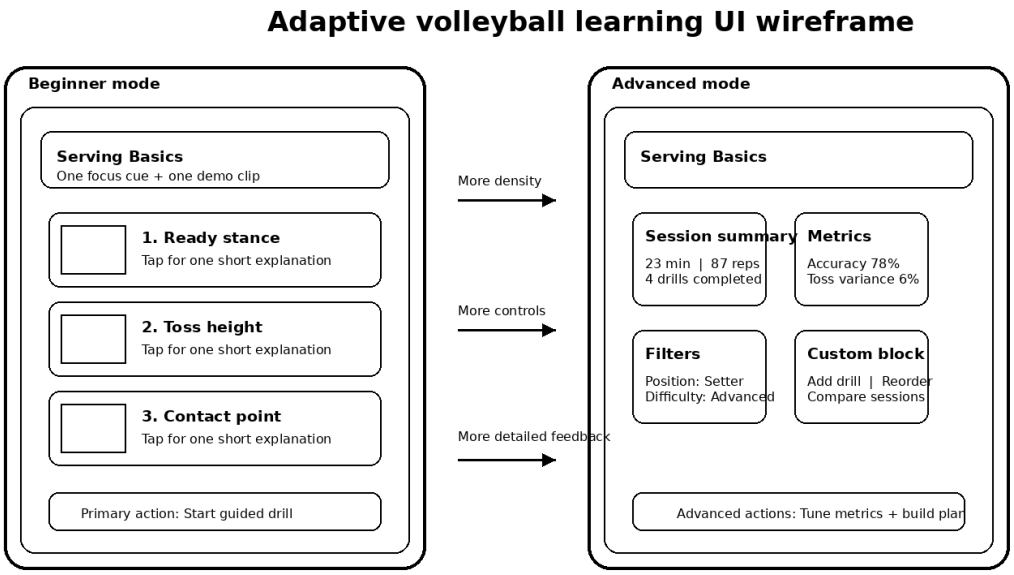

Figure 9. Two-Mode Adaptive Volleyball Learning Interface Derived from the Empirical Findings

From a practical design perspective, the study supports a staged interaction model for volleyball learning. In beginner mode, the app should display one technical cue at a time, use short clips rather than dashboards, and keep session recovery visible on the same screen. In advanced mode, the app should compress instructional scaffolding and expand analytic scaffolding: comparative metrics, filterable drill libraries, and configurable practice sequences. The microcopy revisions show that even short interface text should be level-sensitive. Recovery language belongs in novice surfaces because novice-coded complaints centered on support and account recovery. Dense option labels and delayed premium surfaces belong in advanced contexts because advanced-coded complaints centered on control quality and interruption.

More broadly, the paper demonstrates a replicable way to move from heterogeneous public datasets to a concrete adaptive design artifact. Review mining alone would reveal what users disliked but not what structural UI regimes are available. Screen analytics alone would reveal what low- and high-complexity screens look like but not which learner concerns should trigger them. Combining the two produced a defensible bridge from empirical evidence to design specification. That bridge is the main HCI contribution of the paper.

The study has four clear limitations. First, the Google Play subset was learning-and-training adjacent rather than volleyball-specific. The filtering rule combined category-code selection with a learning lexicon because the public file did not expose human-readable categories for every row. That procedure was reproducible and broad enough for empirical comparison, but it still treated adjacent educational and training apps as a proxy for a future volleyball app. Second,

sentiment labels were derived from star ratings rather than human sentence-level annotation. The confusion matrix showed that the neutral class remained difficult, so the sentiment benchmark should be interpreted as a scalable operationalization rather than a gold-standard affect model.

Third, the MASC analysis described structural screen complexity, not actual learner performance on those screens. A screen in the high-complexity cluster is structurally denser; it is not automatically cognitively harmful in every context. Fourth, the paper translated measured review and screen evidence into a volleyball UI blueprint, but it did not run a deployment study inside a live volleyball app. The blueprint is therefore an empirically grounded design specification rather than a completed field validation. A next-step study should combine live telemetry, learning outcomes, and possibly attention data from UEyes-like setups to validate whether the novice and advanced modes alter gaze allocation, task completion, and skill retention in the expected direction.

CONCLUSION

This paper developed a reproducible empirical basis for adaptive UI design in a volleyball learning app. A learning-and-training subset of 7,328 Google Play reviews was mined for sentiment and complaint themes, and a structured mobile screen dataset of 7,065 screens was used to benchmark screen recognition and structural complexity. The best review model was a Linear SVM with 0.548 macro-F1, the best topic model was LDA with four topics, the strongest screen classifier was a fusion Linear SVM with 0.938 macro-F1, and the structural analysis identified stable low- and high-complexity screen regimes.

The findings were coherent across data sources. Novice-coded complaints concentrated on support, billing, and account recovery, which supports a guided UI with visible recovery and low information density. Advanced-coded complaints concentrated on control quality and monetization friction, which supports a denser expert mode with more precise controls and fewer interruptions. The resulting blueprint therefore adapts not only content but also density, feedback granularity, interaction complexity, and explanatory microcopy. For researchers, the paper shows how review mining and UI analytics can be integrated into one HCI-oriented design pipeline. For practitioners, it provides a concrete, evidence-based starting point for building a volleyball learning interface that behaves differently for beginners and advanced learners.

REFERENCES

Abdul, A., Vermeulen, J., Wang, D., Lim, B. Y., & Kankanhalli, M. (2018). Trends and Trajectories for Explainable, Accountable and Intelligible Systems: An HCI Research Agenda. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1-18. https://doi.org/10.1145/3173574.3174156

Brusilovsky, P., & Millán, E. (2007). User Models for Adaptive Hypermedia and Adaptive Educational Systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The Adaptive Web*, 3-53. https://doi.org/10.1007/978-3-540-72079-9_1

Chen, N., Lin, J., Hoi, S. C. H., Xiao, X., & Zhang, B. (2014). AR-Miner: Mining Informative Reviews for Developers from Mobile App Marketplace. *Proceedings of the 36th International Conference on Software Engineering*, 767-778. https://doi.org/10.1145/2568225.2568243

Corbett, A. T., & Anderson, J. R. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253-278. https://doi.org/10.1007/bf01099821

Deka, B., Huang, Z., Franzen, C., Hibschman, J., Afergan, D., Li, Y., Nichols, J., & Kumar, R. (2017). RICO: A Mobile App Dataset for Building Data-Driven Design Applications. *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 845-854. https://doi.org/10.1145/3126594.3126651

Dinata, A. R., Hardiansyah, F., & Sari, J. F. (2025). Designing Emotionally Adaptive Interfaces: Affective UX Model for Enhancing Engagement in Gamified Learning Apps. *International Journal of Graphic Design*, *3*(2), 247–262. https://doi.org/10.51903/ijgd.v3i2.3098

Findlater, L., & McGrenere, J. (2004). A Comparison of Static, Adaptive, and Adaptable Menus. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 89-96. https://doi.org/10.1145/985692.985704

Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., & Sadeh, N. (2013). Why People Hate Your App: Making Sense of User Feedback in a Mobile App Store. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1276-1284. https://doi.org/10.1145/2487575.2488205

Gajos, K. Z., Czerwinski, M., Tan, D. S., & Weld, D. S. (2006). Exploring the Design Space for Adaptive Graphical User Interfaces. *Proceedings of the Working Conference on Advanced Visual Interfaces*, 201-208. https://doi.org/10.1145/1133265.1133306

Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M., & Weld, D. S. (2008). Predictability and Accuracy in Adaptive User Interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1271-1274. https://doi.org/10.1145/1357054.1357252

Hattie, J., & Timperley, H. (2007). The Power of Feedback. *Review of Educational Research*, *77*(1), 81-112. https://doi.org/10.3102/003465430298487

Jameson, A. (2003). Adaptive Interfaces and Agents. In J. A. Jacko & A. Sears (Eds.), *The Human-Computer Interaction Handbook*, 305-330. https://doi.org/10.1201/9781410608352.ch15

Kuhn, J., Chen, Y., & Chan, E. (2024). AI-Driven Mobile UI Pattern Recognition and Design Topic Mining on RICO: Semantic Clustering and Screenshot-Based Topic Classification. *Journal of Advanced Computing Systems*, *4*(5), 67-83. https://doi.org/10.69987/jacs.2024.40506

Jiang, Z., Wu, H., Shi, C., Zhao, Y., White, R. W., & Bendersky, M. (2024). UEYES: An Eye-Tracking Dataset Across User Interface Types. *arXiv*. https://arxiv.org/abs/2402.05202

Kim, N. W., Bylinskii, Z., Borkin, M. A., Isola, P., Sunkavalli, K., Oliva, A., & Pfister, H. (2017). BubbleView: An Interface for Crowdsourcing Image Importance Maps and Tracking Visual

Attention. *ACM Transactions on Computer-Human Interaction*, *24*(5), 1-36. https://doi.org/10.1145/3131603

Koedinger, K. R., D'Mello, S. K., McLaughlin, E. A., Pardos, Z. A., & Rosé, C. P. (2015). Data Mining and Education. *Wiley Interdisciplinary Reviews: Cognitive Science*, *6*(4), 333-353. https://doi.org/10.1002/wcs.1350

Kulesza, T., Burnett, M., Wong, W.-K., & Stumpf, S. (2015). Principles of Explanatory Debugging to Personalize Interactive Machine Learning. *Proceedings of the 20th International Conference on Intelligent User Interfaces*, 126-137. https://doi.org/10.1145/2678025.2701399

Lim, B. Y., & Dey, A. K. (2010). Toolkit to Support Intelligibility in Context-Aware Applications. *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, 13-22. https://doi.org/10.1145/1864349.1864359

Maalej, W., Nayebi, M., Johann, T., & Ruhe, G. (2016). Toward Data-Driven Requirements Engineering: Extracting Actionable Intelligence from Mobile App Reviews. *Proceedings of the 2016 IEEE 24th International Requirements Engineering Conference*, 227-236. https://doi.org/10.1109/re.2016.33

Marrero, W. M. (2019). *Google Play Store Data* [Data set]. Zenodo. https://doi.org/10.5281/zenodo.2839188

Martin, W., Sarro, F., Jia, Y., Zhang, Y., & Harman, M. (2017). A Survey of App Store Analysis for Software Engineering. *IEEE Transactions on Software Engineering*, *43*(9), 817-847. https://doi.org/10.1109/tse.2016.2630689

Mayer, R. E. (2009). *Multimedia Learning* (2nd ed.). Cambridge University Press. https://doi.org/10.1017/cbo9780511811678

Mendez, L., & Okafor, S. (2026). Adaptive Graphic Interaction Model: A Mixed-Method Framework for Future Factory Design. *International Journal of Graphic Design*, *4*(1), 1–16. https://doi.org/10.51903/ijgd.v4i1.3194

Pagano, D., & Maalej, W. (2013). User Feedback in the AppStore: An Empirical Study. *Proceedings of the 21st IEEE International Requirements Engineering Conference*, 125-134. https://doi.org/10.1109/re.2013.6636712

Petrova, S., & Watanabe, K. (2025). User-Centered Mobile Navigation: Evaluating Local Usability for Improved UX. *Journal of Technology Informatics and Engineering*, *4*(3), 478–492. https://doi.org/10.51903/jtie.v4i3.457

Shute, V. J. (2008). Focus on Formative Feedback. *Review of Educational Research*, *78*(1), 153-189. https://doi.org/10.3102/0034654307313795

Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science*, *12*(2), 257-285. https://doi.org/10.1207/s15516709cog1202_4

Wulf, G., & Lewthwaite, R. (2016). Optimizing Performance Through Intrinsic Motivation and Attention for Learning: The OPTIMAL Theory of Motor Learning. *Psychonomic Bulletin & Review*, *23*, 1382-1414. https://doi.org/10.3758/s13423-015-0999-9

Chen, Y., & Chan, E. (2023). Multimodal UI Representation Learning: Ablation of Screenshot, Wireframe, and View-Hierarchy Proxies on an Uploaded 168-Screen Dataset. *Journal of Advanced Computing Systems*, *3*(1), 1-15. https://doi.org/10.69987/jacs.2023.30101

Zaki, A., & Abdallah, M. (2023). MASC: A Dataset for the Development and Classification of Mobile Applications Screens. *Research Square*. https://doi.org/10.21203/rs.3.rs-3786876/v1