Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/24466
DOI URL: http://dx.doi.org/10.21474/IJAR01/24466

ISSN (O): 2320-5407
ISSN (P): 3107-4928

RESEARCH ARTICLE

SCALABLE MOBILE APPLICATION DESIGN SYSTEMS AND REUSABLE USER INTERFACE COMPONENTS FOR SAUDI ARABIA'S DIGITAL BANKING TRANSFORMATION

Ashique Ibrahim Ukkadan

Manuscript Info

Manuscript History
Received: 14 August 2026
Final Accepted: 16 September 2026
Published: October 2026

Abstract

Mobile banking in Saudi Arabia is increasingly the primary channel through which customers authenticate, transfer money, pay their bills and manage their financial relationships. The pace of feature delivery can, however, lead to a proliferation of interface inconsistencies, accessibility issues and duplicated effort. This review considers whether a scalable design system and a suite of reusable UI component s can balance the need for efficient development with the particular requirements of the regulated financial services sector. A structured integrative synthesis is used to bring together the research on Saudi banking usability, digital adoption, accessibility, interface localisation, modular frontend engineering and cross-platform performance. The 30 publications identified via DOI (2020-2025) are interpreted through a clearly defined set of architectural and human-centred criteria rather than being aggregated into any form of unsupported effect estimate. The synthesis makes a clear distinction between visual reuse and behavioural/transactional reuse and proposes a layered design system model comprising governed tokens, accessible primitives, financial workflow patterns and institution-specific compositions. The findings suggest that standardisation is most likely to deliver benefit when the shared components encode accessible states, Arabic directionality, transaction disclosure, error recovery and security boundaries, with the potential for audited variation. The reviewed literature provides little in the way of evidence on the downstream impact of component reuse on release quality, customer outcomes and inclusion. A comparative evaluation agenda is therefore proposed around traceable artefacts, task-based testing, operational telemetry and controlled rollouts. The review delivers a defensible design architecture and a set of testable propositions for banking teams and researchers, without in any way claiming that the proposed model has been experimentally validated.

"© 2026 by the Author(s). Published by IJAR under CC BY 4.0. Unrestricted use allowed with credit to the author."

Introduction:-

Saudi Arabia's banking transformation is not simply a shift from branch to screen. It's a transformation in the way financial obligations are explained, how counterparties are verified and how institutions make high value

Corresponding Author:- Ashique Ibrahim Ukkadan

transactions understandable on a small screen. The mobile apps now incorporate everything from the most basic account enquiries through to authentication, card control, payment initiation, beneficiary management and service support. Each of those additions has an impact on navigation depth, disclosure and the potential for user error. There is evidence from the Saudi banking apps of ongoing complaints about app functionality, the constant need for updates and the lack of adequate customer support [2]. That in itself is a warning against the assumption that more digital features necessarily equates to a more mature experience.

Accessibility is a particularly important test. Alayed's assessment of Arabic banking apps found significant conformance issues with respect to the core tasks that blind users would be expected to carry out, even when there were differences between the banks involved [1]. A similar analysis of Saudi fintech websites found errors which would effectively exclude disabled users [3]. This is about actual access to service rather than about preference. If a user can't identify the payee, can't see the amount or can't understand that a transfer has failed then the interface has failed at a point where there could be financial and personal ramifications. A good design system must therefore get right the semantics of interaction and the mechanisms for recovery, not just have consistent icons and colours.

A design system is viewed as a managed collection of reusable interface decisions, documented patterns, implementation components and verification rules. It is not the same as a static style guide which will describe the look and feel but won't necessarily contain the actual things you can use to build. It's also not the same as a generic component library which may provide you with buttons and cards but won't necessarily govern how those components are used in the context of a sensitive banking transaction. Research into the construction of design systems has looked at the interplay between reusable units, collaborative working and ongoing maintenance [4]. The work on adaptive interfaces has also highlighted the importance of separating the underlying component structure from the visual and organisational variants that can be configured [5]. For a financial organisation this is very much at the heart of the issue of shared infrastructure versus product ownership.

Modularity doesn't necessarily lead to scalability. Micro-frontend reviews have reported increased team autonomy but also greater integration complexity, duplicated assets and a larger monitoring burden [6,7]. Similar work on mobile application frameworks has also found that there is no one-size-fits-all solution that can be applied irrespective of the requirements [8]. This suggests that the design system problem is really a coordination problem across design, engineering, accessibility, security and product governance. The current review looks at how that coordination can enable Saudi digital banking without at the same time sacrificing clarity, platform performance or customer autonomy.

The contribution is explicitly architectural and evaluative. We're not claiming that any Saudi bank at present is using the pattern in question. Instead we are extracting the transferable mechanisms from the published evidence, separating those mechanisms from the context-specific assumptions and specifying the observable outcomes which would be the subject of further testing. We are also not in any way suggesting that a mobile banking adoption model can be substituted for evidence of the effectiveness of the individual components. Continuance research has established that perceived quality and trust are important to the customer [9,10] but that in itself does not prove that any particular design system implementation will lead to higher retention. The separation of the evidence from the inference is the key point.

Aim, objectives and research questions:-

The purpose of this review is to establish an evidence-based architecture and evaluation framework for scalable, reusable mobile banking interfaces that are appropriate to the linguistic, accessibility, security and service delivery context of Saudi Arabia. The analysis is structured around five objectives: to identify the common usability and inclusion issues in digital banking; to categorise the design system assets according to the appropriate level of reuse; to assess the architecture and platform trade-offs in terms of maintainability and performance; to define the governance mechanisms that will allow safe variation across the financial journey; and to produce a set of measurable propositions for future implementation studies.

The four research questions are: RQ1 - which of the documented banking interface deficiencies should drive component requirements? RQ2 - in what instances do shared design tokens, primitives, patterns and workflow assemblies provide tangible reusability and in what instances should product specificity be explicitly defined? RQ3 - how do the requirements of Arabic localisation, assistive technology, authentication and performance impact the architecture? RQ4 - what empirical measures could be used to determine whether adoption leads to improved release

quality and improved customer outcomes? Taken together these questions take the discussion beyond generic advice on intuitive applications to a testable account of reusable banking interaction design.

Review Methodology:-

We went with a structured integrative review as the research problem covers empirical user studies, technical comparison work and software architecture literature which is simply not something that can be meaningfully brought together in a single quantitative effect. The unit of synthesis was a design relevant claim backed by a publication that could be traced, not an individual application or respondent. The evidence base we have curated is precisely 30 journal articles and peer-reviewed conference papers from the period 2020 to 2025, all with stable DOIs. The review was conducted using topic led discovery and backward reference checking across the relevant computing, information systems and HCI literature. This is an interpretive synthesis rather than a statistically exhaustive census of all the studies that could possibly have been included.

Search expressions combined the terms: mobile banking, Saudi Arabia, user experience, accessibility, blind users, Arabic localisation, design systems, reusable components, frontend architecture, micro-frontends, cross platform development, security and continuance intention. The search logic was to combine banking domain terms with design/quality terms and engineering searches provided the relevant implementation evidence. Publisher pages, indexing records and DOI records were all checked to establish the source. Articles were kept if they reported on relevant banking behaviour, interface barriers, reuse architecture, localisation issues or mobile implementation trade-offs. General commentary without a proper scholarly publication record, undated material and articles outside of the specified timeframe were all excluded.

For each source that was retained the review extracted the publication context, study design, setting, observed result, limitation and potential implication for component requirements. The evidence was then grouped into six analytical domains: financial journey usability; accessibility and inclusive interaction; localisation; component architecture and reuse; cross platform behaviour; and governance, trust and evaluation. Where the evidence came from outside of Saudi Arabia or from a non-banking application it was flagged as transferable reasoning rather than as direct Saudi evidence. Cross study comparison focused on areas of agreement, areas of incompatible assumptions and the boundaries of inference. Contradictory findings on technical performance were kept as such rather than being averaged into a false ranking.

Quality appraisal separated out direct user observation from opinion-based models, small framework studies and secondary synthesis. The appraisal looked at task representativeness, population coverage, whether defects were manually checked, whether the performance tests used the same workload as the actual task and whether the causal claims went beyond what the study design actually allowed. The approach has three limitations: the search is not demonstrably complete; the methodological diversity means that effect sizes cannot be synthesised; and design-system outcomes are typically inferred from studies of the adjacent architecture rather than being measured directly in the Saudi banking context. These are all part of the interpretation, not omissions which we would be expected to hide. Table 1 shows the evidence to design translation and Figure 1 shows the proposed system boundary.

Table 1. Evidence-to-design translation for reusable banking interfaces

Evidence domain	Observed challenge or finding	Design-system implication	Evidence
Saudi banking journeys	Update-related instability and functional complaints	Reusable status, recovery and transaction-state patterns	[2,17]
Accessibility	Barriers within real financial tasks for blind users	Semantic controls, tested focus order and nonvisual feedback	[1,3,21]
Arabic localisation	Directionality, terminology and numeral presentation	Bidirectional layout rules and language-aware components	[16]
Design-system organisation	Reuse requires coordinated maintenance and adaptation	Versioned tokens, shared ownership and component contracts	[4,5]
Modular delivery	Independence may increase integration and duplication costs	Compatibility rules and deliberate domain boundaries	[6,7,29]
Mobile runtime	Cross-platform benchmarks vary by workload	Device-specific profiling and platform adapters	[8,20,22–24]
Continued use and trust	Experience, support, security and satisfaction relate to adoption	Evaluate complete journeys, not only visual consistency	[9–13,27]

Banking-specific requirements emerging from the literature:-

4.1 Task integrity and recovery. The quality of the banking interface has to be judged in terms of the complete financial journey rather than on individual screens. Saudi app reviews highlight the issues that are uncovered by updates and by failures which in fact block the delivery of a service that you would otherwise be able to use [2]. Topic modelling of international banking feedback also shows that the ratings are correlated with security, ease of use and customer service [17]. A reusable transfer pattern therefore needs to define the sequence from selection and validation through to disclosure, explicit confirmation, execution, receipt and recovery. Reusing a confirmation button without the corresponding state transitions would standardise the look and feel but would leave the higher risk interaction in a state of fragmentation. A good design system would make the loading state, the interrupted state, the duplicate submission state and the incomplete processing state just as reusable as the successful state.

Continuance studies explain why these factors are important without necessarily establishing component causality. The Saudi adoption study found perceived usefulness and perceived ease of use to be relevant to adoption of digital banking [11]. Work at the regional and international level has also found that trust, satisfaction and facilitating conditions are all associated with continued use of mobile banking [9,10,12,13,15]. The architectural inference is that the provision of predictable task outcomes will reduce the amount of unnecessary cognitive and operational uncertainty. That said, claims of improvement will require actual completion rates and customer evidence post-deployment. They cannot be inferred from intentions reported in a cross-sectional survey.

4.2 Accessibility as a component contract. Alayed's task based inspection of Saudi mobile banking apps shows that accessibility failures occur within the actual service flows, specifically for the visually impaired Arabic speaker [1]. The assessment of Saudi digital services and of international banking apps further underlines the real world importance of consistent semantics, readable content and assistive technology compatibility [3,19,21]. The point is not to treat accessibility as an afterthought. The shared primitives must be accessible by default with accessible labels, logical focus order, scalable type, status announcements, appropriate touch targets and non-visual error identification. A design system release must fail its verification if these contracts are broken irrespective of the visual quality.

Accessibility should also be considered in the light of the changing state of a transaction. The message may dynamically change an amount, may require additional verification or may even remove a beneficiary. Such changes must be announced in a way that is not completely silent and not in a way that is excessively intrusive. Reviews of visual assistance apps have shown that positive app reviews are a poor proxy for actual assessment of task performance [14]. For a banking library the relevant acceptance scenarios would be large text, screen reader navigation, reduced motion, limited connectivity and one handed use. Component documentation should define the expected behaviour in each state so that accessibility doesn't rely on the interpretation of individual members of the team.

4.3 Arabic language and bidirectional interaction. Arabic localisation is much more than a matter of string replacement. The RTL (right to left) nature of the language has implications for reading order, icon orientation, the way numbers are displayed, truncation and the way in which things are spatially related. Some transaction data (e.g. account numbers, payment references) will have to maintain their internal order in order to be meaningful. A design token is not going to be able to deal with all of these issues. The components themselves will need direction aware layout rules, numerical isolation, tested bilingual labels and stable affordances as you switch languages. Financial terminology will need to be chosen on the basis of what the customer will understand rather than a literal translation. The aim is semantic equivalence between the language versions, not just screens that are mirror images of each other.

Saudi financial apps will also be serving customers from a range of linguistic backgrounds. The design system should be able to handle English and Arabic content of differing lengths without compromising the readability of amounts, currencies, payee names and confirmation text. The design review will need to check that the language changes don't break an incomplete task or change the meaning of a warning. Accessibility and localisation come together in that the spoken label, visual placement and direction of the numbers all have an impact on how the user interprets the payment. These requirements will be captured at the component acceptance level prior to them being discovered via service complaint.

4.4 Financial legibility and informed action. Another requirement is that customers should be able to understand what they are getting into prior to committing to an irreversible or time-consuming transaction. The interface should

be able to clearly differentiate between estimated and actual charges, between charges which are scheduled and those which are due immediately, and between the initiation of a transaction and its actual completion. If an account name is truncated the customer must still have a clear path to verify the beneficiary. If the network connection fails the screen should not in any way suggest that resubmission is safe without the customer checking the status of the transaction. These are not general banking business rules but are stable categories of interaction risk. Shared patterns can be used to provide a structure for information hierarchy and for explicit confirmation, leaving the service specific financial content to be provided by the appropriate authorised teams. A recent synthesis of banking service quality also treats responsiveness, reliability and contextual value as inter-related rather than as separate attributes [30].

Reuse architecture, scalability and performance:-

5.1 A hierarchy of reusable decisions. The design system literature we've reviewed presents reusable components as part of a maintained organisational workflow rather than as discrete graphic assets [4,5]. On that basis we propose four layers. Foundation tokens cover the core semantics of typography, spacing, colour, elevation, motion and direction. Accessible primitives are the actual controls (buttons, inputs, alerts, lists, modals etc). Banking patterns take those primitives and assemble them into validated beneficiary cards, amount fields, account selectors, transaction summaries and risk disclosures. Journey assemblies then stitch the patterns together with the institution's specific journey orchestration, APIs, eligibility rules and compliance controls. Only the first three layers are candidates for broad reuse; the fourth layer must have explicit ownership of the actual financial behaviour.

This hierarchy guards against two potential pitfalls. Too much standardisation can mask the actual differences in payment methods or bank policies behind a single 'universal' widget. Too much local variation can result in incompatible forms, ambiguous status language and duplicated accessibility issues. The design system therefore needs to provide a stable semantic interface alongside clearly defined extension points. A good example would be an amount entry component which could be configured for different currencies and decimal rules but which would not allow the removal of accessible labels or the removal of precision checks. A transaction summary pattern could accept service specific fields but would require the amount, recipient, charges, timing and final action to all remain discoverable. The result of such contracts is that reuse becomes a constraint on quality as much as it is a way of reducing effort.

5.2 Modular ownership without uncontrolled fragmentation. The micro-frontend research does point to the benefits of independent delivery, but also highlights the integration overhead, the risk of duplicate dependencies and the potential for inconsistent user experiences [6,7,29]. This is research into modular frontends rather than modular design systems so the relevance is purely in terms of architectural analogy rather than any direct evidence of benefit to the bank. A bank with multiple service teams can of course share design packages and still have domain ownership. It would not be appropriate to simply go and assemble the independently deployed visual bits if the end result is going to be a poor navigation experience, poor runtime performance or poor auditability. Shared versioning and shared compatibility testing is the better way of ensuring you don't end up with inconsistent behaviours.

A release process should incorporate semantic versioning of components, documented migration paths, change impact analysis and staged adoption across products. Breaking changes to error messaging or focus order in particular need to be reviewed very carefully as the impact of such changes is felt across many journeys. The component registry should expose usage, owners, dependencies, accessibility status, language coverage and OS version support. The reusable asset is not just the source code itself but also the design rationale, usage restrictions, interaction tests and the full review history. A component that looks the same but is implemented in a different way by each team is not genuine technical reuse.

5.3 Platform-neutral specification and platform-sensitive execution. Research into native vs cross-platform mobile development has failed to identify a single framework that is best in all circumstances [8,20,22–24,28]. Resource consumption is highly dependent on the complexity of the interaction, the amount of animation, the device itself and what exactly you are measuring. A scalable banking design system should therefore not be equated with a requirement for a single rendering technology. A platform-neutral semantic specification can define the component states and the business independent rules, with the implementation adapters then being responsible for preserving the native assistive technology behaviour and the performance sensitive controls. Regression benchmarks should include launch latency, memory pressure, rendering stability, battery consumption and screen reader output on a range of representative devices.

Security sensitive operations require particular care in terms of boundary definition. Authentication prompts, device trust, session expiry and transaction authorisation should be based on the established platform and banking services rather than be recreated through the use of ornamental UI elements. Work on biometric banking has shown that the choice of authentication is in fact influenced by device characteristics and privacy related behaviour [27]. The role of the design system is to provide clear and understandable entry, permission, fallback, cancellation and failure states around those services. It is not the role of the design system to go and implement or independently certify the security of the identity verification itself. That separation is essential for both flexibility of implementation and for clarity of responsibility.

5.4 Design debt and component retirement. Reusable libraries accrue debt as the products they're used in evolve. A styling token from a long ago version of a screen may still be in use, a component that was perfectly appropriate for a simple account enquiry might be used in a completely different context (a complex payment for instance). The library therefore needs a deprecation policy as much as a contribution policy. Components that are deprecated need to remain discoverable with clear guidance on what to use instead, who the affected product owners are, what the migration deadlines are and what the documented accessibility risks are. There should be a requirement for evidence that a genuine difference in customer or regulatory need exists before a new variant is created. In that way the design system can manage variation in a transparent manner rather than just allow it to grow in an invisible fashion. The principle is the same as modular engineering governance, just applied to the user facing consequences where even a technically compatible change can have an impact on understanding.

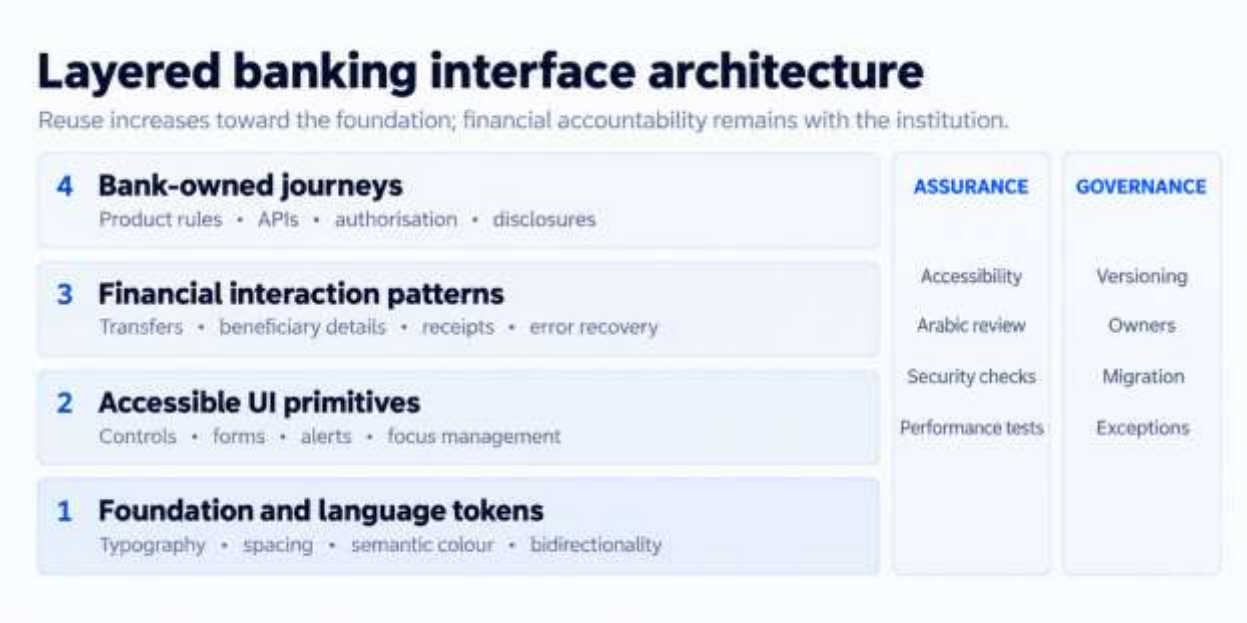


Figure 1. Proposed layered architecture separating shared interface contracts from bank-owned service responsibilities.

Critical synthesis and proposed conceptual framework:-

The three propositions that emerge across the evidence domains are: first, reuse leads to improved consistency only if the shared assets contain interaction rules rather than merely visual similarity; second, accessibility and localisation need to be represented in the component interfaces, the tests and the documentation as otherwise the application level of repair is very difficult to scale across multiple products; and third, delivery scalability is a joint property of the technical architecture and the governance. Individual teams can deliver very quickly but if you have shared dependencies and shared conventions you will still end up with integration failures. These are all inferential propositions. They are derived from a range of different bodies of evidence and theory but have not been jointly validated in the Saudi banking deployments.

Figure 1 shows the proposed architecture as a layered contract. There's a common semantic layer underpinning the accessible controls, the financial patterns and the bank owned journeys. There are two horizontal controls (assurance and governance) that apply to all layers. Assurance is concerned with security review, accessibility testing, Arabic

language testing and mobile performance testing. Governance is about design ownership, contribution rules, change approval and version compatibility. This separation of concerns reduces the risk that a shared component will in fact acquire business authority without anyone being aware of it. The payment workflow remains accountable to the bank's transaction services even if the interface is composed of reusable patterns.

The framework also makes it clear where reuse is not appropriate. Credit eligibility decisions, product specific fees, authentication policy, fraud signals and any legally significant disclosures cannot simply be made to be reusable on the basis of having similar screens. You can reuse the UI shell in those cases, but the data, the permissions and the decision logic will need to be in a contract and will need to be validated independently. On the other hand focus handling, empty state messaging, number formatting, interactive feedback and error summaries are all good candidates for common infrastructure as variation in those areas is generally of little value to the customer. Reuse should be about semantic stability and risk not about trying to maximise the % of code that you can share.

Figure 2 shows a closed governance loop from evidence intake through to component change and observed service outcome. Customer complaints and usability studies provide the patterns. The design and engineering teams then specify the reusable contracts. The implementation is then examined via automated and participatory verification. Staged releases produce the observable performance and task evidence. This evidence then feeds into the next review. Response studies have demonstrated that systematic engagement with user feedback can be a driver of product improvement [25]. Crucially the loop makes the difference between simply reporting an issue and having proof that an interface change has in fact resolved it. A change that is accepted should have a testable hypothesis and an identifiable owner.

The architecture as proposed can be viewed as a series of obligations. A primitive makes the promise that it is observable and operable; a financial pattern makes the promise that its states convey the meaning of an action; a journey makes the promise that the bank will deal with that action in the correct manner. Each layer may be dependent on the guarantees of the layers below it, but no lower layer can validate the claims of the layer above. This is of course vital when components are being shared across personal accounts, corporate payments and card products. You can then audit the reuse of a component without effectively hiding the differences in authorisation, settlement, cancellation rights and customer entitlements. Those differences also provide a practical basis for deciding when a shared interaction should be forked, parameterised or simply left as a local implementation.

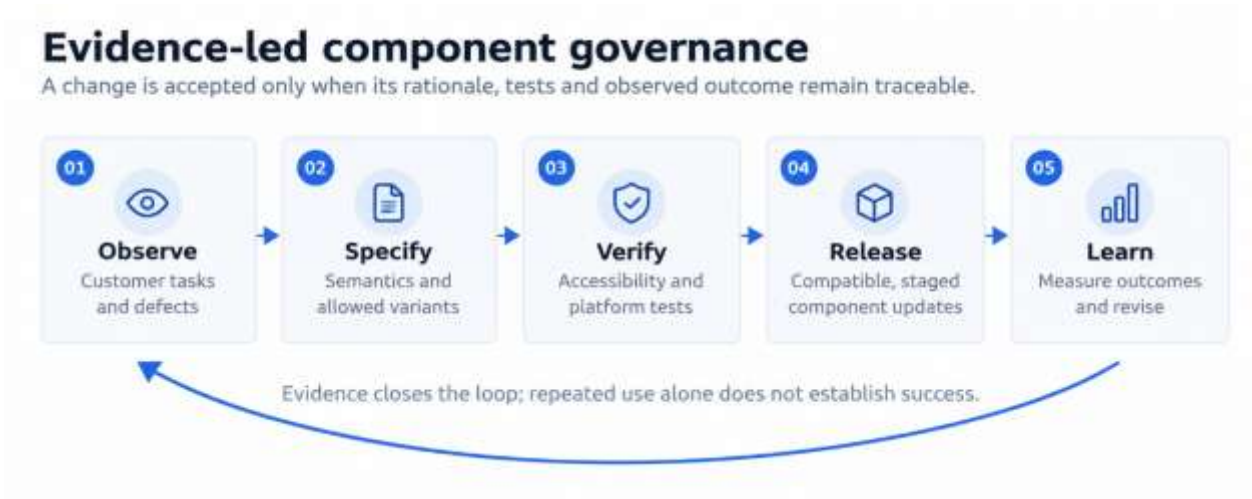


Figure 2. Proposed governance cycle for specifying, verifying, releasing and revising shared banking components.

Evaluation agenda and testable propositions:-

The biggest research gap is the lack of any longitudinal, component level evidence of a design system intervention leading to banking outcomes in the Saudi context. There is a lot of work out there on user complaints, accessibility, adoption models and framework comparisons but this is all at a very different level of analysis [1,2,11,18,26]. A good evaluation would be to take a set of baseline financial journeys and then compare those against journeys that have been implemented using a set of specified reusable assets. The selection of journeys should be representative of both common and high risk tasks, across a range of device classes, in both Arabic and English and with people who

have disabilities. A simple before and after would be very susceptible to the effect of other concurrent product changes and so matched journey comparisons or a phased roll-out would be preferable where the organisation is able to do that.

P1 is that governed reusable financial patterns will reduce cross-journey inconsistency and regression defects as compared to independently maintained interfaces. P2 is that embedding accessibility and bidirectional language contracts at the component level will improve task completion for customers using assistive technologies. P3 is that reuse will only reduce change lead time if versioning, documentation and integration ownership are all sufficiently mature. P4 is that runtime performance and customer trust will not necessarily be improved by reuse unless it has been tested independently. These are propositions to be measured rather than presented as observed results. Table 2 details the evaluation instruments, comparison units and potential confounders.

Evidence collection should be a mix of interface audits, moderated task observation, release engineering logs, telemetry and customer stories. Task success should be defined in terms of completion, correct completion and completion with assistance. Timing should be considered in conjunction with errors and effort as an extremely short path can also be the result of incomplete verification. Accessibility assessment needs to be done manually and with actual user groups, not just via automated scans. The delivery metrics should include defect escape, migration effort, component adoption and rollback frequency. The trust metrics are tricky to interpret as external events (marketing, service outages, security incidents etc) can all have an impact on customer perception of trust independently of the interface design.

Evaluation should also look at the denominator in each metric. A decreasing number of reported defects could be due to better software, less testing or users simply giving up on a difficult task and not reporting it. Component adoption rates can go up just because teams are putting their own bespoke interfaces into the shared containers. So the better indicators are those which are clearly defined in terms of operation. A reused component is an identical versioned implementation which adheres to the same tested semantic contract. A successful financial transaction is one in which the outcome of the transaction is verified. Researchers should report all missing observations, device exclusions, participant recruitment and any deviations from the protocol. Qualitative incidents which demonstrate a profound level of misunderstanding should not be obscured by an average level of satisfaction, especially where the group in question have few alternatives.

Table 2. Proposed evaluation protocol and interpretation safeguards

Hypothesis area	Primary indicator	Evidence source	Principal confounder
Consistency	Inconsistent states and navigation defects per journey	Expert inspection and issue tracker	Concurrent redesign
Inclusive task success	Correct completion without assistance	Observed tasks with assistive-technology users	Prior familiarity
Arabic parity	Directional and terminology errors per workflow	Bilingual task audit	Translation quality
Delivery efficiency	Cycle time for equivalent feature changes	Release records and code history	Team staffing
Release reliability	Post-release defect and rollback rates	Production incidents and QA logs	Backend outages
Runtime quality	Interaction latency, energy and memory	Device-level performance tests	Device generation
Customer assurance	Comprehension of transfer status and confirmation	Interviews and task questions	External trust events

Practical, organisational and policy implications:-

For product leaders the starting point is an inventory of the financial interactions that we repeat, rather than a complete visual refresh. The initial library should be built around the elements that we use most frequently and which carry clear risk: amount entry, beneficiary selection, transaction status, error resolution, permission explanations and disclosure patterns. For each asset you need to document user intent, state transitions, accessible alternatives, localisationbehaviour and the degree of restriction of variation. A staged migration is the way to go in order to be able to compare the outcomes without putting all of your journeys at risk at the same time. The

governance committees should be made up of designers, mobile engineers, quality people, accessibility reviewers and the relevant banking risk owners rather than just being a visual branding committee.

For engineering managers a design system should be treated as maintained product infrastructure (supported contribution process, release notes, compatibility guarantees, testing on a representative set of devices, ownership of deprecated features etc). Procurement of cross platform frameworks should be based on actual banking interaction workloads rather than on vendor popularity. For inclusion and regulatory functions the audit evidence should demonstrate that the accessibility rule exists AND that the core customer journeys are in fact usable in Arabic with assistive technology. A governed specification can enable repeatable assurance without in any way replacing the institution's own accountability for secure transactions.

A sensible governance model would have a central stewardship team responsible for component semantics and shared verification, with the financial product teams themselves being responsible for the content of their workflows and the integration of services. Exception requests would need to include the reason for the variation, the associated risk, the impact on accessibility states and a review date. That is better than having a central team who approve everything which would become a massive delivery bottleneck, or having completely uncontrolled local copies which would kill off standardisation. Managers should be looking at the queue time for design system contributions as well as the downstream reduction in bugs. If an exception is occurring in a number of products then that may be a sign of a missing pattern rather than a failure of compliance by the teams in question. In that situation the shared contract would need to be re-evaluated in the light of actual customer needs.

Limitations and Future Research:-

The review draws on a deliberately mixed body of evidence. The Saudi banking usability work provides some very useful local insights. The component architecture and platform work tends to come from other countries or from non-banking contexts. The move from that work to banking design governance is made in a very conceptual way rather than demonstrated through any actual experiments. A number of the sources are focused on customer intent or app store reviews which again don't necessarily reflect actual accessibility, task correctness or operational risk. The body of literature is also very heterogeneous in terms of study quality and platform age so any technical observations are not going to be transferable to newer OSs or devices. No pooled effect size, no market wide prevalence estimate and no bank specific compliance conclusion is claimed.

Further work should involve testing the component hierarchy across multiple implementations in Saudi banks, publish pre-registered evaluation criteria and report both the positive and negative results. Comparative studies could look at whether the available primitives actually improve journey completion (all else being equal in terms of interface familiarity), whether an Arabic layout that contracts reduces errors related to direction and whether a design system migration has any effect on defect escape rates. Longitudinal work could look at governance maturity, developer onboarding, version drift and the opportunity cost of excessive reuse. Inclusion work should involve users with visual, motor and cognitive access needs as well as those with limited digital literacy. The evidence would be far more useful if it were publicly available in the form of replication materials rather than just a series of unexamined success stories.

The framework also makes an assumption about organisational capacity for sustained multidisciplinary oversight. Smaller organisations may simply not have the specialist access to staff or the engineering capacity to support a parallel mobile implementation. Larger organisations may be beset by issues of fragmented ownership and deeply entrenched legacy components. Future work in this area should look at what the minimum governance capabilities need to be before reuse becomes worthwhile. There would be value in cross-institutional collaboration to test shared semantic requirements without the need to share any of the underlying transaction logic or customer data. An open research protocol should be developed which maintains the distinction between interface artefacts which can be safely made available to the external world and internal security controls which would be inappropriate to disclose. This would allow for independent assessment of inclusion and clarity without compromising the appropriate level of protection around the sensitive financial infrastructure.

Conclusion:-

Scalable mobile banking design is more than just the re-use of UI components. The evidence we've reviewed suggests that the Saudi financial journey is still rife with usability problems and accessibility exclusions. Broader

software engineering research also shows that modularity can both create and solve complexity. A legitimate design system therefore requires shared semantic foundations, accessible controls, validated financial patterns and a clear separation from institution specific transaction logic. Arabic directionality, customer understanding, error recovery, performance and security related presentation are all part of those shared contracts.

The paper's main contribution is an architecture that ties reuse decisions to verification and accountability, with testable assertions rather than just unsubstantiated claims of success. The approach enables banking teams to standardise the predictable interactions whilst still protecting the required service differences. Its value will ultimately be determined by the results we see in real customers, real devices and real release cycles. The next stage of evidence will need to tell us not just that components are being reused more often, but that they are in fact helping people to complete financial tasks correctly, on their own and with the confidence that they have.

References:-

1. Alayed, A. (2024). Are Saudi Arabian banks' mobile applications accessible for blind or partially sighted users? A customers' perspective and evaluation. Universal Access in the Information Society, 24, 585–606. https://doi.org/10.1007/s10209-023-01082-y
2. Alhejji, S., Albesher, A., Wahsheh, H., & Albarrak, A. (2022). Evaluating and comparing the usability of mobile banking applications in Saudi Arabia. Information, 13(12), 559. https://doi.org/10.3390/info13120559
3. Nour, R. (2022). An assessment of accessibility and usability of Saudi online FinTech services for people with disabilities. Computational and Mathematical Methods in Medicine, 2022, 8610844. https://doi.org/10.1155/2022/8610844
4. Jeong, J., & Pan, Y. (2024). Design system construction framework using Figma. Journal of Digital Contents Society, 25(3), 615–624. https://doi.org/10.9728/dcs.2024.25.3.615
5. Polishchuk, K., &Brezhniev, E. (2024). The use of artificial intelligence in adapting process of UI design system for end customer requirements. Radioelectronic and Computer Systems, 2024(4). https://doi.org/10.32620/reks.2024.4.02
6. Santos, L. F. C., Silva, M. V. S., Santos, S. R. R., Rocha, F. G., & Barbosa da Silva, E. (2024). Microfront-End: Systematic mapping. Proceedings of the 20th International Conference on Web Information Systems and Technologies, 119–130. https://doi.org/10.5220/0013015400003825
7. Peltonen, S., Mezzalira, L., & Taibi, D. (2021). Motivations, benefits, and issues for adopting micro-frontends: A multivocal literature review. Information and Software Technology, 136, 106571. https://doi.org/10.1016/j.infsof.2021.106571
8. Souha, A., Benaddi, L., Ouaddi, C., &Jakimi, A. (2024). Comparative analysis of mobile application frameworks: A developer's guide for choosing the right tool. Procedia Computer Science, 236, 597–604. https://doi.org/10.1016/j.procs.2024.05.071
9. Saibaba, S. (2024). Examining the determinants of mobile banking app continuance intention in India: An extension of the IS success model. Journal of Internet Commerce, 23(1), 50–89. https://doi.org/10.1080/15332861.2023.2236428
10. Hidayat-ur-Rehman, I., Ahmad, A., Khan, M. N., & Mokhtar, S. A. (2021). Investigating mobile banking continuance intention: A mixed-methods approach. Mobile Information Systems, 2021, 9994990. https://doi.org/10.1155/2021/9994990
11. Alnemer, H. A. (2022). Determinants of digital banking adoption in the Kingdom of Saudi Arabia: A technology acceptance model approach. Digital Business, 2(2), 100037. https://doi.org/10.1016/j.digbus.2022.100037
12. Rabaa'i, A. A., & Abu ALMaati, S. (2021). Exploring the determinants of users' continuance intention to use mobile banking services in Kuwait: Extending the expectation-confirmation model. Asia Pacific Journal of Information Systems, 31(2), 141–184. https://doi.org/10.14329/apjis.2021.31.2.141
13. Nguyen, G.-D., & Dao, T.-H. T. (2024). Factors influencing continuance intention to use mobile banking: An extended expectation-confirmation model with moderating role of trust. Humanities and Social Sciences Communications, 11, 276. https://doi.org/10.1057/s41599-024-02778-z
14. Pundlik, S., Shivshanker, P., & Luo, G. (2023). Impact of apps as assistive devices for visually impaired persons. Annual Review of Vision Science, 9, 111–130. https://doi.org/10.1146/annurev-vision-111022-123837
15. Arıkan, E., &Ozecan, M. (2022). An integrative model on user satisfaction and continuance intention for using m-banking. Business and Economics Research Journal, 13(3). https://doi.org/10.20409/berj.2022.384
16. Anvari, S. S., & Woods, P. (2024). Localizing graphical user interfaces for right-to-left languages: A practical study. The Human Side of Service Engineering, 143, 231–234. https://doi.org/10.54941/ahfe1005102
17. Oh, Y. K., & Kim, J.-M. (2022). What improves customer satisfaction in mobile banking apps? An application of text mining analysis. Asia Marketing Journal, 23(4), Article 3. https://doi.org/10.53728/2765-6500.1581

18. Ali, W., Riaz, O., Mumtaz, S., Khan, A. R., Saba, T., & Bahaj, S. A. (2022). Mobile application usability evaluation: A study based on demography. IEEE Access, 10, 41512–41524. https://doi.org/10.1109/ACCESS.2022.3166893
19. Alsaeed, D., Alkhalifa, H., Alotaibi, H., Alshalan, R., Al-Mutlaq, N., Alshalan, S., Bintaleb, H. T., &Alsahow, A. M. (2020). Accessibility evaluation of Saudi e-government systems for teachers: A visually impaired user's perspective. Applied Sciences, 10(21), 7528. https://doi.org/10.3390/app10217528
20. Oliveira, W., Moraes, B., Castor, F., & Fernandes, J. P. (2023). Analyzing the resource usage overhead of mobile app development frameworks. Proceedings of the Evaluation and Assessment in Software Engineering Conference, 152–161. https://doi.org/10.1145/3593434.3593487
21. Borowska-Beszta, A., Smieszek, M., & Borowska-Beszta, B. (2025). Mobile banking apps in Poland and their accessibility for consumers with disabilities: A case study. Disability & Society, 40(3), 571–592. https://doi.org/10.1080/09687599.2023.2263631
22. Białkowski, D., &Smołka, J. (2022). Evaluation of Flutter framework time efficiency in context of user interface tasks. Journal of Computer Sciences Institute, 25. https://doi.org/10.35784/jcsi.3007
23. Markowski, M., &Smołka, J. (2023). A comparative analysis of the Flutter and React Native frameworks. Journal of Computer Sciences Institute, 29. https://doi.org/10.35784/jcsi.3794
24. Michałowski, M., &Skublewska-Paszkowska, M. (2022). Analysis of the performance of iOS applications developed using native and cross-platform technology. Journal of Computer Sciences Institute, 23. https://doi.org/10.35784/jcsi.2924
25. Srisopha, K., Link, D., & Boehm, B. (2021). How should developers respond to app reviews? Features predicting the success of developer responses. Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, 119–128. https://doi.org/10.1145/3463274.3463311
26. Alzahrani, A., Beloff, N., & White, M. (2024). Key factors influencing mobile banking adoption in Saudi Arabia. Proceedings of the 19th Conference on Computer Science and Intelligence Systems, 39, 531–536. https://doi.org/10.15439/2024F8633
27. Cho, I., Kwak, J. H., & Lee, B. G. (2025). Mobile banking usage through biometric authentication: Effects of smartphone attributes and privacy consent index. IEEE Access, 13, 202919–202937. https://doi.org/10.1109/ACCESS.2025.3631849
28. Uciński, M., &Dzieńkowski, M. (2022). A comparative analysis of performance of Flutter and Xamarin development frameworks. Journal of Computer Sciences Institute, 25. https://doi.org/10.35784/jcsi.3025
29. Kaushik, N., Kumar, H., & Raj, V. (2024). Micro frontend based performance improvement and prediction for microservices using machine learning. Journal of Grid Computing, 22, 44. https://doi.org/10.1007/s10723-024-09760-8
30. Palamidovska-Sterjadovska, N., Rasul, T., Lim, W. M., Ciunova-Shuleska, A., Ladeira, W. J., Santini, F. D. O., &Bogoevska-Gavrilova, I. (2025). Service quality in mobile banking. International Journal of Bank Marketing, 43(6), 1195–1230. https://doi.org/10.1108/IJBM-02-2024-0105