computers

MDPI

Article

A Security Assessment for Interactive Mobile Healthcare Systems: A Lifecycle Framework

Yousra Odeh

Software Engineering Department, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman 11733, Jordan; y.odeh@zuj.edu.jo or yousraodeh.odeh@gmail.com

Abstract

Mobile health systems have become a central component of the health sector, reshaping how healthcare is accessed, delivered, and experienced, especially in developing countries with limited infrastructure. Mobile health systems are interactive and enable continuous interaction among patients, health professionals, and mobile technology. While these systems improve access to care and patient engagement, they also raise important concerns about how security is handled in highly interactive socio-technological environments. The current state of security assessment for mobile health systems tends to be biased towards technical security controls without addressing the required user interactions during mobile health system development in relation to providing security measures and evidence. This study addresses this gap by proposing a Phase–Role Interaction Security Assessment framework for mobile Health, namely, the PRISM-mH framework. The framework considers three dimensions: interaction-oriented security development lifecycle phases, the roles of the stakeholders involved, and evidence supporting the security requirements. With its help, security can be assessed continuously while taking into consideration the phases and role interactions. This framework is aimed at enhancing the visibility of security during user interactions, while also offering a systematic reference for researchers and practitioners to assess and enhance their security through the mobile health system lifecycle. Considering that PRISM-mH is proposed to address user interaction, technical control, and governance—factors that have not been considered consistently in past mobile health security assessments—this framework opens the door to more comprehensive and evidence-based security assessment, with a goal of enhancing the security position in mobile health development. The author argues that such a framework can help to propose a structure for an agile-driven approach for future studies and provide scoring-based guidance for informing the level of security maturity in mobile health technologies.

Keywords: interactive mobile healthcare system; healthcare system security; security assessment model; user–system interaction; digital health; software engineering

Check for updates

Academic Editor: Makuochi Samuel Nkwo

Received: 11 August 2026
Revised: 5 October 2026
Accepted: 7 October 2026
Published: 9 October 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Over the past two decades, mobile health (m-health) applications focused on tracking health and wellness [1,2], managing chronic diseases [3,4], and monitoring and following up with patients through telemedicine [5,6] have seen significant global growth. In 2024, 43% of Americans used m-health applications [7]. About 70% of people in India, on the other hand, have used apps that track medicine, symptoms, and nutrition [7]. In Jordan, the latest statistics show a high adoption rate, with approximately 48% of Jordanians using m-health applications [8]. These applications often process sensitive health data governed

by rigorous regulations, such as the Health Insurance Portability and Accountability Act (HIPAA), the General Data Protection Regulation (GDPR), and other regional health data regulations and acts [9].

Cyberthreats have increased since health applications and backend infrastructures have become increasingly cloud-driven and networked [10–12]. The COVID-19 pandemic has witnessed high use of m-health applications. This has necessitated ongoing development to meet users' needs and changing health conditions. This has reflected a radical change in the user experience [13]. Also, this has led to risks becoming higher since any service interruptions or data breaches may result in catastrophic consequences [11,14,15]. The main cyberthreats include, but are not limited to, malware, data breaches, ransomware, smishing, and phishing [16,17]. Recent statistics show a growth in cyberthreats, which highlights the need to assess the security of m-health interactions continuously [18]. Around 5.4 million mobile cyberattacks occurred worldwide in December 2023, with a 147-percent increase from December 2022 [19]. This threat is also apparent from the detection of over 989,000 unique global phishing attacks in the fourth quarter of 2024, which witnessed an increase over the previous quarter [20].

Security assessment models (SAMs) have emerged as structured frameworks and essential tools for evaluating an organization's security capabilities, identifying current gaps and limitations, and assessing its readiness level to guide required incremental improvements [21,22]. Originating from capability assessment paradigms in software engineering [23,24], a SAM follows a systematic level of progress, moving from an initial or ad hoc level to an optimized level through a structured, layered roadmap. A well-defined SAM supports stakeholders' security decision-making for enhancing current security practices and aligning them with m-health objectives.

Since the COVID-19 pandemic, the digital transformation of m-health applications has reached critical milestones and introduced more cybersecurity challenges, leading technical and non-technical stakeholders to conduct security assessments using SAMs. Applications of SAMs in healthcare include evaluating security at different levels of healthcare organizations to implement a structured roadmap [22], assessing e-health services in the cloud environment and guiding an organization's security practices [25], assessing healthcare organizations through the integration of human factors, policies, and technical aspects [26], measuring security performance through predefined metrics to manage threats and associated resources [27], and enhancing security practices to improve the overall quality of healthcare services [28]. A review of current efforts in Jordan shows that SAMs have been applied to quality management in healthcare information systems [29,30]. Ramadan and Arafeh proposed a quantitative quality maturity assessment framework based on five quality levels of maturity and six quality drivers. They applied this framework to local hospitals in Jordan. The findings supported its ability to evaluate the current quality status and identify ongoing progress. In this limited body of literature, no work has focused on interactive assessment approaches that explicitly consider m-health development phases while distinguishing responsibilities among interacting roles and requiring tangible evidence to support assessment judgments. This limitation also reveals a significant absence of assessments of real interactive m-healthcare applications in Jordan to evaluate their security readiness level to support technical and non-technical stakeholders' decisions.

In practice, security decisions in m-health projects are distributed across business analysts, developers, quality engineers, and compliance stakeholders. However, current models in the literature on healthcare socio-technical applications fail to show this interactive distribution, which leads to limited traceability of security decisions, unclear accountability, and a lack of articulation of the continuous interaction of security assessments. Without a validated, interactive, lifecycle-based, and evidence-driven security

assessment framework, m-health application product owners, designers, developers, Quality Assurance (QA) engineers, medical professionals, and regulators lack a roadmap to evaluate current security capabilities at each phase. This may limit the ability to identify deficiencies and required best practices for each lifecycle phase.

As m-health continues to develop and expand healthcare delivery, the need for interactive SAMs becomes even more essential to increase awareness and protect patients, medical professionals, and healthcare systems from cybersecurity threats. This research aims to provide a theoretical contribution to SAMs and a practical interactive framework for m-health applications to evaluate security readiness and enhance it accordingly.

2. Research Methodology

The development of the proposed Phase–Role Interaction Security Assessment framework for mobile Health (PRISM-mH) was conducted based on the Design Science Research Methodology (DSRM) phases, which comprise problem identification and motivation, objectives of the proposed solution, design and development, evaluation of the solution, demonstration, and knowledge communication [31]. The six DSRM phases are structured and overlap to improve the output artifact. DSRM begins by bridging the gap between theory and practice and moves to designing the proposed artifacts that should be useful to humanity. The resulting artifact solution can be in the form of a framework, model, algorithm, method, or tool. The DSRM of this research is illustrated in Figure 1.

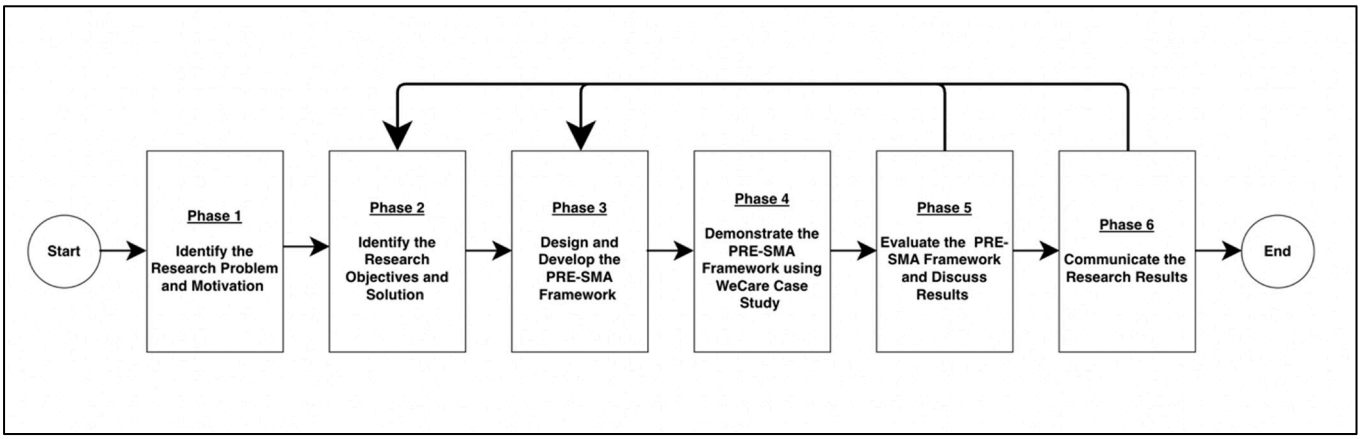


Figure 1. Research methodology.

2.1. Problem Identification and Motivation

Existing SAMs do not accurately reflect how security decisions are conducted for the specified interactions during m-health application development. Three main gaps were identified. First, few frameworks include auditing activities post-development; they describe assessment levels or compliance requirements at a high level [32], yet they rarely explain how security is assessed across specific development stages in an interactive manner [33–37]. In existing research, security appears as a final checkpoint or an external audit activity instead of being continuously shaped by requirements, design alternatives, coding practices, and deployment decisions.

The second gap is the limited attention to the interactive responsible roles. Security is often discussed in general terms, without linking it to the interacting roles involved in the development or governance teams accountable for security practices. In m-health projects, responsibilities are distributed across analysts, designers, developers, quality assurance engineers, project managers, and compliance officers. Without an explicit link between interacting roles, the m-health application, and security expectations, the security assess-

ment becomes vague and sometimes subjective. Third, existing assessment approaches do not require documented evidence and depend on scoring mechanisms with limited informal reporting [33–36]. Accountability and traceability of interactions are essential in healthcare-related systems, since any lack of evidence-based verification likely undermines the evaluation of security assessments.

2.2. Objectives of the Proposed PRISM-mH Framework

The primary goal of the PRISM-mH framework is to provide a structured, interaction-oriented security assessment approach for m-health applications that begins in the early development lifecycle phases, considering the roles involved and the required evidence to support each phase. Evidence support is needed to identify security gaps as early as possible. This framework shifts the current mindset of security assessment in m-health, and this can be translated into three sub-objectives as follows.

Objective 1: Unlike general security checklists, the PRISM-mH framework adopts security-based development lifecycle phases. Each phase extends to capture how users interact with system components, illustrated by responding to respective security requirements that are elaborated into interaction-focused assessment questions. This approach ensures that security is examined at the point of introduction and development. Through this systematic, structured evaluation across development phases, the framework identifies security gaps early that may otherwise remain hidden in later stages. When problems emerge in later development phases, fixing them becomes more difficult and expensive.

Objective 2: The PRISM-mH framework aims to make security interactions more visible and traceable by linking each phase to accountable roles, as m-health application development involves collaboration among multiple stakeholders. Without clear responsibility assignments, security could be inconsistently addressed.

Objective 3: The framework encourages the use of verifiable artifacts such as documented requirements, architecture design, quality assurance reports, user-interface screenshots, logs, and scan results. Evidence is critical in the m-health environment and security assessment outcomes to inform security readiness.

2.3. The Design and Development of the PRISM-mH Framework

PRISM-mH was designed and developed iteratively. Section 3 explains this framework. The work in this phase began with a systematic review of m-health application lifecycle models and m-health application security engineering models, leading to the identification of six fundamental lifecycle phases: Requirements Engineering, Architecture and Design, Implementation, Testing, Deployment, and Monitoring for maintenance, forming the core of the layers' hierarchy in the framework. The phases were aligned with software development lifecycle (SDLC) models [38,39] and cross-verified against security recommendations for m-health applications available in the scientific literature [40,41].

Table 1 shows how the six SDLC phases are mapped to the PRISM-mH framework. P indicates that this layer embodies this phase. C means that this layer is active during this phase and contributes to another layer(s). Blank means not applicable.

SDLC Phase-based mapping:

Layer 1: Business Modeling and Requirements → Requirements Engineering Phase;

Layer 2: Design and Architecture Security → Design and Architecture Phase;

Layer 3: Security Development (coding) → Implementation Phase;

Layer 6: Security Testing → Testing Phase;

Layer 7: Deployment and Distribution → Deployment Phase;

Layer 8: Monitoring and Incident Response → Monitoring and Maintenance Phase.

Table 1. Mapping PRISM-mH layers to SDLC phases.

	Layers/Phases	P1: Requirements Engineering	P2: Design and Architecture	P3: Implementation	P4: Testing	P5: Deployment	P6: Monitoring and Maintenance
Phase-Anchored Security Layers	L1: Business Modeling and Requirements	P					
	L2: Design and Architecture Security	C	P				
	L3: Security Development (coding)		C	P			
	L6: Security Testing			C	P		
	L7: Deployment and Distribution				C	P	
	L8: Monitoring and Incident Response					C	P
Cross-Cutting Layers	L4: Data Protection and Privacy	C	C	C	C	C	C
	L5: Authentication and Access Management		C	C	C		
	L9: Compliance and Governance	C	C	C	C	C	C
	L10: User Awareness and Usability		C	C		C	C

The remaining four layers are cross-cutting and span all phases:

Layer 4: Data Protection and Privacy;

Layer 5: Authentication and Access Management;

Layer 9: Compliance and Governance;

Layer 10: User Awareness and Usability.

The security requirements and content of each PRISM-mH layer were not directly derived from the source standards. The author developed them using a documented, five-step structured process to ensure that the selection, exclusion, merging, transformation, and validation of controls were clear and repeatable. The five-step process is explained below.

Step 1: Extraction. This step is part of the literature review and involved searching IEEE Xplore, ACM Digital Library, Google Scholar, Scopus, and ScienceDirect databases. These were complemented by four source standards: OWASP Mobile Application Security Testing Guide (MASTG) [42], information security management standards according to ISO/IEC 27001:2022 [43], security and privacy controls according to NIST Special Publication 800–53 Revision 5 [44], and OWASP Software Assurance Maturity Model (SAMM) [45], alongside prior m-health security models identified in the literature [22,25–30,40]. The search combined two sets of strings: m-health terms and security terms. The first set of search strings was: "mHealth ", "m-health", "healthcare application", and "mobile health". The second set of security terms search strings was: "security", "privacy", "security assessment", "security requirement", "data protection", and "authentication". The two sets were combined using the Boolean operators AND/OR.

Step 2: Screening and Exclusion. Each candidate control study was assessed with respect to inclusion and exclusion criteria. A candidate control study was considered for inclusion only if it (a) was relevant to the m-health users and their context of use, (b) depicted the user–system interaction (e.g., user-interface screenshot, consent log timestamp), and (c) was published in English. Candidate controls were excluded if they were entirely organizational or enterprise IT controls that lacked interaction impact, were applied to backend infrastructure, or were not published in English. Duplicate controls that had already been considered were also excluded. This filtering condition follows the core design principle in PRISM-mH, which states that the security of an interactive mobile application cannot be established from the existence of controls alone [46,47]. Table 2 shows the candidate control reduction from this step and their grouping into PRISM-mH requirements.

Step 3: Merging. Controls from different sources that the same security principle were grouped under one PRISM-mH control requirement. This was applied only if they coincided in three attributes: the security objective addressed, the user–system interaction points governed, and the type of evidence that would demonstrate fulfilment. This resulted in 79 requirements, as shown in Table 2.

Step 4: Transformation. The resulting controls from Step 3 were converted from the conventional presence-based question ("Is the control present/implemented?") to an interaction-oriented one, along with the evidence type. Then the author elaborated each security requirement into a set of security questions and mapped it to three dimensions: (a) the layer where it applies, (b) the role accountable for implementing the requirement or answering the question, and (c) the type of evidence at the interaction level to be gathered that indicates the requirement's fulfillment. The derivation of required evidence for each requirement is stakeholder-based and should not conflict with the sources.

Step 5: Validation and traceability. The PRISM-mH framework underwent two rounds of revisions. In the first round, after an internal review by an external digital health expert, overlapping layers were combined and/or divided, and the interactive-security assessment questions were relabeled to enhance the conceptual understanding. In the second round, after a structured review by a second external security expert, the layers were amended,

the evidence types were refined, and the identified assessment questions were validated too [31]. Steps 4 and 5 ended with the resulting 56 requirements, as shown in Table 2.

Table 2. The reduction in the number of candidate controls and their grouping into PRISM-mH requirements.

Stage	Count
Candidate controls extracted from standards and literature	214
Remaining controls after applying inclusion/exclusion criteria and removing duplicates	121
Distinct controls after consolidating overlaps	79
Final PRISM-mH requirements	56
Layer 1 requirements in PRISM-mH	6
Layer 2 requirements in PRISM-mH	4
Layer 3 requirements in PRISM-mH	4
Layer 4 requirements in PRISM-mH	5
Layer 5 requirements in PRISM-mH	5
Layer 6 requirements in PRISM-mH	7
Layer 7 requirements in PRISM-mH	7
Layer 8 requirements in PRISM-mH	5
Layer 9 requirements in PRISM-mH	8
Layer 10 requirements in PRISM-mH	5

Table 3 summarizes the experts' validation role, the data collection methods, the type of material reviewed, and how feedback was used to refine the PRISM-mH framework design. Table 4 shows the traceability records of all contributing source controls, the standards involved, and the action taken during the design of the PRISM-mH framework.

Table 3. Experts' input in the design and development of the PRISM-mH framework.

Aspect	What Was Reported?
Expert roles involved	The framework was designed and reviewed by three domain experts: the author and two external experts in the m-health security and software domains, with more than seven years of experience in medical systems and security.
Input method	Expert feedback was collected through structured review meetings that involved a walkthrough of the layers, security requirements/questions, assigned roles, and respective type of evidence. A draft version of PRISM-mH was provided during the review.
Material reviewed	Experts reviewed: (1) the PRISM-mH layer structure and definitions; (2) the mapping of security requirements/questions to lifecycle phases; (3) the evidence types associated with each layer; (4) the WeCare case study assessment judgments.
Use of feedback	Feedback led to: merging of overlapping layers, refinement of evidence type descriptions, revision of requirement labels for clarity, and reforming of assessment questions to make them interaction-oriented. Two full revision cycles occurred between the initial and final framework versions.
Disagreement/revision	Minor disagreements arose regarding the granularity of evidence types in the Development, Deployment and Monitoring phases. These were resolved through a follow-up discussion session, resulting in, for example, consolidation of three originally separate evidence items into one broader category.

Table 4. Traceability of PRISM-mH framework requirements to their source security controls.

Layer No.	Requirement ID	Requirement Description	Source	Derivation Action
1	BMR-UI-01	High-Risk User Interactions	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
1	BMR-UI-02	Misuse Interactions	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
1	BMR-UI-03	Documentation Of Interactions	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
1	BMR-UI-04	Compliance Of Interactions	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
1	BMR-UI-05	Interactions Of Data	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
1	BMR-UI-06	User Roles In Interactions	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
2	DAS-UI-01	Interaction Data Flow	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
2	DAS-UI-01	Interaction Isolation	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
2	DAS-UI-01	Adapting Risky Interactions	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
2	DAS-UI-01	Protocols Of Security Interactions	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
3	DS-UI-01	Runtime Interactions	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
3	DS-UI-02	User-Interface Protection	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
3	DS-UI-03	Overlay Attack Testing	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
3	DS-UI-04	Behavioral Monitoring Of Runtime Devices	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
4	DPP-UI-01	Informed Consent Presentation	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
4	DPP-UI-02	Privacy Policy Accessibility	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
4	DPP-UI-03	User-Confirmed Data Deletion	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
4	DPP-UI-04	Third-Party Data-Sharing Disclosure	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
4	DPP-UI-05	Periodic Privacy Preference Review	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
5	AAM-UI-01	Context-Sensitive Multi-Factor Authentication	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
5	AAM-UI-02	Session Timeout and Re-Authentication	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated

Table 4. *Cont.*

Layer No.	Requirement ID	Requirement Description	Source	Derivation Action
5	AAM-UI-03	Account Lockout and Secure Recovery	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
5	AAM-UI-04	Role Transparency and Access Clarity	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
5	AAM-UI-05	MFA Onboarding Prompt	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
6	TV-UI-01	Communication with Backend Services	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
6	TV-UI-02	Interaction Screens	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
6	TV-UI-03	Malicious Overlays	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
6	TV-UI-04	User Interaction Behavior	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
6	TV-UI-05	Authentication Testing	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
6	TV-UI-06	Compromised Interactions	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
6	TV-UI-07	Interactions Tracking Review	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
7	DD-UI-01	User Interaction with Verified Application	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
7	DD-UI-02	Protection of User Interaction from Vulnerable Versions	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
7	DD-UI-03	User Interaction Privacy at Runtime	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
7	DD-UI-04	Continuity of Trusted User Interaction Across Releases	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
7	DD-UI-05	User Interaction Trusted Channels	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
7	DD-UI-06	User Confidence in Interaction After Updates	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
7	DD-UI-07	Tracking of Interaction-Affecting Deployment Issues	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
8	MIR-UI-01	Compromised Device Warning	ISO 27001/NIST 800-53	Reformulated
8	MIR-UI-02	Suspicious Login Alert	ISO 27001/NIST 800-53	Reformulated
8	MIR-UI-03	Security Incident Notification	ISO 27001/NIST 800-53	Merged

Table 4. *Cont.*

Layer No.	Requirement ID	Requirement Description	Source	Derivation Action
8	MIR-UI-04	User-Accessible Activity Log	ISO 27001/NIST 800-53	Reformulated
8	MIR-UI-05	Unsafe Network User Decision Point	ISO 27001/NIST 800-53	Reformulated
9	CG-UI-01	Healthcare Regulatory Compliance	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
9	CG-UI-02	Interaction Governance Review	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
9	CG-UI-03	Third-Party SDK Assessment	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
9	CG-UI-04	Data Protection Impact Assessment	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
9	CG-UI-05	Regular Compliance Reviews	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
9	CG-UI-06	User Data Transparency	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
9	CG-UI-07	Incident Documentation and Review	MASTG/ISO 27001/NIST 800-53/SAMM	Merged
9	CG-UI-08	Responsibility Assignment	MASTG/ISO 27001/NIST 800-53/SAMM	Reformulated
10	UAU-UI-01	Security Onboarding Walkthrough	ISO 27001/NIST 800-53	Reformulated
10	UAU-UI-02	Proactive Security Notifications	ISO 27001/NIST 800-53	Reformulated
10	UAU-UI-03	Plain-Language Security Messaging	ISO 27001/NIST 800-53	Reformulated
10	UAU-UI-04	User-Driven Security Update Action	ISO 27001/NIST 800-53	Merged
10	UAU-UI-05	In-Application Help and Awareness Resource	ISO 27001/NIST 800-53	Reformulated

The following requirements related to the health sector have been extracted from sources related to the field: ISO 27799:2016 (Health Informatics—Information Security Management System (ISMS) in Health Using ISO/IEC 27002), security and privacy requirements under HIPAA/HITECH, GDPR Article 9 on processing special category (health) data, and the m-health security literature, as presented in Section 2 [22,25–30]. A candidate control was described as healthcare-related if the risk involved was specific to, or significantly more likely to occur in, the m-health environment, for instance, clinical and diagnostic data management, patient data processing consent, tele-consulting and prescription procedures, and the participation of users who do not necessarily have any expertise in the field. This step included creating new requirements that were not part of the general requirements, as well as refining the interaction points and evidence types associated with those requirements (such as the requirement for consent evidence before collecting health

data in Layer 4). Since m-health applications operate under different jurisdictions, PRISM-mH does not specify a single jurisdictional framework. The jurisdictional obligations are considered under the Compliance and Governance layer (Layer 9) by mapping regulations to interactions. Each regulation is mapped to the user–system interaction that it controls.

GDPR, HIPAA, and ISO/IEC 27001, along with Jordan's Personal Data Protection Law No.24 of 2023 (as the WeCare case study is based in Jordan), are the set of regulations considered for this research. Adopters who work within different legal frameworks should build the layer in question using the regulations they are subject to. This design approach ensures that the fundamental requirements of the framework remain independent of regulations while clearly stating the jurisdictional responsibilities (requirements CG-UI-01 through CG-UI-08 and DPP-UI-01 through DPP-UI-05). Table 5 summarizes the regulatory sources and interaction scope to which each was mapped.

Table 5. Healthcare and jurisdiction-specific regulatory sources and their interaction scope.

Regulation Source	Regulation Type	Mapped Interaction Scope	Primary Layers
ISO 27799:2016	Health Sector Security Standard	Handling clinical and health data across interactions.	L4, L9
HIPAA/HITECH	Jurisdictional (US)	Consent, access, breach notification interactions.	L4, L8, L9
GDPR	Jurisdictional (EU)	Consent, transparency, deletion, third-party sharing.	L4, L9
ISO/IEC 27002	General ISMS	Governance and control review interactions.	L9
Jordan's Personal Data Protection Law No.24 of 2023	Jurisdictional (Jordan)	Transparency in processing data at the point it is collected or used. Controls on third party data sharing. Notifications on processing, sharing, and breaching.	L4, L9

2.4. Demonstration of the PRISM-mH Framework Using the WeCare Study

PRISM-mH is demonstrated through a practical example using the WeCare application. WeCare was chosen as a pilot demonstration because it is classified as an m-health application that manages user interactions with sensitive data. WeCare assessment was performed by reviewing various evidence sources for each of the ten layers. This included requirements documentation, architecture documentation, data flow diagrams, user-interface screenshots, quality assurance reports, authentication configuration details, deployment records, usability test results, and compliance documentation. Where evidence was submitted, it was reviewed against the corresponding stated requirement/questions. When no evidence was provided, its absence was considered a finding itself.

To assess security requirement implementation in WeCare, five assessment labels are identified (yes, no, missing evidence, partial, and not applicable) to denote the extent to which WeCare fulfilled the required security implementation as a medical system. During the security requirement assessment, evidence is examined in one of two ways: direct or indirect. Blank is used when evidence examination is not applicable. After assessment, the identified gap should be described using one of the five labels. Section 4 presents the detailed meanings of the evaluation labels used in WeCare.

Applying the framework provided a structured basis for reviewing the WeCare team's security practices, indicating areas for potential improvement and those for consideration in future development cycles.

2.5. Evaluation of the PRISM-mH Framework

This phase was conducted to assess the usefulness and effectiveness of the PRISM-mH framework after applying it to WeCare. In particular, the evaluation focused on how effectively the framework captured security considerations across development phases and how each responsible role could provide evidence to support the security assessment.

The PRISM-mH framework's content was evaluated independently by a panel of 3 experts, who were not involved in PRISM-mH development, described in Section 2.3. The selection criteria for the panel members included holding a postgraduate degree or professional certification in software security, health informatics, or secure/agile software engineering. The three experts needed to have at least 5 years of experience in the field and proficiency in at least one domain of PRISM-mH. The questionnaire was used individually by each expert for the evaluation of the framework. The rating of four qualities, namely, relevance, clarity, completeness, and feasibility, for each layer and all requirements in the layers was done by the experts based on a scale of four points, ranging from not relevant/clear/complete/feasible to highly relevant/clear/complete/feasible. The same four-point rating was used to evaluate the content, that is, layer requirements. Then, these ratings were aggregated to generate the evaluation results. If there was any disagreement among the experts on any of the items, the items were revised until the requirements were met or until the item was dropped altogether. Any disagreements were settled by consensus rather than by the judgment of the author.

Finally, this phase allowed WeCare owners to gain knowledge of security practices that were clearly defined in the implemented interactions, as well as areas where documentation was limited or responsibilities were not formally assigned.

2.6. Communication

This is the final phase in the adopted DSRM. It involved documenting and publishing the structural design of the interactive PRISM-mH framework and disseminating the findings from the WeCare case study to both academic and practitioner audiences. This is to share the rationale behind the framework design and demonstrate applicability in m-health applications to support adoption in relevant medical systems.

3. The PRISM-mH Framework

The PRISM-mH framework is a socio-technical structured security assessment approach that shapes how security is identified, handled, communicated, and monitored continuously in m-health throughout all the development lifecycle phases. Security assessment progresses incrementally across all phases rather than being conducted at the end of development. The framework extends traditional software engineering phases by embedding an interaction perspective, linking each phase to how security is experienced and enacted through user–system interaction rather than focusing solely on its technical implementation. In the PRISM-mH framework, security is managed with respect to when it was introduced, who was responsible, and whether documentation support was available, as shown in Table 6. Therefore, the PRISM-mH framework is based on three pillars: the development phase, the assigned role, and the required evidence.

Table 6. The PRISM-mH framework.

Phase	Responsible Role	Security Assessment Requirement/Questions	Evidence
1—Business Modeling & Requirements	Business Analyst/Product Owner	BMR-UI-01—High-Risk User Interaction Workflows 1. Have high-risk user–system interaction points, workflows, and threat scenarios been identified and modeled at specific interaction points within the business case and requirement specification? 2. Has the system's handling of high-risk interaction points been specified within requirement specification? 3. Does the supporting evidence reflect how these scenarios unfold during actual user engagement?	High-risk user–system interaction workflows and threat scenarios Business case document Requirement specification
	Business Analyst/Product Owner	BMR-UI-02—Misuse Interaction 1. Have misuse interaction scenarios been identified and documented with sufficient detail to show how a user or attacker could exploit each interaction point? 2. Is there evidence that these misuse cases were derived from realistic user–system interaction flows? 3. Is the risk rank of each misuse interaction identified? 4. Have the consequences and impact of each misuse interaction point on the user and the system been identified?	Misuse cases
	Business Analyst/Product Owner	BMR-UI-03—Interaction Documentation 1. Are user–system interaction security risks formally documented at each trust boundary and risky interaction point? 2. Does the documentation include a workflow model showing how patients encounter these points during real interaction with the system?	Trust boundary diagram User interaction workflow model that shows risky interaction points between patient and system
	Compliance Officer	BMR-UI-04—Compliance of Interactions 1. Have regulatory obligations (GDPR, HIPAA, ISO 27001, etc.) been mapped to specific user–system interaction points, including the risky ones? 2. Is there documented evidence showing which regulatory requirement governs each interaction point at which user data is collected, accessed, or transmitted?	Compliance checklist Legal review report
	Security Lead	BMR-UI-05—Interaction Data 1. Have data sensitivity levels been classified in relation to each user–system interaction? 2. Does a data classification matrix link sensitivity category to the specific interaction points at which sensitive data is collected, processed, or shared?	Data classification matrix for each interaction point
	Project Manager	BMR-UI-06—User Roles in Interactions 1. Has a security focal point been assigned? 2. Do the requirements specify how each user role is permitted to access data at each interaction point, with a RACI chart and role assignment document confirming accountability for security decisions across interaction workflows?	RACI chart Role assignment document

Table 6. *Cont.*

2—Design and Architecture Security	Solution Architect/Security Architect/Developer	DAS-UI-01—Interaction Data Flow 1. At which user–system interaction points are security keys exchanged or required during the m-health workflow? 2. How does the architectural design protect continuity at each of those points? 3. Is there evidence in interaction data flow diagrams showing that key management decisions were derived from the actual interaction flow rather than from generic technical requirements?	Interaction data flow diagrams System architecture
	Solution Architect/Security Architect	DAS-UI-02—Interaction Isolation 1. At which user-context transition points could one user's session be observed by another during interaction? 2. How does the architectural design isolate each session at those specific transitions? 3. Is there evidence in secure session models and user-context transition maps showing that the isolation logic was shaped by the actual sequence of interactions between user and system?	Secure session models User-context transition maps
	Solution Architect/Security Architect/Security Expert	DAS-UI-03—Risky Interaction Adaptation 1. Which risky interaction contexts—such as authentication over an untrusted network, or data submission from a shared device—have been identified? 2. How does the architecture change its trust and communication behavior at each of those risky interaction points? 3. Is there evidence in communication trust sequence diagrams showing the specific architectural responses triggered by each real user interaction scenario?	Communication trust diagrams Sequence diagrams System architecture
	Security Architect/Security Expert/Network Engineer	DAS-UI-04—Interaction Security Protocols 1. At which user interaction points—such as login, data transmission, and session handoff—are security protocols applied to protect the exchange? 2. How are those protocols explicitly represented in the architecture design for each interaction scenario? 3. Is there evidence in architecture diagrams and sequence documentation that protocol decisions were mapped to the specific interaction contexts they were designed to govern?	Security protocols shown in architecture and diagrams.
	Developer	DS-UI-01—Runtime Interactions 1. At which runtime interaction points—such as authentication, symptom entry, medical record access, or payment—is the user most exposed to manipulation attempts? 2. How does the code detect and prevent interference at each of those specific moments? 3. Is there evidence from runtime attack simulations and behavioral monitoring logs confirming that interaction integrity was tested and maintained at each of those points?	Runtime interaction attack simulations Interaction manipulation test logs Runtime behavioral monitoring logs

Table 6. *Cont.*

3—Security Development (Code Level)	Developer	DS-UI-02—User-Interface Protection 1. At which interaction screens does the user enter or view sensitive healthcare information—such as diagnosis details, credentials, or prescriptions? 2. How does the implementation prevent external capture or monitoring of those specific screens during live user interaction? 3. Is there evidence from user-interface protection reports confirming that protection was verified at the exact interaction points where sensitive content is displayed or entered?	User-interface protection reports Runtime interaction attack simulations
	Developer/Security Expert	DS-UI-03—Overlay Attack Testing 1. At which healthcare interaction points—such as login screens, prescription entry, or payment confirmation—could a malicious overlay deceive a user into entering sensitive data on a fake interface? 2. How does the application detect overlay attempts and respond at each of those interaction points? 3. Is there evidence from overlay attack testing showing what the user sees and experiences when an overlay is detected during a healthcare interaction?	Overlay attack testing results, Interaction manipulation test logs User-interface protection reports
	Developer/Security Expert	DS-UI-04—Runtime Device Behavioral Monitoring 1. At which interaction points could a compromised device silently alter what the user inputs, sees, or submits—such as during authentication or medical data entry? 2. How does the application detect device compromise and protect the interaction at those moments? 3. Is there evidence from runtime behavioral monitoring logs and session hijacking tests showing that the user's interaction remains protected even when the underlying device environment is untrusted?	Runtime behavioral monitoring logs Runtime interaction attack simulations Session hijacking prevention evidence, Interaction manipulation test logs.
	Developer/DBA/UX/UI Designer/QA Engineer	DPP-UI-01—Informed Consent Presentation 1. At which interaction point before data collection begins does the user encounter the consent screen? 2. How is the consent presented so that it is understandable to a non-technical healthcare user rather than dense legal language? 3. Is there a timestamped record of the user's consent action and a test report confirming that no data is collected prior to the user's explicit interaction with the consent screen?	Timestamped consent log per session UI screenshot of consent screen at first launch Test report confirming no data collection precedes consent

Table 6. *Cont.*

4—Data Protection & Privacy	UX/UI Designer/QA Engineer/Product Owner	DPP-UI-02—In-Application Privacy Policy Accessibility 1. At which interaction point within the application does the user encounter access to the privacy policy? 2. How many interaction steps does it take to reach privacy policy from the main interface? 3. Is the policy rendered within the application rather than redirecting the user to an external site? 4. Is there evidence from a usability review confirming that the privacy information is accessible and readable during normal use without leaving the interaction context of the application?	Usability review confirming access within two taps UI screenshot showing in-app privacy policy rendering Onboarding walkthrough confirming privacy link is visible
	DBA/Developer/UX/UI Designer/QA Engineer	DPP-UI-03—User-Confirmed Data Deletion 1. At which interaction point does the user receive confirmation that their data has been deleted or their account closed? 2. What does the user see at that deletion or closing moment? 3. Is it clear that the deletion or closing action has been completed? 4. Is there a UI screenshot of the confirmation message and a backend deletion audit log with a timestamp confirming that deletion occurred at the exact moment the user initiated the interaction?	Backend deletion audit log with timestamp UI screenshot of deletion confirmation message QA test report verifying consistent confirmation behavior UI screenshot of named disclosure screen
	Compliance Officer/DPO/Developer/DBA	DPP-UI-04—Third-Party Data-Sharing Disclosure 1. At which interaction point before data sharing is completed does the user encounter a disclosure of which third parties will receive their data? 2. How is each third party identified in plain language at that moment rather than in buried terms and conditions? 3. Is there evidence that the disclosure interaction gives the user a genuine and understandable opportunity to approve or deny each sharing action before it takes effect?	Compliance review confirming HIPAA/GDPR alignment Acknowledgment log recorded per sharing event Application log of review prompts per user account
	UX/UI Designer/QA Engineer/Developer	DPP-UI-05—Periodic Privacy Preference Review 1. At which interaction points within the application—such as after a policy update or at a defined interval—does the user encounter a prompt to review their privacy and data-sharing preferences? 2. How is the prompt designed to guide the user through their current settings in understandable language? 3. Is there evidence from a UI screenshot showing the interaction moment at which the prompt appears and how the user is guided through updating their preferences?	UI screenshot of proactive privacy prompt Confirmation that preference update completes within the same session

Table 6. *Cont.*

5—Authentication & Access Management	Security Architect/Security Manager/UX/UI Designer/QA Engineer	AAM-UI-01—Context-Sensitive Multi-Factor Authentication 1. At which specific interaction points within the application—such as opening a medical record, initiating a payment, or changing account credentials—is multi-factor authentication triggered? 2. How is the verification step explained to the user at that precise moment so they understand what is being asked? and why? 3. Is there evidence from test logs and UI documentation showing that users can complete the MFA process at each of those interaction points without errors or confusion?	User interaction log confirming MFA prompt at sensitive access points UI screenshot of MFA verification screen Usability test confirming users successfully complete the step
	Developer/Security Manager/QA Engineer	AAM-UI-02—Session Timeout & Re-Authentication 1. At which interaction point does the session timeout occur and what does the user see at that moment? 2. How is the timeout message written so that a non-technical user understands they need to log in again rather than interpreting it as an error? 3. Is there evidence from UI screenshots and test logs confirming that the re-authentication interaction presented at that point is clear, friendly, and does not result in loss of unsaved user data?	UI screenshot of session expiry notification Usability test confirming users understand why they were logged out Interaction log confirming re-authentication required before any further action
	QA Engineer/Developer/UX/UI Designer	AAM-UI-03—Account Lockout & Secure Recovery 1. At which interaction point is the user informed that their account has been temporarily locked? 2. What language and guidance does the user encounter at that exact moment? 3. Is there evidence from UI screenshots and QA test logs confirming that the recovery process presented at that interaction point requires no technical knowledge—and that a typical m-health user can complete it independently without contacting support?	UI screenshot of lockout notification message UI screenshot of step-by-step recovery guidance Usability test confirming users complete recovery without confusion
	Product Owner/UX/UI Designer/	AAM-UI-04—Role Transparency & Access Clarity 1. At which interaction points within the application does the user encounter a restricted area or feature they cannot access? 2. How is the access boundary communicated to the user at each of those moments so they understand what they can and cannot do rather than receiving a generic error? 3. Is there evidence from UI screenshots and RBAC documentation confirming that users can clearly understand their own interaction permissions during normal use of the application?	UI screenshot showing access boundary message Usability test confirming users understand their permissions User feedback confirming role clarity during interaction

Table 6. *Cont.*

	Product Owner, Security Manager	AAM-UI-05—MFA Onboarding Prompt 1. At which interaction point during first use does the user encounter the MFA onboarding prompt? 2. How is the benefit of MFA explained at that specific moment in language that makes sense to a non-technical healthcare user rather than a technical one? 3. Is there evidence from UI mockups and test logs showing that users can complete the MFA setup at that interaction point independently without requiring help from a support team?	UI screenshot of MFA setup prompt during onboarding Usability test confirming users understand the purpose of MFA Interaction log confirming MFA setup is offered at first-use interaction
6—Testing & Validation	QA/Penetration Tester	TV-UI-01—Communication with Backend Services 1. At which interaction points does the user's data travel between the application and backend services—such as during login, record submission, or health data retrieval? 2. Has interception been tested at each of those specific points? 3. Is there evidence from communication security test logs showing that data exchanged during real user interaction scenarios cannot be intercepted or modified in transit?	Penetration testing reports MITM validation results
	QA/Security Tester	TV-UI-02—Interaction Screens 1. At which interaction screens containing sensitive healthcare data—such as medical records, prescription details, or authentication screens? 2. Has screen capture protection been specifically tested? 3. How were the tests designed to simulate real-world capture attempts during active user interaction rather than in isolation? 4. Is there evidence from protection test reports confirming that each of those interaction screens resists external observation?	Screen protection validation reports QA reports
	QA/Security Tester	TV-UI-03—Malicious Overlays 1. At which healthcare interaction points—such as login, symptom reporting, medication entry, or payment—has overlay attack testing been conducted? 2. What did the application do when an overlay was simulated at each of those interaction points? 3. Is there evidence from overlay attack test results showing both the attack scenario tested and the application's response at the point of user interaction?	Overlay attack testing results Runtime interaction test logs
	Security Tester	TV-UI-04—User Interaction Behavior 1. At which runtime interaction points—such as data entry, form submission, or session handoff—has user behavior been tested against manipulation attempts? 2. How were the tests designed to reflect realistic user actions at those specific interaction moments rather than generic system tests? 3. Is there evidence from runtime behavioral monitoring logs and interaction manipulation test logs showing what the user would experience if manipulation were attempted at each of those points?	Runtime interaction attack simulations Runtime monitoring reports

Table 6. *Cont.*

6—Testing & Validation	QA/Security Lead	TV-UI-05—Authentication Testing 1. At which authentication and session interaction points—including login, step-up authentication, session handoff, and logout—has protection been verified? 2. How was continuity of protection tested across the full interaction lifecycle from the user's first authentication to the end of their session? 3. Is there evidence from test logs confirming that session security is maintained at each step the user takes from entry to exit?	Authentication testing logs Session validation reports
	Security Tester	TV-UI-06—Compromised interactions 1. At which user interaction points—such as authentication, data submission, or active session use—could a compromised device silently influence what the user sees or sends? 2. How were that specific interaction points tested under simulated device compromise conditions? 3. Is there evidence from emulator-based tests and runtime behavioral monitoring logs showing that the user's interaction experience and data remain protected even when the device environment is untrusted?	Root/jailbreak testing results Emulator detection reports
	Project Manager	TV-UI-07—Interaction Tracking Review 1. At which interaction points have security issues been identified during development—such as a vulnerable login flow or an unprotected data entry screen? 2. How are those issues tracked from the interaction scenario in which they were discovered to resolution? 3. Is there evidence in a tracking review report that each issue is linked to the specific interaction it affects and that resolution was verified at that interaction point before the issue was closed?	Review meeting records Issue tracking reports
7—Deployment & Distribution	DevOps Engineer	DD-UI01—User Interaction with Verified Application 1. At the point when a user first downloads or opens the application, how is it ensured that the version they are interacting with is authentic and has not been tampered with? 2. What verification mechanism is in place at that interaction moment? 3. Is there evidence in the release process documentation and code signing records showing that integrity verification is completed before the application reaches the user's interaction environment?	Signing certificate Release process document

Table 6. *Cont.*

7—Deployment & Distribution	DevOps Engineer	DD-UI02—Protection of User Interaction from Vulnerable Versions 1. At which interaction point is the user informed or protected when they are running an outdated or unsafe version of the application? 2. How does the app communicate this risk to the user in a way they can understand and act on? 3. Is there evidence that a version management process detects vulnerable versions and alerts users at the point of interaction before they continue using a version that puts their data at risk?	Version control records Update policy documentation
	Developer	DD-UI03—User Interaction Privacy at Runtime 1. At which interaction points during live use could a diagnostic or debugging interface expose sensitive healthcare information—such as session tokens, health records, or credentials? 2. How has the production build been configured to prevent this exposure during real user interaction? 3. Is there evidence confirming that all debugging features were verified as disabled before users first interacted with the production version of the application?	Build configuration evidence Production settings
	Security Expert	DD-UI04—Continuity of Trusted User Interaction Across Releases 1. Across each release and update cycle, how is the security of user interactions specifically verified before the updated version is placed in front of users? 2. What interaction-level testing is conducted as part of the release process? 3. Is there evidence that an interaction-focused security review is completed and documented for each deployment before users begin interacting with the new version?	Certificate management logs Release validation reports
	DevOps/Release Manager/Security Lead	DD-UI05—User Interaction Trusted Channels 1. Through which channels does the user first encounter and download the application? 2. How has the authenticity of each distribution channel been verified so that users do not inadvertently interact with a tampered or counterfeit version? 3. Is there evidence confirming that the application is exclusively distributed through authenticated channels and that no interaction with the app is possible through unverified sources?	Official store release records Signed application evidence Release management documentation.

Table 6. *Cont.*

7—Deployment & Distribution	Security Lead	DD-UI06—User Confidence in Interaction After Updates 1. At the interaction point when a user encounters an application update, how does the user know the update originates from a legitimate and trusted source rather than a malicious one? 2. What verification evidence is visible or available to the user at that moment? 3. Is there evidence of a code signing and update verification process that protects the user's interaction with every update from the source to installation?	Update verification policies Release signing evidence
	Release Manager	DD-UI07—Tracking of Interaction-Affecting Deployment Issues 1. When a deployment issue is identified that affects the quality or safety of user interaction—such as a release that introduces a broken authentication flow or exposes a sensitive screen—at which interaction points is the impact felt? 2. How is the issue formally recorded in relation to those specific interaction scenarios? 3. Is there evidence in deployment issue tracking reports showing that every interaction-affecting issue is traced from detection to verified resolution before the affected version continues to be used?	Release review records Deployment issue tracking reports
8—Monitoring & Incident Response	Security Operations Center (SOC)/Security Analyst/UI/UX Designer/QA Engineer	MIR-UI-01—Compromised Device Warning 1. At which interaction point does the user encounter a warning when their device is detected as compromised—such as immediately at login or during an active session? 2. What language and guidance does the user see at that exact moment? 3. Is there evidence from a UI screenshot of the warning message showing that the alert is presented in plain, non-technical language that an m-health user can understand and immediately act upon without needing additional instruction?	UI screenshot or push notification template of the alert Interaction log recording user confirmation or block action Usability test confirming users understand and act on the alert
	Security Analyst/Incident Response (IR) lead	MIR-UI-02—Suspicious Login Alert 1. At which interaction point is the user alerted when a login is attempted from an unrecognized device—is it a push notification, an in-app message, or an email? 2. What does the user see at that alerting moment? 3. What choices are presented to them? 4. Is there evidence showing the alert format and the full interaction flow through which the user can confirm, deny, or block the suspicious attempt without needing technical knowledge?	Sample breach notification message or template IR plan confirming user communication protocol is included IR drill report confirming notification process has been tested

Table 6. *Cont.*

8—Monitoring & Incident Response	Network Security/QA Engineer/SOC	MIR-UI-03—Security Incident Notification 1. At which interaction point is the user informed when a security incident has affected their account—during active use, at next login, or via a push notification? 2. What does the user see and experience at that specific security incident moment? 3. Is there evidence from a push notification template or in-application alert screenshot showing that the information presented at that interaction point tells the user exactly what happened and what steps to take, in language free of technical jargon?	UI screenshot of user-accessible activity log Test log confirming session history accuracy Privacy policy section explaining the activity log to the user
	SOC Manager/QA Engineer/UI/UX designer/	MIR-UI-04—User-Accessible Activity Log 1. At which interaction point within the application can users access their recent login and data access history? 2. How is the information presented so that a non-technical user can understand what it means and identify anything suspicious? 3. Is there evidence from a UI screenshot of the activity log showing that the information is readable, meaningful, and reachable through a clear interaction path during normal use of the application?	UI screenshot of network warning with user choice options Usability test confirming users understand the risk message Interaction log recording user decision at the warning point
	IR Lead/QA Engineer/UI/UX designer/Network Security	MIR-UI-05—Unsafe Network User Decision Point 1. At which interaction point does the user encounter a warning when an unsafe network is detected—at login, upon connection, or during active data transfer? 2. What decision is the user presented with at that warning moment and in what language? 3. Is there evidence from a UI screenshot of the network warning showing that the user is given clear, non-technical options and guidance to make an informed security decision before continuing their interaction with the application?	UI screenshot or push notification template of the alert Interaction log recording user confirmation or block action Usability test confirming users understand and act on the alert
9—Compliance & Governance	Compliance Officer	CG-UI01—Healthcare Regulatory Compliance 1. At which user–system interaction points do healthcare regulations such as GDPR, HIPAA, and ISO 27001 impose specific requirements—such as consent at data collection, access control at record retrieval, or audit logging at sensitive transactions? 2. How has each regulatory obligation been mapped to the interaction scenario it applies to? 3. Is there evidence in a compliance checklist and legal review report confirming that each regulated interaction point has been assessed and found compliant?	Compliance audit report

Table 6. *Cont.*

9—Compliance & Governance	Security Officer	CG-UI02—Interaction Governance Review 1. At which interaction points involving healthcare data—such as data entry, access, sharing, or deletion—are governance controls applied to ensure secure and compliant behavior? 2. How are those specific interaction-level processes reviewed at a governance level? 3. Is there evidence in an interaction governance review record showing that the actual conduct of user–system interactions—not just system configurations—has been assessed against security and privacy standards?	Security governance reports Review documentation
	Security Lead	CG-UI03—Third-Party SDK Assessment 1. At which user interaction points do third-party SDKs or external services participate in handling, processing, or displaying the user's data—such as an analytics SDK active during health record access, or an authentication library used at login? 2. How has each integrated component been assessed for its effect on the security of those specific interactions? 3. Is there evidence in a third-party assessment report showing interaction-level security findings for each component and any mitigations applied?	Vendor risk report Third-party assessment records
	Data Protection Officer (DPO)	CG-UI04—Data Protection Impact Assessment 1. At which user–system interaction points is personal health data processed in ways that carry privacy risk—such as at the point of data collection, sharing with a third party, or storage after a session ends? 2. How have those interaction-level processing activities been assessed under applicable data protection requirements? 3. Is there a completed Data Protection Impact Assessment (DPIA) documenting the specific privacy risks identified at each interaction point and the mitigations that have been applied?	DPIA report
	Governance Board	CG-UI05—Regular Compliance Reviews 1. At which scheduled intervals are the security and compliance of user–system interactions formally reviewed—including assessments of whether interaction-level controls remain effective and whether new interaction risks have emerged? 2. How do those reviews specifically examine interaction practices rather than only system configurations? 3. Is there evidence of completed review reports at defined intervals showing that interaction-level compliance has been assessed against current regulatory standards?	Review schedules Governance meeting records

Table 6. *Cont.*

9—Compliance & Governance	Compliance Officer	CG-UI06—User Data Transparency 1. At which interaction points within the application does the user encounter information about how their data is collected, with whom it is shared, and how it is protected? 2. Is this presented within the interaction itself or only in an external document the user must seek out? 3. How is the information written so that a non-technical user can understand it at the moment of interaction? 4. Is there evidence that users encounter meaningful data transparency at the specific interaction points where their data is processed, rather than only in buried policy text?	Privacy policy reviews User consent records
	Security Officer	CG-UI07—Incident Documentation & Review 1. When an interaction-related security incident occurs—such as a session being hijacked during medical record access or consent bypassed during data collection—at which interaction point did it originate and how was the user affected at that moment? 2. How is the incident formally documented in relation to that specific interaction scenario? 3. Is there evidence in an incident documentation record that links each incident to the interaction point at which it occurred and identifies the compliance implications under applicable healthcare regulations?	Incident review records Compliance tracking logs
	Governance Board	CG-UI08—Responsibility Assignment 1. For each interaction point at which a user's healthcare data is at risk—such as during consent collection, session management, or third-party data transfer—how have responsibilities for protecting that specific interaction been assigned at the governance level? 2. Who is accountable when an interaction-level security failure occurs? 3. Is there evidence of a governance responsibility matrix showing which named roles are responsible for securing and monitoring each category of user–system interaction within the m-health application?	Governance policies Role assignment documentation
10—User Awareness & Usability	UX Designer/Security Officer	UAU-UI-01—Security Onboarding Walkthrough 1. At which interaction point during first use does the security onboarding walkthrough appear? 2. How does it guide the user through each key security feature—such as setting up authentication, understanding privacy settings, and recognizing security alerts—using language and visuals appropriate for a non-technical healthcare user? 3. Is there evidence from a UI mockup or usability test showing that a first-time user can complete the onboarding interaction and correctly act on the security guidance they receive?	UI mockup of onboarding security flow Usability test results confirming user comprehension User journey map showing security steps embedded in first use

Table 6. *Cont.*

10—User Awareness & Usability	Developer/UX Designer/Security Officer	UAU-UI-02—Proactive Security Notifications 1. At which interaction point within the application does the user encounter a notification when a security action is required—such as a password expiry warning appearing before the session begins rather than only after lockout occurs? 2. How is the notification written so that the user understands what action is needed and why it protects them? 3. Is there evidence from UI screenshots or test records showing that the notification is triggered proactively at the right interaction moment rather than leaving the user to discover the issue independently?	UI screenshot of in-app proactive security notification QA test log confirming notifications appear at correct interaction point Interaction log showing user response to the notification
	UX Designer/Product owner/Developer	UAU-UI-03—Plain-Language Security Messaging 1. At each interaction point where the user encounters a security warning, prompt, or instruction—such as a consent screen, an MFA prompt, a session timeout message, or a security alert? 2. How has the language been designed and verified to be free of technical jargon and understandable to an everyday m-health user with no security background? 3. Is there evidence from a plain-language or usability review confirming that each security message was evaluated against the comprehension needs of a non-technical healthcare user before deployment?	Plain-language audit report of all security-facing messages Microcopy review confirming absence of technical terms User guide excerpt showing non-technical security explanations
	UX Designer/Security Officer	UAU-UI-04—User-Driven Security Update Action 1. At which interaction point does the user encounter a notification that a security-relevant update is available? 2. How is the significance of the update explained to the user at that moment in terms of what it means for their own health data safety rather than in technical version language? 3. Is there evidence from a UI screenshot of the update notification and interaction path showing that the user is guided to act voluntarily with a clear and plain-language understanding of why the update matters for their security?	UI screenshot of update recommendation message Help content explaining the security benefit of updating Interaction log recording user response to update prompt
	Communications Team/UX designer	UAU-UI-05—In-Application Help & Awareness Resource 1. At which interaction point within the application can users access the help section containing security guidance, privacy explanations, and FAQs? 2. How many interaction steps does it take to reach from a key point of use such as the main screen or a security alert? 3. How is the content written to serve a non-technical m-health user who encounters a security concern during real use? 4. Is there evidence that the help content is accessible within the app's interaction flow, written in everyday language, and reviewed on a regular basis to remain accurate and current?	User manual or in-application help content sample Awareness campaign materials (FAQs, security tips) Content review log confirming regular updates aligned with application changes

It is necessary to highlight that the PRISM-mH framework is not a conventional security assessment method; instead, it is an interaction-oriented assessment tool. Conventional

security assessments ask whether a given control exists—whether it has been implemented, configured, or documented. PRISM-mH takes a fundamentally different stance. Rather than confirming the presence or absence of a control, PRISM-mH asks how that control appears during real user–system interaction: at which points users encounter it, whether they understand it, whether observable evidence exists at the point of interaction, and how the interaction shapes security decisions and outcomes. This distinction is consequential in m-health contexts, where a technically implemented control may still fail to protect users if it is invisible, confusing, or bypassed during actual use. Therefore, PRISM-mH frames each assessment requirement as an interaction-centered question, shifting the focus from "control present or absent" to "security as experienced and evidenced during user–system interaction." This interaction-oriented perspective is applied consistently across all ten layers of PRISM-mH. Each requirement in the framework is therefore expressed not as a system property to verify but as an interaction-centered question that examines whether security is visible, usable, and evidenced at the point where the user and system meet. The following examples illustrate this contrast across representative requirements:

Conventional: Are high-risk interaction workflows identified? (Layer 1).

PRISM-mH: Have high-risk user–system interaction points, workflows, and threat scenarios been identified and modeled at specific interaction points within the business case and requirement specification?

Conventional: Is consent collected? (Layer 4).

PRISM-mH: At which interaction point before data collection begins does the user encounter the consent screen, and is there a timestamped record of the user's consent action and a test report confirming that no data is collected prior to the user's explicit interaction with the consent screen?

3.1. Layer 1: Business Modeling and Requirements

This layer marks the earliest phase of m-health interactive development. Its purpose is not simply to enumerate security requirements but to identify and model how users are anticipated to securely interact with healthcare services, interfaces, and devices during sensitive healthcare workflows—and what could go wrong at each of those interaction points. PRISM-mH frames this layer around interaction-centered questions: At which points in the user's journey does a security risk emerge? How do users encounter that risk? How should the system respond at the point of interaction? And what evidence from the interaction itself confirms that the risk has been addressed?

The business analyst maps high-risk user–system interaction workflows—such as a patient logging in to access medical records, a clinician submitting a prescription, or a user authorizing a payment for a home visit—and models misuse scenarios that show how a user or attacker could exploit each interaction point. Rather than treating regulations such as HIPAA or GDPR as a compliance checklist, this layer maps each regulatory obligation to the specific interaction scenario it governs: which regulation applies when a user consents to data sharing, which applies when a clinician accesses a patient record remotely, and which applies at the payment confirmation screen.

The evidence produced at this layer is therefore interaction-grounded. It includes trust boundary diagrams and user journey security maps showing where the user is exposed to risk during real use, misuse-case models capturing how each vulnerable interaction point could be exploited, a compliance checklist that links each regulatory obligation to the interaction it governs, a data classification matrix that associates sensitivity levels with the interaction points at which that data is exchanged, and a Responsible, Accountable, Consulted, and Informed (RACI) chart that assigns accountability for security decisions to specific roles across the interaction workflow. Any change to a user–system interaction in

the m-health application requires a corresponding revision to the interaction models and associated evidence at this layer.

3.2. Layer 2: Design and Architecture Security

The purpose of this layer is not only to develop infrastructure that ensures prevention of any access to data by unauthorized parties but to create an infrastructure that would protect specific interaction points where users and system components exchange sensitive data and where evidence exists that the design was influenced by the specific interaction process and not by technical conventions. PRISM-mH sets the following interaction-based questions for the layer: What interaction points in the m-health workflow are used for exchanging security keys, and does the interaction data flow diagram contain any evidence that security key management was performed based on the interaction process? At which user-context transition points could unauthorized observation occur, and do secure session models and user-context transition maps demonstrate how isolation was designed for each transition? Which risky interaction contexts—such as authentication over an untrusted network or data submission from a shared device—have been identified, and are there communication trust sequence diagrams showing how the architecture adapts at each of those points? At which user interaction points are security protocols applied, and are they explicitly represented in architectural documentation for each interaction scenario they govern?

The evidence at this layer is therefore interaction-structured: interaction data flow diagrams that trace the path of sensitive data through each user–system interaction, secure session models and user-context transition maps that show how the architecture isolates each interaction session, communication trust sequence diagrams demonstrating how architectural trust is established and maintained at each risky interaction point, and architectural documentation showing which security protocols apply at which user interaction scenarios. Architectural design patterns that support secure and controlled interactions are the multi-layer architecture [48], the zero-trust architecture [49], and the microservice architecture [50].

3.3. Layer 3: Security Development (Coding)

This layer translates interaction-centered architectural security decisions into code, determining how security is experienced by users during real-time interactions with the m-health application. PRISM-mH frames this layer around interaction-oriented questions that ask not whether security controls exist in the code but how they protect the user at the moment of interaction. Where are the critical points of runtime interaction when the user is most vulnerable to manipulation in terms of authentication, entering symptoms, accessing medical records, or paying, and is there evidence from attack simulation tests indicating the user interaction integrity at each of those points? Where is the user entering or viewing sensitive data on interaction screens, and is there evidence from protection reports indicating the resistance of those screens to external capture during real-life interaction? Where are the critical points of healthcare interaction where a malicious overlay can deceive the user, and is there evidence from overlay test results of what the user experiences in such situations? Where are the critical points of interaction where the manipulated device can alter input and output for the user, and is there evidence from behavioral monitoring logs indicating the protection of user interaction in that untrusted environment?

Evidence in this layer is interaction-related, which refers to the logs of attack simulation at runtime for a particular interaction scenario, user-interface protection analysis for all sensitive interaction screens, attack overlay test results for the application response in

each interaction point tested, and the logs of runtime behavioral monitoring that prove interaction integrity even under an adversarial device environment.

3.4. Layer 4: Data Protection and Privacy

The assessment of data privacy and protection on PRISM-mH occurs not in terms of system architecture that needs to be evaluated, but rather in terms of the user's experience interacting with the m-health. The focus is how the user feels when his/her data is being processed, either being collected, shared, stored, or deleted, and whether or not the process is comprehensible for the user and under his/her control. This issue is especially important for m-health applications, in which users interact with sensitive personal information and make decisions about privacy issues.

In this layer, we ask the following questions: At which point of interaction prior to the data gathering process is the consent form displayed to the user, and is it formulated in a language that the average, non-technical healthcare consumer can comprehend, or is it simply dismissed without being comprehended? At which point of interaction within the application is the privacy policy displayed to the user, and is it done so within the application, or is the user taken away from the interaction context? When deleting the data or closing the account, what message is displayed to the user, and is there an interaction point where this is timestamped? At which interaction point is the user informed about the third-party entities receiving his/her data, and is this done in such a way that he/she actually gets a chance to approve/disapprove this action? Where do the prompts occur to review the privacy preferences, and do they appear during real use, or when the user seeks it?

The evidentiary basis at this layer, thus, is the timestamped consent log that indicates that the user has explicitly taken an action before any data collection, screenshots from the consent page and privacy policy pages as presented to the user during his/her interaction, the deletion audit log on the backend that confirms the deletion by the user at the time s/he requested it, and screenshots of third-party disclosures and privacy preference requests as shown to the user.

3.5. Layer 5: Authentication and Access Management

In PRISM-mH, authentication and access management are assessed not by whether these mechanisms exist but by how users experience them during real use. In m-health, authentication happens again whenever a user opens a medical record, pays for a service, changes credentials, returns to an expired session, or reaches a restricted feature. A technically sound mechanism may still fail to protect users if it appears without explanation, reads as a system error, or locks them out with no clear way back. For older adults, caregivers, and chronic patients, such failures can lead to abandoned tasks or unsafe workarounds such as sharing credentials.

This layer asks the following questions. At which interaction points, such as opening a medical record or starting a payment, is Multi-Factor Authentication (MFA) triggered, and how is the step explained to the user at that moment (AAM-UI-01)? When a session expires, does the message make it clear that the user must log in again rather than suggesting an error, and is unsaved data kept (AAM-UI-02)? When an account is locked, can a typical user follow the recovery guidance on their own, without contacting support (AAM-UI-03)? When users reach a restricted feature, is the access limit explained rather than shown as a generic error (AAM-UI-04)? During first use, is the benefit of MFA explained in non-technical language, and can users complete setup without help (AAM-UI-05)?

Accountability is shared. The security architect or manager establishes the policies of step-up authentication, sessions, and account lockout. The product owner decides how

MFA onboarding and role permissions are communicated. UX/UI developers write the messages that appear to users. The software developer implements the re-authentication process in a form that does not lose data. QA testers confirm the ability of representative users to perform the process.

3.6. Layer 6: Testing

Security testing in PRISM-mH is an interaction-based process. It not only determines whether tests have been carried out but more importantly, whether the tests were conducted on the interactions in which security vulnerabilities would impact the user. In the case of m-health applications, checking that encryption is implemented is not enough. It is necessary to test whether the interaction in which the user inputs her/his symptom details, logs in for a telemedicine appointment, or pays for services is secure without being intercepted, manipulated, or disrupted by an attacker during that interaction.

This layer asks the following questions: At which interaction points does user data travel between the app and backend services, and is there evidence from communication security test logs showing that data exchanged during those interactions cannot be intercepted or modified in transit? At which interaction screens containing sensitive healthcare data has screen capture protection been specifically tested, and is there evidence from protection reports confirming that those screens resist observation? At which healthcare interaction points has overlay attack testing been conducted, and what did the application do when an overlay was simulated at those moments? At which runtime interaction points has user behavior been tested against manipulation, and is there evidence from interaction manipulation logs showing what a user would experience if tampering were attempted at those points? Have authentication and session interaction points been verified throughout the full interaction lifecycle from login to logout? At which interaction points could a compromised device influence application behavior, and is there evidence from device compromise tests at those points? Are interaction-related security issues tracked from the interaction scenario in which they were identified to resolution?

3.7. Layer 7: Deployment and Distribution

In PRISM-mH, security deployment assessment is conducted from the interaction perspective of how the user will be experiencing and dealing with the m-health application once it gets downloaded, installed, and updated, and not just the technical aspects of the deployment process. The important issue is whether the current version of the application that the user will be working with is genuine, validated, and safe when he or she uses it.

This layer queries the following: What assurance is provided at the time the user installs or accesses the application to ensure that what the user is interacting with is genuine, and is there any documentation to support the process of verifying the integrity of the application before it reaches the user? At what interaction point is the user warned about their interaction with an outdated or insecure version of the software, and is there any indication that a process has been established for managing versions that notifies the user prior to allowing continued interaction with the application? At what interaction points during live operation can debugging interfaces be used to access sensitive data, and is there any verification that debugging functionality has been removed from the software before user interaction with the production version? How is the security of user–system interactions verified in each new version of the application before it is deployed in a release cycle? What are the channels the user initially uses to download the application, and how are those channels verified? How does the user verify the update is valid? What mechanism is used to track down the problem in deployment affecting user interaction?

3.8. Layer 8: Monitoring and Incident Response

Monitoring and incident response in PRISM-mH are assessed not from the perspective of what the security team observes in system logs but from the perspective of what the user encounters during their interaction with the application when something goes wrong. In m-health contexts, users have a right to know when their interaction environment is compromised, when their account has been accessed from an unrecognized device, or when a security incident has affected their health data. Also, users should have actionable guidance at the moment of that interaction, in language they can understand and act on without technical expertise.

This layer asks the following questions: At which interaction point does the user encounter a warning when their device is detected as compromised, and is there evidence that the alert is presented in plain language that the user can act on immediately? At which interaction point is the user alerted when a login is attempted from an unrecognized device? What choices are presented to the user at that moment, and is there evidence of the interaction flow through which the user can confirm, deny, or block the attempt? At which interaction point is the user informed of a security incident affecting their account, and is there evidence that the notification tells the user exactly what happened and what to do next in non-technical language? At which interaction point can users access their recent login and data access history in a readable and meaningful format? At which interaction point does the user encounter a warning when an unsafe network is detected, and is there evidence that the warning enables the user to make an informed security decision before continuing the interaction?

3.9. Layer 9: Compliance and Governance

When evaluating compliance and governance in PRISM-mH, these aspects are evaluated not through documentation processes but through attributes of the actual process of governing user–system interactions. Rather than asking whether the organization has a compliance document, the relevant questions include whether the interaction between users and health information occurs in compliance with regulations at the time of the interaction, whether the governance processes focus on the process of interaction, and whether the user is made aware of his/her data processing at the time of the interaction.

This layer asks the following: At which user–system interaction points do regulations such as GDPR, HIPAA, and ISO 27001 impose specific requirements, how has each obligation been mapped to the interaction scenario it governs, and is there evidence in a compliance checklist confirming that each regulated interaction has been assessed? At which interaction points involving healthcare data are governance controls applied, how are those interaction-level processes reviewed, and is there evidence of an interaction governance review record? At which user interaction points do third-party SDKs participate in handling user data—such as analytics SDKs active during medical record access—and is there evidence of a third-party assessment showing interaction-level security findings for each component? At which interaction points is personal data processed in ways that carry privacy risk, and is there a Data Protection Impact Assessment (DPIA) documenting risks at each such interaction point? Are compliance reviews conducted at defined intervals, and do they assess interaction-level practices? At which interaction points do users encounter transparency about their data? When incidents occur, are they documented in relation to the interaction point at which they originated? Are responsibilities for protecting each interaction point assigned and monitored at the governance level?

3.10. Layer 10: User Awareness and Usability

Awareness and usability of PRISM-mH are considered from the user's point of view regarding the user's experience at each security-related interaction point. For controls to secure a user's safety, the user needs to understand and interact with them at the exact moment that is necessary for their operation. In the case of WeCare, when the users are elderly patients, family caregivers, or healthcare providers who may not be highly technologically literate, every interaction with security features should be user-friendly.

This layer considers the following: At what interaction point in first use is the security onboarding experience presented; how well does it walk a user through security-related features using language and visuals that are appropriate for the healthcare consumer who is not a security expert, and is there any evidence that a first-time user could complete the security onboarding experience? At what interaction point is the required security action communicated to the user? Is the user proactively notified before any risk occurs rather than afterwards? And is there any indication that the notification is understandable to the user at this interaction point? At what interaction point are any of the security-related warning messages, security prompts, or security instructions presented, such as in the case of consent screens, MFA prompts, or session timeout messages? Have they been verified to be technically jargon-free and understandable to the healthcare consumer? If users realize how their personal and clinical data are being kept safe, they will use the application more enthusiastically. Where is the information about a security-related update received by the user at an interaction point, and is there any indication that the update message communicates the importance of the update to the user? Where can the users reach the help section within the application through interaction points? How many interaction points are involved in getting to it, and is there any indication that it helps the non-technical user?

4. The WeCare M-Health Application Case Study

The WeCare Application is a Jordanian pioneering home healthcare platform that functions as a digital bridge between individuals seeking professional medical care and qualified Healthcare Providers (HCPs) [51]. It was founded in 2017 by a group of healthcare professionals and technical engineers. It was developed to address the logistical and financial challenges of traditional clinical visits by bringing the hospital experience directly to the patient's doorstep [52].

WeCare supports not only patients but also HCPs by offering healthcare professionals a platform to manage their schedules, build their personal brands, and increase their income through autonomous practice [53]. Thus, WeCare targets these categories of audiences: care seekers—including families who seek care for elderly parents or children, chronic patients, or busy or independent professionals—and medical organizations such as hospitals and clinics that use the application's integrated ERP (Enterprise Resource Planning) system to manage their home-care business units.

The application offers a variety of services that facilitate interactions between users, the application system, and their devices, which can be summarized as follows:

- Professional medical care service through home visits, nursing services, and telemedicine.
- Specialized therapy and wellness services that involve physiotherapy, nutrition and dietetics, and maternal and childcare.
- Diagnostics and logistics services for lab sample collection, and a medical equipment marketplace to rent or purchase essential medical devices.
- Education services, where the application helps HCPs to develop their knowledge, practice, and training.
- Medical tourism services that assist in exploring international healthcare options with the assigned expert guidance.

- Clinic appointment booking for consultations with top specialists at WeCare.
- The digital wallet service, which offers a transparent payment system, where users can track their healthcare spending.
- Reporting and feedback services, which are requested at the end of the service and payment. This is critical for maintaining the platform's standard of care.

5. Results of Applying the PRISM-mH Framework Using the WeCare Case Study

The PRISM-mH framework is demonstrated using the WeCare m-health application. For the first time, the security interaction of the WeCare m-health application is evaluated across development phases with evidence. This itself is a novel contribution for the application. Table 7 shows the results of the study evaluation in terms of the layers, the primary roles responsible for creating evidence, security requirements, evidence type examined, completeness of evidence implementation, evidence accessibility, evidence availability, and reported gaps, if any. Table 7 is filled with assessment labels that have the following interpretation.

Table 7. Demonstrating the PRISM-mH framework using the WeCare case study.

Phase	Responsible Role	Security Requirement	Evidence Type Examined	Implementation	Evidence Accessibility	Evidence Availability	Gap
1—Business Modeling & Requirements	Business Analyst	BMR-UI-01—High-Risk User Interaction Workflows	Business case document, requirement specification with missing high-risk scenarios	Partial	Direct	Available	ACD
	Business Analyst	BMR-UI-02—Misuse Interaction	Misuse cases	No		Not available	ACD
	Business Analyst	BMR-UI-03—Interaction Documentation	User interaction workflow model	No		Not available	ACD
			Presentation of risky interaction points	No		Not available	ACD
	Business Analyst/Project Manager	BMR-UI-04—Compliance of Interactions	Compliance checklist, legal review report	Partial	Indirect	Available	ACD, AA
	Security Lead	BMR-UI-05—Interaction Data	Data classification matrix	No		Not available	ACD
	Project Manager	BMR-UI-06—User Roles in Interactions	Role assignment document	No		Not available	ACD
2—Design & Architecture Security	Solution Architect	DAS-UI-01—Interaction Data Flow	Data flow diagrams	Partial	Direct	Available	ACD
	Solution Architect	DAS-UI-02—Interaction Isolation	Design document	No		Not available	ACD
	Solution Architect/Security Expert	DAS-UI-03—Risky Interaction Adaptation	Data flow diagram, storage model	No		Not available	ACD
	Security Expert	DAS-UI-04—Interaction Security Protocols	Cryptographic architecture design	Partial	Direct	Not available	ACD

Table 7. *Cont.*

3—Development Security (Code Level)	Developer	DS-UI-01—Runtime Interactions	Runtime behavioral monitoring logs	No		Not available	ACD
	Developer	DS-UI-02—User-Interface Protection	User-interface protection reports	No		Not available	ACD
	Developer/Security Expert	DS-UI-03—Overlay Attack Testing	User-interface protection reports	No	Indirect	Not available	ACD
	Developer/Security Expert	DS-UI-04—Runtime Device Behavioral Monitoring	Runtime behavioral monitoring logs	No		Not available	ACD
4—Data Protection & Privacy	Developer/UI designer	DPP-UI-01—Informed Consent Presentation	UI consent screenshots Timestamp consent log pee session	No		Not available	ACD
	QA/DBA	DPP-UI-02—In-App Privacy Policy Accessibility	UI screenshot showing in-app privacy policy-Onboarding walkthrough confirming privacy link is visible	Yes	Direct	Available	
	Developer/UI designer	DPP-UI-03—User-Confirmed Data Deletion	UI screenshot of deletion confirmation message	Yes	Direct	Available	
	Developer/DBA	DPP-UI-04—Third-Party Data-Sharing Disclosure	Acknowledgment log recorded per sharing event	No		Not available	ACD, AA
	UX/UI Designer/QA Engineer	DPP-UI-05—Periodic Privacy Preference Review	Missing confirmation that preference update completes within the same session	No		Not available	ACD
5—Authentication & Access Management	Security Manager/UX/UI Designer/QA Engineer	AAM-UI-01—Context-Sensitive Multi-Factor Authentication	Config evidence	No		Not available	ACD
	QA Engineer/Developer	AAM-UI-02—Session Timeout & Re-Authentication	Usability test confirming users understand why they were logged out	Yes	Direct	Available	
	QA Engineer	AAM-UI-03—Account Lockout & Secure Recovery	Authentication design doc	Partial	Indirect	Not available	ACD
	Product Owner	AAM-UI-04—Role Transparency & Access Clarity	RBAC matrix	Partial	Indirect	Not available	ACD
	Product Owner/Security engineer	AAM-UI-05—MFA Onboarding Prompt	Usability test	Partial	Indirect	Not available	ACD
6—Testing & Validation	QA/Penetration Tester	TV-UI-01—Communication with Backend Services	Penetration testing reports	Partial	Indirect	Not available	ACD, AA
	QA/Security Tester	TV-UI-02—Interaction Screens	QA reports	No	Indirect	Not available	ACD
	QA/Security Tester	TV-UI-03—Malicious Overlays	Runtime interaction test logs	No		Not available	ACD
	Security Tester	TV-UI-04—User Interaction Behavior	Runtime monitoring reports	No		Not available	ACD
	QA/Security Lead	TV-UI-05—Authentication Testing	Authentication testing logs	Partial	Direct	Available	ACD

Table 7. *Cont.*

	Security Tester	TV-UI-06—Compromised Interactions	Emulator detection reports	No		Not available	ACD
	Project Manager	TV-UI-07—Interaction Tracking Review	Review meeting records, issue tracking reports	No		Not available	ACD
7—Deployment & Distribution	DevOps Engineer	DD-UI01—User Interaction with Verified App	Signing certificate, release process document from Google, Apple stores	Yes	Direct	Available	
	DevOps Engineer	DD-UI02—Protection of User Interaction from Vulnerable Versions	Application update policy, version control records	No		Not available	ACD
	Developer	DD-UI03—User Interaction Privacy at Runtime	Build settings	Yes	Indirect	Available	
	Security Officer	DD-UI04—Continuity of Trusted User Interaction Across Releases	Certificate management log	No		Not available	ACD
	DevOps/Release Manager/Security Lead	DD-UI05—User Interaction Trusted Channels	Screenshots or documentation showing the official store listing, store security validation reports, signing certificate details, and release management policy.	Yes	Direct	Available	
	Security Lead	DD-UI06—User Confidence in Interaction After Updates	Application update policy, version control	No		Not available	ACD
	Release Manager	DD-UI07—Tracking of Interaction-Affecting Deployment Issues	Missing release review records	No		Not available	ACD
8—Monitoring & Incident Response	QA Engineer	MIR-UI-01—Compromised Device Warning	QA reports, interaction log, UI screenshots	No		Not available	ACD
	Security expert	MIR-UI-02—Suspicious Login Alert	Breach notification message, IR drill report	No		Not available	ACD
	QA Engineer	MIR-UI-03—Security Incident Notification	UI screenshots of user-accessible activity log, test log confirming session history accuracy	No		Not available	ACD
	QA Engineer/UI/UX designer	MIR-UI-04—User-Accessible Activity Log	UI screenshot of network warning, Interaction log	No		Not available	ACD
	Network Security	MIR-UI-05—Unsafe Network User Decision Point	UI screenshot of the unsafe network alert, network monitoring report, interaction log	No		Not available	ACD

Table 7. *Cont.*

9—Compliance & Governance	Project Manager	CG-UI01—Healthcare Regulatory Compliance	Compliance audit report	No		Not available	ACD, AA
	Security Expert	CG-UI02—Interaction Governance Review	Evidence repository	Partial	Indirect	Available	AC
	Security Lead	CG-UI03—Third-Party SDK Assessment	Vendor risk report	No		Not available	ACD
	Security Expert	CG-UI04—Data Protection Impact Assessment	DPIA report	No		Not available	ACD, AA
	Project Manager/Domain Expert	CG-UI05—Regular Compliance Reviews	Review calendar	No		Not available	ACD, AA
	Compliance Officer	CG-UI06—User Data Transparency	User consent records/forms	No		Not available	ACD, AA
	Security Expert/Project Manager	CG-UI07—Incident Documentation & Review	Incident review records	No		Not available	ACD
	Security Expert	CG-UI08—Responsibility Assignment	Role assignment documentation	No		Not available	ACD, AA
10—User Awareness & Usability	UX Designer/Security expert	UAU-UI-01—Security Onboarding Walkthrough	User manual, awareness campaign content	Partial	Indirect	Available	AC
	Developer/UX Designer/Security Officer	UAU-UI-02—Proactive Security Notifications	UI screenshot, usability test, test log confirming notifications appear at correct interaction point	Yes	Direct	Available	
	UX Designer/Product owner	UAU-UI-03—Plain-Language Security Messaging	UI text, user guide	Yes	Direct	Available	
	UX Designer/Security Officer	UAU-UI-04—User-Driven Security Update Action	UI screenshots, Help content	No		Not available	ACD
	Communications Team	UAU-UI-05—In-Application Help & Awareness Resource	Content review, user manual, help content.	Partial	Direct	Available	AC

The labels below are for the Required Evidence Implementation column. It is necessary to explain the meaning of the labels, as a missing document is not the same as a missing control. A missing control is not the same as an item that is not applicable.

Yes. The requirement is implemented, and sufficient evidence is provided.

Partial. The requirement is partly implemented, inconsistently implemented, or supported by incomplete evidence.

No—control absent. The control or practice is not implemented. This means the requirement is applicable but not satisfied based on the available evidence. This likely indicates an actual weakness.

Missing evidence. The assessor could not judge the requirement because the necessary documentation, logs, screenshots, test reports, or access were not available. A "Missing evidence" result indicates that the required interaction-level evidence was not available for

review. It denotes an assurance and documentation gap rather than a confirmed security weakness. Absent evidence should not be read as a vulnerability.

Not applicable. The requirement does not apply or is not related to this case, with the reason briefly described.

The labels below are for the Evidence Accessibility column:

Direct access means that the assessor examined primary artifacts first-hand—such as by reviewing documentation and/or source code repositories, inspecting live authentication configuration files, or navigating the deployed application interface—without intermediary interpretation.

Indirect access means that the assessor relied on documentation or artifacts prepared by a third party, such as pre-compiled QA reports, compliance summaries, or written descriptions of system behavior provided by the development team, where independent verification of the underlying system state was not possible. Also, it indicates that the assessor relied on existing documentation or artifacts to indicate the status of the evidence.

Blank denotes not applicable due to unavailability of evidence to assess.

For the Evidence Availability column, two labels are considered: available means the evidence is available whether the control is implemented or not, and not available means that evidence is not available, also regardless of control implementation.

Regarding the Gap column in the framework, the following labels are distinguished:

AC means the reported gap is due to the absence of control implementation.

AD means the reported gap is due to the absence of documentation or materials such as design, source code, etc.

ACD means the reported gap is due to the absence of control implementation and materials together.

AA means the reported gap is due to the absence of an assessor.

Blank means no gap is identified here.

5.1. Results of Layer 1: Business Modeling and Requirements in WeCare

The assessment of WeCare at this layer revealed significant gaps in how the application models and documents user–system interaction scenarios at the requirements stage. When assessed against the interaction-oriented questions in PRISM-mH Layer 1, the high-risk workflows were absent from the formal requirements. Critical interactions such as logging into the patient account, secure data collection during the first time launching the application based on consent, the creation of the telemedicine session by both the patient and the healthcare professional, and payment authorization using the digital wallet were not considered at any particular point in the business case and requirement specification. This document specified security generally without indicating at what point in the interaction process the user is vulnerable.

Misuse interaction instances were completely absent, as no document shows how a user or attacker could use any particular interaction point, such as the HCP credential management interaction, lab sample collection authorization interaction, and third-party payment interaction. A trust boundary diagram and user interaction workflow diagram for interaction points between patient and system were not present in the requirements. Hence, this indicates the absence of a formal requirement specification for high-risk interactions where users are more vulnerable. Compliance requirements were mentioned in the documents but not linked to interaction points. Hence, it could not be determined in the current document which regulation applies to which interaction point, such as GDPR at the consent interaction during the data-sharing interaction with a laboratory services third party.

There is no classification of data sensitivity in relation to interactions between users and the system. There is no matrix that maps the data sensitivity levels to the exact interactions where sensitive data, like diagnostic data, payment information, or credentials of HCPs, is collected, processed, or shared. There is no document or RACI chart confirming the accountability of security-related decisions regarding the interactions in question. These gaps at the requirement stage create downstream risks for all subsequent layers, as the interaction-level security models needed to inform architectural, development, and testing decisions were never established.

5.2. Results of Layer 2: Design and Architecture Security in WeCare

WeCare's interaction architecture was not evaluated based on infrastructure security but on its capacity to ensure the continuity of secure interactions among users, devices, back-end services, and third parties—within the scope of home visit scheduling, telemedicine consultations, laboratory sample delivery and handling, and digital wallet payments.

The findings show that WeCare does not have the models and documentation required to prove that interaction-level security is an intentional architectural decision of the system. Some interaction data flow diagrams were incomplete and did not identify any potential interaction risks and thus did not allow for understanding how security keys are passed from the patient's device to the backend service during authentication or how session tokens are handled in the telemedicine consultation handover process. Also, there were no secure session models, user-context transition diagrams, or communication trust sequence diagrams to understand whether there were architectural decisions about users' security at the moments when their roles or contexts change (e.g., during the transition from home visits to payments).

The cryptographic architecture documentation was incomplete. There were indications that some cryptography was designed, but no architectural documentation showed how each protocol corresponds to a particular interaction scenario in which it is chosen to provide protection. Therefore, it could not be verified whether the architecture adapted properly to the varying risky interaction scenarios. Without such interaction-level architectural models, the security architecture of WeCare cannot be evaluated in light of the interaction scenarios it is meant to secure.

5.3. Results of Layer 3: Development Security (Code Level) in WeCare

The development assessment of WeCare revealed several weaknesses in its ability to secure its users while they interact with it during runtime. When evaluated against the questions of this layer from an interaction point of view, it became clear that WeCare lacks security for user interactions at the code level.

In the process of user authentication and medical data entry at the interface interaction point, WeCare was found to have inadequate anti-tampering measures and monitoring of user-interface interactions. This implies that the user's input during login, symptom reporting, or telemedical consultation could have been silently tampered with at the point of interaction. The behavioral runtime logs were missing. This indicates that there are no current tests for the interaction's integrity at these important interfaces. At the points of interaction, where patients are viewing or entering their most sensitive health data—for instance, displaying the results of diagnosis or entering prescriptions—the screenshot prevention feature was inadequately implemented, while keylogging protection measures were partly implemented. This implies that at these interaction points, user activities can be monitored by a third party without the user's knowledge.

In addition, overlay attack protection has been identified as incomplete. The login page, payment screen, and prescription interaction screen—the places where the user is

most likely to be deceived into providing their credentials and health information using a fake interface—overlay detection was not proven to be implemented completely. There was no information on any tests of how the user experience would be affected if an overlay attack was detected during an interaction with the application during treatment. Device compromise protection at WeCare does not include root detection, jailbreak detection, or emulator detection. In other words, when the user interacts with the WeCare application from a compromised device, WeCare cannot detect the compromised state of the device and cannot protect the user experience.

5.4. Results of Layer 4: Data Protection and Privacy in WeCare

Based on the identified questions of this layer, and by analyzing the data protection and privacy of the WeCare application, this layer revealed a significant gap in relation to the consent interaction question since WeCare does not provide users with an informed consent screen before data collection starts. This means that when the user starts interacting with the application for the first time and provides some personal health data, such as his/her name, medical records, or location for visit planning, there is no consent screen that explains the data collection by WeCare, nor a timestamp for the consent record.

The first positive finding is that both the backend and the mobile application permit the user to view the privacy policy in the application. WeCare's onboarding tour also verified that the privacy link is visible in the application. The second positive finding is that a deletion confirmation message is displayed during the process of closing an account, and there is proof of this interaction via a UI screenshot. The third finding is negative, as the interaction of disclosing data to third parties is absent in the application. There is no evidence that the user is informed regarding who will receive his/her data, including lab service providers, medical equipment providers, and payment processing companies integrating with WeCare, before the actual data-sharing interaction takes place. Also, there is no acknowledgment log confirming that the user sees that information and has a chance to accept or reject the sharing of his/her information at this particular interaction point. Another missing interaction point is the periodic privacy preference review prompt. There is no interaction point where the user is prompted to review or update his/her data-sharing preferences during regular use of the application. These gaps leave WeCare users uninformed about key aspects of how their health data is handled at the moments when those decisions are most consequential.

5.5. Results of Layer 5: Authentication and Access Management in WeCare

The assessment of authentication and access control of WeCare revealed some positive outcomes in certain interaction scenarios but also identified quite a few vulnerabilities. When compared to the interaction-oriented questions raised in this layer, the most prominent vulnerability identified was the lack of MFA at the moments of interaction with a medical record or payments. Namely, WeCare does not employ MFA when a user accesses sensitive healthcare information or makes a payment from the digital wallet. In other words, in scenarios when security is critical (e.g., when a patient views his/her diagnosis, when a doctor sends a prescription, or when a user makes a payment for a house visit), MFA is not applied, but only one factor is used instead.

WeCare performed well in terms of user session timeout interactions, as users are signed out after an idle period of time and are provided with a prompt and comprehensible message about signing in again. The study revealed that the user understands the reason they were signed out and how to sign back in, and this is an important finding from an interaction perspective. WeCare handles the account lockout problem partially in the process of authentication. However, the recovery process provided to the user in case of a

lockout interaction is not documented, and it is not clear whether the user will be able to recover the account on their own.

RBAC is partially implemented in WeCare, yet the communication of access boundaries at the interaction level appeared to be poor. It appears that users may find themselves accessing restricted zones of the application without any clear understanding of why access is limited and what specific permission of RBAC prevents them from doing this. It causes confusion at the interaction point, rather than giving directions to the user's interaction boundaries in the application. The MFA onboarding step during the first interaction was partially addressed, as users were given some information about security features, yet the interaction did not clearly communicate the significance of activating MFA in non-technical language for average healthcare users.

5.6. Results of Layer 6: Testing and Validation in WeCare

The testing and validation assessment of WeCare revealed that security testing was conducted to a limited extent and was not systematically targeted at the specific interaction points where users are most exposed during real use. When assessed against the interaction-oriented questions of this layer, the gaps were significant across nearly all interaction-focused test categories.

Penetration testing was partially conducted and provided some evidence regarding backend communication security. However, confirmation of Man-In-The-Middle (MITM) protection during live user interactions, such as telemedicine sessions or payment authorizations, was insufficient. It could not be confirmed that the data exchanged at those specific interaction points is protected from interception in real-world network conditions. Interaction screens were not included in QA test reports, including the specific screens where users enter medical data, view diagnostic results, or authorize payments. No documentation was found confirming that those screens had been tested for external capture or monitoring vulnerability. Testing for malicious overlay attacks at critical interaction points, such as the login screen, prescription entry, and payment confirmation, was absent, making it unverifiable whether WeCare can detect and warn the user when an overlay is simulated at those healthcare interaction points.

Runtime manipulation testing was not conducted, and no interaction manipulation test logs were available. This includes the absence of assessing whether user behavior at specific interaction points, such as form submission and session handoff, could be tampered with. Compromised device testing at the interaction level was similarly absent. No emulator detection tests or root/jailbreak detection tests were conducted to verify how the application behaves at user interaction points when the device environment is untrusted. Authentication testing was partially covered for session-related interactions, but the coverage across the full authentication interaction lifecycle was incomplete. Finally, no structured tracking system was employed to record interaction-related security issues from detection to resolution, meaning that vulnerabilities identified during testing cannot be consistently linked to the specific interaction scenarios they affect or confirmed as resolved at the interaction level.

5.7. Results of Layer 7: Deployment and Distribution in WeCare

The findings of the WeCare deployment demonstrate a mixture of strengths and weaknesses. Some aspects of WeCare's deployment indeed protect the interaction between the user and the application, but some other deployment methods do not guarantee the necessary protection of the interaction process. When WeCare is assessed using the interaction-oriented questions of this layer, the application employed trustworthy distribution channels. WeCare can be installed only from the Google Play Store and the Apple App

Store, which have security measures allowing users to make sure that when downloading the application for the first time, they start the interaction process with a certified copy of it.

The debug flags were turned off before the release of WeCare into production, indicating that at the points of interaction, sensitive information such as session tokens or patient records may potentially become visible using diagnostic tools. This risk was addressed and documented by the build settings evidence. However, in the case of the WeCare update interaction, there is no evidence provided for the verification of the update source. Certificate management logs were absent, and no documentation described how signing certificates are managed across deployment cycles, making it impossible to confirm that the interaction between the user and the updated application is consistently protected from tampered updates.

Version management was identified as a significant gap. At the interaction point, when a user runs an outdated or vulnerable version of WeCare, no mechanism was found to inform the user of the risk or guide them to update before continuing their interaction with the application. Similarly, no evidence of interaction-level continuity across releases was available: there was no documentation showing that each release cycle included an interaction-focused security review confirming that security at user interaction points was maintained across deployments. Deployment issues that affect user interactions were not formally tracked, and no deployment issue tracking report was available to confirm that interaction-affecting problems were identified, assigned, and resolved before the affected version continued to be used.

5.8. Results of Layer 8: Monitoring and Incident Response in WeCare

The Monitoring and Incident Response assessment of WeCare revealed the complete absence of user-facing security communication across all five interaction-oriented assessment areas. When assessed against the interaction-oriented questions of this layer, WeCare was found to leave its users entirely uninformed and unable to act at each of the interaction points where their security decisions are most consequential.

At the device compromise interaction point, WeCare does not detect or respond to a compromised device environment. When a user interacts with WeCare on a rooted or jailbroken device—a situation that is not uncommon among non-technical healthcare users who may not be aware of the risks—no warning appears at any interaction point. There is no UI change, no alert, and no guidance to help the user make an informed decision about whether to continue. At the suspicious login interaction point, WeCare does not alert the user when their account is accessed from an unrecognized device. No notification, push message, or in-application alert allows the user an opportunity to confirm, deny, or block the login attempt. This means that if a patient's WeCare account is accessed from an unknown location while they are using the application for a telemedicine consultation, neither the patient nor the application responds to the event at the interaction level.

The security incident notification interaction was similarly absent. No mechanism was found to inform a user when a security incident has affected their account, and no push notification template or in-application alert evidence was available. The user cannot interact with WeCare's activity log at the activity log interaction point, so they have no option to look at their recent logins and data access activity history on the application. A user suspecting unauthorized access to their account cannot engage with any interaction point within WeCare where they may check their session activity history. Finally, there is no warning about the insecure connection shown at any interaction point within the user session. If a user connects to unsecured Wi-Fi while booking a home visit or engaging in a telemedicine session using WeCare, there is no warning, thus leaving the user uninformed about the security risks of their actions.

5.9. Results of Layer 9: Compliance and Governance in WeCare

The assessment of WeCare in this layer revealed several weaknesses in how compliance and governance practices support secure user interactions within the application. Although some evidence was archived, limited evidence was available to indicate how the governance processes were effectively used to monitor user interactions with confidential health information while using the system. In terms of regulation, there was insufficient evidence to show that interactions between users and their healthcare information met regulations such as GDPR, HIPAA, or ISO regulations. Some security practices were not adequately followed to create a report or conduct an audit. Accordingly, there is high uncertainty of whether healthcare information being used in user interactions is always managed according to regulations. There was no clear evidence demonstrating that security reviews of interaction flows, access practices, or operational data handling were regularly conducted. Moreover, third-party software development kits (SDKs) and external services were not analyzed in the framework of vendor risk assessments, making it unclear what their impact is on interactions and data protection at runtime.

A major concern relates to the absence of a privacy impact assessment. It seemed unclear whether there had been any privacy impact assessment done to ascertain the means by which users' data will be collected, stored, and distributed throughout the process of interaction with the application. Moreover, it was clear that users did not have sufficient knowledge about the way their data would be managed during the use of the application. Hence, this reflects the absence of any predefined schedule for conducting governance activities and addressing any related challenges.

5.10. Results of Layer 10: User Awareness and Usability in WeCare

WeCare's user awareness and usability evaluation at this level was the most balanced one of all ten layers, with some real strengths in various interactions and some obvious weaknesses too. WeCare revealed considerable attention to the user experience in several security interactions, especially in communication-related ones.

Whenever there is a need for security measures, such as changing passwords, the application notifies the user directly through a notification presented at the right interaction point before the user gets blocked or faces any kind of risk. The screenshot, usability test, and test logs verified that the notifications are presented at the right interaction point and are understandable to the user. WeCare revealed positive findings in the security messaging language of prompts and instructions found when the user interacts with the application. They had been written using understandable English and avoided the use of any technical jargon.

However, the security onboarding walkthrough on first usage was only partially complete. New users are given some sort of onboarding guidance, but it does not entirely include all the security-related interactions a user would undergo, such as how to react to a security warning during a telemedicine visit or how to set up the privacy settings before transmitting health information. This implies that users might start using some of the sensitive functionalities related to healthcare without fully comprehending the security mechanisms at their disposal and how to use them. The second point where WeCare had a large security gap was the security update notification interaction point, since there is no security update notification explaining why the update is important to users' health data.

6. Discussion

PRISM-mH is designed to be applicable both during the development process and as a structured checklist reference for reviewing security practices at the post-development stage in m-health applications. Rather than limiting the security focus to technical con-

trols, the framework is structured to examine how security considerations are addressed progressively across the development lifecycle—from planning and design to deployment and operational monitoring. The PRISM-mH framework is conceptual and not technical. Existing security assessment models evaluate m-health systems for a compliance destination by validating all their components after their implementation is completed. However, PRISM-mH suggests that m-health environments are highly dynamic and interactive, and, therefore, security in those environments cannot be evaluated retrospectively. This change in perspective from viewing security as a static destination to seeing it as an integral part of the development process represents the conceptual novelty of the model. PRISM-mH's structure is composed of three pillars for a reason: linking each security requirement to a phase in the m-health development process, an accountable role, and a piece of evidence serves as an attempt to fill the existing gap in the literature. Current security assessment models specify security in general and fail to describe how and when security requirements are enacted. Instead, PRISM-mH acknowledges that these considerations should be viewed as interdependent.

What makes PRISM-mH unique from other current security assessment frameworks is its consistent use of the interaction approach in all ten layers. Instead of analyzing the security issues based on the infrastructure or the backend side of things, PRISM-mH seeks to understand what impacts security at the interface level—what the user sees, what the user is supposed to do, and whether the risk is communicated sufficiently for users to react. The adoption of an interaction-oriented assessment approach is particularly well-suited to m-health applications, given that such systems inherently require patients to engage actively with the interface, rather than passively receiving a service, in order to manage their own health information. For example, the lack of a consent form before data acquisition is not only an indicator of a lack of compliance with laws and regulations but also a gap at the interface level, because the possibility for the patient to exercise informed control over their data is not adequately enabled through the interface.

The WeCare case study gives the framework its empirical grounding, and the results are instructive in several ways. Across ten layers, WeCare performed better at the development and deployment levels. Runtime protections, session management, and official store distribution all showed partial compliance. WeCare falls short in the layers that require sustained organizational commitment rather than a one-time technical effort, such as Monitoring, Compliance and Governance, Data Protection, and User Awareness. The drawbacks in WeCare's monitoring layer cannot be resolved by the developers themselves. The absence of alerts for potential security breaches, such as compromised devices, suspicious logins, user logs, and potentially malicious networks, indicates that there has been no well-defined process for how these security threats should be communicated to the end-users. This is not a single-role responsibility in WeCare, and no data shows that the issue has been addressed before. PRISM-mH made this more apparent than a regular audit would have.

Similarly, in the Compliance and Governance layer, it became evident that WeCare not only failed in terms of documenting the data governance process but also failed in terms of building the organizational infrastructure necessary to do so, as there was no DPIA, no scheduled reviews, and no assessment of vendor risks. The role-driven nature of the framework used provided the assessment with the opportunity to establish not only where there were gaps but also who was responsible for addressing them.

The findings provide an empirical basis for positioning PRISM-mH in the socio-technical context, which implies that the integration of user-interface, technical control, and governance factors within the framework can be useful in evaluating actual m-health systems. Security in m-health systems is not simply a technical problem that good code

solves. It is a governance problem, a communication problem, a design problem, and a role accountability problem simultaneously. Such frameworks that consider only one of these dimensions may not provide enough analysis of the other ones, especially in cases when the security results are based on the interaction of behavior, technology, and governance. One of the main goals in developing PRISM-mH was the consideration of these dimensions within a single evaluation structure in a comprehensive way. The PRISM-mH framework is based on interdisciplinary coordination in m-health environments, where the involved roles are software engineers, network engineers, compliance officers, healthcare professionals, designers, privacy officers, security experts, and managers. Thus, PRISM-mH is designed to support closer alignment between security technical experts and governance bodies by making it explicit that security constitutes a shared responsibility.

A further intended contribution of PRISM-mH is that it is designed to function as an assessment tool specified for guiding the improvement of m-health security interactions over time. The framework can be applied during the development of an m-health application or at the post-deployment stage, as illustrated in the WeCare case study. This dual applicability positions PRISM-mH as a potential continuous improvement artifact that may be employed periodically and incrementally to track security assessment progress across the development lifecycle, from early requirement specification to operational monitoring. Therefore, it offers a structured basis for stakeholders to identify gaps and make evidence-informed adjustments as the application evolves and user interactions grow in complexity. In this sense, PRISM-mH may serve as both a diagnostic reference and a strategic planning aid for improving m-health security practices and interactions.

The PRISM-mH framework has certain limitations. Firstly, it relies heavily on the availability of formal documentation as evidence. In healthcare environments where rapid delivery of medical software increments and formal documentation are essential, the agility of delivery often remains a challenge. Second, the lack of quantitative assessment scores also limits maturity measurement and progress for benchmarking potential. Thus, quantitative scoring mechanisms are stated in the plans. Third, a single case study of evaluation, while informative at this stage of the research, does not allow generalization, and further multi-case and comparative evaluations are noted as future directions. These are directions for future development, including AI-supported gap prediction [54–58] and multi-application evaluations that would expand both the rigor and reach of the framework [59].

An analytical comparison of PRISM-mH with OWASP MASVS and OWASP SAMM is presented in Section 7. However, the effectiveness of PRISM-mH has not yet been compared empirically with existing frameworks by applying them to the same applications. The author therefore plans a side-by-side empirical comparison of PRISM-mH and established security assessment models in m-health environments.

7. Comparative Analysis of PRISM-mH and Existing Security Assessment Models

This section compares PRISM-mH with an established security assessment model that has been used to evaluate m-health applications, to show where the two converge and where PRISM-mH departs from existing practice. The OWASP Mobile Application Security Verification Standard (MASVS), version 2.1 [60], was selected as the primary comparator, together with its companion Mobile Application Security Testing Guide (MASTG) [42]. MASVS was chosen for three reasons. First, it is the most widely adopted standard for verifying the security of mobile applications and has been used in empirical security evaluations of m-health applications [61]. Second, its structure is publicly documented and stable, which allows a transparent, component-by-component comparison. Third, MASTG was one of the knowledge sources used in designing PRISM-mH (Section 2.3); thus, the

comparison also shows what PRISM-mH contributes beyond one of its own foundations. Because MASVS is a verification standard rather than a lifecycle model, the OWASP Software Assurance Maturity Model (SAMM) v2.0 [45] is included as a secondary reference point. SAMM is lifecycle-based, which makes it possible to separate the contribution of lifecycle coverage from the other distinguishing features of PRISM-mH.

The comparison was conducted as a structured, criteria-based analytical comparison of the published structure and content of each model. Nine criteria were derived from the objectives of PRISM-mH (Section 2.2) and from the gaps identified in the Introduction.

- (C1) The unit of assessment;
- (C2) Lifecycle coverage;
- (C3) The assignment of accountable roles;
- (C4) The evidence required to support a judgment;
- (C5) The orientation of the assessment questions (control presence versus user–system interaction);
- (C6) Coverage of organizational, governance, and compliance aspects;
- (C7) Specificity to the healthcare context;
- (C8) The form of the assessment output;
- (C9) The point in time at which the model is applied.

The comparison was carried out at two levels. At the structural level, the three models were compared against the nine criteria, as shown in Table 8. At the component level, each of the eight MASVS control groups was mapped to the PRISM-mH layers that address the same security concern, and the additional elements that PRISM-mH introduces for that concern are identified in Table 9.

Table 8. Structural comparison of OWASP MASVS v2.1, OWASP SAMM v2.0, and PRISM-mH.

Criterion	OWASP MASVS v2.1 (with MASTG)	OWASP SAMM v2.0	PRISM-mH
C1. Unit of assessment	The mobile application (client side and its communication with endpoints).	The organization's software assurance program.	The m-health application together with the development process and the roles that produce it.
C2. Lifecycle coverage	Not structured by lifecycle. Applied mainly to a built application.	Structured by business functions spanning Governance, Design, Implementation, Verification, and Operations.	Ten layers aligned with lifecycle phases, from Business Modeling and Requirements to Monitoring, Governance, and User Awareness.
C3. Accountable roles	Not specified. Verification is performed by a security tester.	Practices are owned by the organization. Individual roles are not assigned per requirement.	Every requirement is linked to an accountable role (e.g., business analyst, architect, developer, QA engineer, DevOps, compliance officer).
C4. Evidence required	Test results from static and dynamic analysis (MASTG test cases).	Questionnaire-based assessment of practices. Evidence is not prescribed per requirement.	A defined evidence type for every requirement (e.g., misuse-case models, interaction data flow diagrams, timestamped consent logs, overlay test results, RACI charts, DPIA), with direct or indirect access recorded.

Table 8. *Cont.*

Criterion	OWASP MASVS v2.1 (with MASTG)	OWASP SAMM v2.0	PRISM-mH
C5. Question orientation	Control presence and correctness ("Does the app securely store sensitive data?").	Practice maturity ("Is there a defined process for. . .?").	User–system interaction ("At which interaction point does the user encounter. . ., and what evidence confirms it?").
C6. Organizational, governance, and compliance coverage	Outside scope.	Covered (Governance function).	Covered, and tied to specific interaction points (Layers 1 and 9).
C7. Healthcare specificity	Generic to all mobile applications. Widely adopted for evaluating m-health apps.	Generic to all software.	Designed for m-health workflows (e.g., symptom entry, telemedicine sessions, prescriptions, payments for care) and regulations (e.g., HIPAA, GDPR).
C8. Output	Pass/fail per control or test case, under MAS testing profiles.	Maturity score (levels 0–3) per practice.	Per-requirement assessment label (yes, partial, no, missing evidence, not applicable), evidence availability and access, and a gap type (AC, AD, ACD, AA).
C9. When applied	Mainly before release or at audit.	Periodically, at the program level.	Progressively during development, or retrospectively after deployment (as in WeCare).

Table 9. Mapping of OWASP MASVS v2.1 control groups to PRISM-mH layers.

MASVS Control Group (Focus)	Corresponding PRISM-mH Layer(s)	What PRISM-mH Adds
MASVS-STORAGE (secure storage, prevention of data leakage)	L4 Data Protection and Privacy, L3 Security Development	Identifies the interaction points at which sensitive data is entered, viewed, or cached; assigns the developer and data owner; and requires evidence such as a data classification matrix linked to interaction points
MASVS-CRYPTO (strong cryptography, key management)	L2 Design and Architecture Security	Asks at which interaction points keys are exchanged and requires interaction data flow diagrams showing that key management follows the interaction process, and assigns the architect
MASVS-AUTH (authentication, local authentication, step-up for sensitive operations)	L5 Authentication and Access Management, L6 Testing	Examines how authentication is experienced across the full session, from login to logout and session handoff, and requires evidence that sensitive health actions (e.g., viewing records, paying) trigger appropriate re-authentication
MASVS-NETWORK (secure traffic, identity pinning)	L2 Design and Architecture Security, L6 Testing	Links protection in transit to specific risky interaction contexts (e.g., telemedicine over an untrusted network), and requires communication trust sequence diagrams and communication security test logs per interaction

Table 9. *Cont.*

MASVS Control Group (Focus)	Corresponding PRISM-mH Layer(s)	What PRISM-mH Adds
MASVS-PLATFORM (IPC, WebViews, secure user interface)	L3 Security Development, L6 Testing	Extends user interface security to screen capture protection on sensitive health screens and overlay attack testing at named healthcare interaction points, with test reports as evidence
MASVS-CODE (up-to-date platform, enforced updates, components without known vulnerabilities, input validation)	L3 Security Development, L7 Deployment and Distribution	Asks how the user is warned about outdated or insecure versions and how update authenticity is verified at installation, and assigns DevOps and release management roles
MASVS-RESILIENCE (platform integrity, anti-tampering, anti-static and anti-dynamic analysis)	L3 Security Development, L8 Monitoring and Incident Response	Goes beyond detecting a compromised device to ask whether the user is warned in plain language, what choices the user is given, and who is accountable for the response
MASVS-PRIVACY (data minimization, unlinkability, transparency, user control)	L4 Data Protection and Privacy, L9 Compliance and Governance, L10 User Awareness and Usability	Specifies where consent and privacy information appear in the user journey and requires timestamped consent and deletion logs, screenshots of the screens users see, third-party SDK assessments, and a DPIA

7.1. Structural Comparison

Table 8 summarizes the structural comparison. MASVS is organized into eight control groups: STORAGE, CRYPTO, AUTH, NETWORK, PLATFORM, CODE, RESILIENCE, and PRIVACY. Each of the control groups contains a small number of high-level controls. Its unit of assessment is the mobile application itself, and verification is carried out mainly by security testers through static and dynamic analysis guided by MASTG test cases [42,60]. In version 2, the architecture, design, and threat-modeling requirements of earlier versions were removed so that the standard concentrates on controls that can be verified in the application. MASVS therefore does not address when in the development lifecycle a control is introduced, who in the organization is responsible for it, or how it is experienced by the user. SAMM, by contrast, is organized around five business functions: Governance, Design, Implementation, Verification, and Operations, comprising fifteen security practices assessed across three maturity levels [45]. SAMM covers the lifecycle and organizational practices, but its unit of assessment is the organization's software assurance program as a whole, and it is generic rather than specific to mobile or healthcare contexts.

PRISM-mH combines features that neither model provides on its own. Like SAMM, it follows the development lifecycle. Like MASVS, it reaches down to concrete, application-level security requirements. In addition, it attaches to every requirement an accountable role, a defined type of evidence, and an interaction-centered assessment question.

7.2. Component-Level Comparison

Table 9 maps each MASVS control group to PRISM-mH. Every MASVS control group has a counterpart in at least one PRISM-mH layer, so the technical scope of MASVS is retained within PRISM-mH. The difference lies in how each concern is assessed. Where MASVS asks whether a control is present and correctly implemented, PRISM-mH asks at which interaction point the control protects the user, which role is accountable for it, and which evidence demonstrates it.

The reverse mapping is equally informative. Four areas of PRISM-mH have no direct counterpart in MASVS:

- Layer 1 (Business Modeling and Requirements). It requires high-risk interaction workflows, misuse cases, and regulatory obligations to be modeled before any code is written.
- Layer 8 (Monitoring and Incident Response). Its operational and user-facing dimensions, such as incident notification, login-history visibility, and alerts about unrecognized devices, are outside MASVS.
- Layer 9 (Compliance and Governance). It covers DPIA, scheduled compliance reviews, and third-party vendor risk assessment.
- Layer 10 (User Awareness and Usability). It assesses whether security information is understandable and actionable for non-technical patients and caregivers.

In addition, Layer 6 (Testing) is partly supported by MASTG test cases. PRISM-mH, however, requires that tests be traced to the specific interactions at which a vulnerability would affect the user. Several of these areas are addressed in generic form by SAMM's Governance and Operations functions, but SAMM does not link them to user–system interaction points or to the m-health context.

This comparison shows that the novelty of PRISM-mH does not lie in the individual security controls it examines, as many of these are shared with MASVS and other established sources, nor does it lie in lifecycle coverage alone, which SAMM also provides. The novelty of PRISM-mH lies in the combination of four features within a single assessment structure: (1) interaction-centered assessment questions that ask where and how a control protects the user, rather than whether it exists; (2) explicit accountable roles for every requirement; (3) a prescribed type of evidence for every requirement, with transparent recording of evidence availability and access; and (4) lifecycle-wide coverage that extends from requirements to governance and user awareness and is specific to the m-health context.

PRISM-mH is therefore complementary to MASVS rather than a replacement for it. MASVS and MASTG remain the appropriate instruments for in-depth technical verification. Their test results can serve directly as evidence within PRISM-mH Layers 3, 5, and 6, while PRISM-mH places those results within the wider interaction, role, and governance context of the m-health application.

This comparison is analytical. It is based on the published structure and content of the models and on the WeCare findings. MASTG and MASVS were not applied to WeCare in parallel. It therefore shows differences in scope and design, not differences in assessment performance. An empirical side-by-side application of PRISM-mH and MASVS to the same m-health applications, including a comparison of the gaps each approach detects, is planned as part of future work.

8. Conclusions

The PRISM-mH framework was proposed in this research as an interaction-centric framework for assessing security in m-health applications. A comparative analysis with OWASP MASVS, a standard widely used to evaluate the security of m-health applications, showed that PRISM-mH retains the technical scope of such standards. It extends that scope through interaction-centered assessment questions, accountable roles, prescribed evidence, and coverage of business modeling, monitoring, governance, and user awareness that control-oriented standards do not address. The link between each security requirement and its development phase, responsible stakeholder, and evidence is used to address the issues associated with the existing approaches to evaluating security in interactive m-health systems—changing the paradigm from traditional security auditing to ongoing and embedded lifecycle security evaluation. As such, the case study of WeCare provides preliminary indications of the feasibility and applicability of PRISM-mH, suggesting that, in general, m-health system security capabilities and vulnerabilities may be unequally

distributed across technological, managerial, and interactional dimensions. This result implies that the security assessment of m-health systems may not be sufficient if it is performed from the perspective of just evaluating the quality of technological controls. Further research will deal with the development of PRISM-mH quantitatively, including AI-assisted scoring.

Funding: This research received no external funding.

Institutional Review Board Statement: No patients, healthcare providers, app users, or employees were recruited for this study. No personal health data or identifiable user data were collected or analyzed. The WeCare case assessment was based only on technical and documentary materials made available to the author for the study. The author declares that she has no employment, consultancy, advisory, ownership, technical, or commercial relationships with WeCare. An informed consent statement and consent to publish have been provided by the Chief Executive Officer of the WeCare Application.

Informed Consent Statement: Informed consent to participate and consent to publish were provided by the Chief Executive Officer of the WeCare Application. No individual patient or user data were collected.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The author thanks Al-Zaytoonah University of Jordan for providing a research-support environment, and WeCare Systems for providing the case study voluntarily for research purposes. Special thanks to Alaa Hussein, the Chief Executive Officer of the WeCare Application, for the professional research support.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. AlSlaity, A.; Suruliraj, B.; Oyebode, O.; Fowles, J.; Steeves, D.; Orji, R. Mobile applications for health and wellness: A systematic review. *Proc. ACM Hum.-Comput. Interact.* **2022**, *6*, 171. [CrossRef]
2. Handel, M.J. mHealth (mobile health)—Using apps for health and wellness. *Explor. J. Sci. Heal.* **2011**, *7*, 256–261. [CrossRef]
3. Debon, R.; Coleone, J.D.; Bellei, E.A.; De Marchi, A.C.B. Mobile health applications for chronic diseases: A systematic review of features for lifestyle improvement. *Diabetes Metab. Syndr. Clin. Res. Rev.* **2019**, *13*, 2507–2512. [CrossRef]
4. Zhou, T.; Wang, R.; Gu, S.J.; Xie, L.L.; Zhao, Q.H.; Xiao, M.Z.; Chen, Y.L. Effectiveness of mobile medical apps in ensuring medication safety among patients with chronic diseases: Systematic review and meta-analysis. *JMIR mHealth uHealth* **2022**, *10*, e39819. [CrossRef]
5. Butt, S.A.; Anjum, M.W.; Hassan, S.A.; Garai, A.; Onyema, E.M. Smart health application for remote tracking of ambulatory patients. In *Smart Healthcare System Design: Security and Privacy Aspects*; John Wiley & Sons: Hoboken, NJ, USA, 2022; pp. 33–55.
6. Elkefi, S. Access and usage of mobile health (mHealth) for communication, health monitoring, and decision-making among patients with multiple chronic diseases. *IISE Trans. Healthc. Syst. Eng.* **2024**, *14*, 179–192. [CrossRef]
7. Statista. mHealth. 2026. Available online: https://www.statista.com/topics/2263/mhealth (accessed on 10 February 2026).
8. Qan'ir, Y.; Khalifeh, A.H.; Eid, M.; Hammad, B.; Al-Batran, M. Mobile health apps use among Jordanian outpatients: A descriptive study. *Health Inform. J.* **2021**, *27*, 14604582211017940. [CrossRef]
9. Alhammad, N.; Alajlani, M.; Abd-Alrazaq, A.; Epiphaniou, G.; Arvanitis, T. Patients' perspectives on the data confidentiality, privacy, and security of mHealth apps: Systematic review. *J. Med. Internet Res.* **2024**, *26*, e50715. [CrossRef]
10. Luh, F.; Yen, Y. Cybersecurity in science and medicine: Threats and challenges. *Trends Biotechnol.* **2020**, *38*, 825–828. [CrossRef]
11. Tarikere, S.; Donner, I.; Woods, D. Diagnosing a healthcare cybersecurity crisis: The impact of IoMT advancements and 5G. *Bus. Horiz.* **2021**, *64*, 799–807. [CrossRef]
12. Ewoh, P.; Vartiainen, T. Vulnerability to cyberattacks and sociotechnical solutions for health care systems: Systematic review. *J. Med. Internet Res.* **2024**, *26*, e46904. [CrossRef]
13. Yang, L.; Wu, J.; Mo, X.; Chen, Y.; Huang, S.; Zhou, L.; Dai, J.; Xie, L.; Chen, S.; Shang, H.; et al. Changes in mobile health apps usage before and after the COVID-19 outbreak in China: Semilongitudinal survey. *JMIR Public Health Surveill.* **2023**, *9*, e40552. [CrossRef]
14. Cartwright, A.J. The elephant in the room: Cybersecurity in healthcare. *J. Clin. Monit. Comput.* **2023**, *37*, 1123–1132. [CrossRef]

15. Gupta, N.; Vashist, S.; Verma, R.; Terang, P.P. Cybersecurity and privacy concern in digital health. In *Navigating Innovations and Challenges in Travel Medicine and Digital Health*; IGI Global: Hershey, PA, USA, 2025; pp. 373–394.
16. Almaiah, M.A.; Al-Zahrani, A.; Almomani, O.; Alhwaitat, A.K. Classification of cyber security threats on mobile devices and applications. In *Artificial Intelligence and Blockchain for Future Cybersecurity Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 107–123.
17. Ikegwu, A.C.; Alo, U.R.; Nweke, H.F. Cyber threats in mobile healthcare applications: Systematic review of enabling technologies, threat models, detection approaches, and future directions. *Discov. Comput.* **2025**, *28*, 152. [CrossRef]
18. Statista. Mobile Cyberattacks Worldwide. 2025. Available online: https://www.statista.com/topics/11997/mobile-cyberattacks-worldwide (accessed on 15 February 2026).
19. Statista. Mobile Users Affected by Cyber Attacks worldwide. 2026. Available online: https://www.statista.com/statistics/1305965/mobile-users-cyber-attacks (accessed on 15 February 2026).
20. Statista. Number of Phishing Attacks Worldwide. 2026. Available online: https://www.statista.com/statistics/266155/number-of-phishing-attacks-worldwide (accessed on 15 February 2026).
21. Büyüközkan, G.; Güler, M. Cybersecurity maturity model: Systematic literature review and a proposed model. *Technol. Forecast. Soc. Change* **2025**, *213*, 123996. [CrossRef]
22. Al-Matari, O.M.; Helal, I.M.; Mazen, S.A.; Elhennawy, S. Adopting security maturity model to the organizations' capability model. *Egypt. Inform. J.* **2021**, *22*, 193–199. [CrossRef]
23. Paulk, M.C. A history of the capability maturity model for software. *ASQ Softw. Qual. Prof.* **2009**, *12*, 5–19.
24. Steenbergen, V.M.; Bos, R.; Brinkkemper, S.; Van De Weerd, I.; Bekkers, W. The design of focus area maturity models. In *Global Perspectives on Design Science Research*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; pp. 317–332.
25. Akinsanya, O.O.; Papadaki, M.; Sun, L. Towards a maturity model for health-care cloud security (M2HCS). *Inf. Comput. Secur.* **2020**, *28*, 321–345. [CrossRef]
26. Wani, T.A.; Mendoza, A.; Gray, K. A sociotechnical approach to bring-your-own-device security in hospitals: Development and pilot testing of a maturity model using mixed methods action research. *JMIR Hum. Factors* **2025**, *12*, e71912. [CrossRef]
27. Barnes, B.; Daim, T. Information security maturity model for healthcare organizations in the United States. *IEEE Trans. Eng. Manag.* **2022**, *71*, 928–939. [CrossRef]
28. Tarhan, A.K.; Garousi, V.; Turetken, O.; Söylemez, M.; Garossi, S. Maturity assessment and maturity models in health care: A multivocal literature review. *Digit. Health* **2020**, *6*, 2055207620914772. [CrossRef]
29. Ramadan, N.; Arafeh, M. Healthcare quality maturity assessment model based on quality drivers. *Int. J. Health Care Qual. Assur.* **2016**, *29*. [CrossRef]
30. Aldiqs, M. MA e-health system assessment, framework, and overview in Jordan. *Jordan J. Nurs. Res.* **2023**, *2*, 81–87.
31. Peffers, K.; Rothenberger, M.; Tuunanen, T.; Vaezi, R. Design science research evaluation. In *Proceedings of the International Conference on Design Science Research in Information Systems*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 398–410.
32. Carvalho, J.V.; Rocha, Á.; Abreu, A. Maturity models of healthcare information systems and technologies: A literature review. *J. Med. Syst.* **2016**, *40*, 131. [CrossRef]
33. Rojas, J.S.; Valencia, E.F.P.; Armas-Aguirre, J.; Molina, J.M.M. Cybersecurity maturity model for the protection and privacy of personal health data. In *Proceedings of the 2022 IEEE 2nd International Conference on Advanced Learning Technologies on Education & Research (ICALTER)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–4.
34. Thamilarasu, G.; Lakin, C. A security framework for mobile health applications. In *Proceedings of the 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*; IEEE: Piscataway, NJ, USA, 2017; pp. 221–226.
35. Novais, T.E.G. Healthcare Cloud Security Maturity Assessment Framework. Master's Dissertation, Universidade do Porto, Porto, Portugal, 2021.
36. Hussain, M.; Zaidan, A.A.; Zidan, B.B.; Iqbal, S.; Ahmed, M.M.; Albahri, O.S.; Albahri, A.S. Conceptual framework for the security of mobile health applications on Android platform. *Telemat. Inform.* **2018**, *35*, 1335–1354. [CrossRef]
37. World Health Organization. Cybersecurity and Privacy Maturity Assessment and Strengthening for Digital Health Information Systems. 2025. Available online: https://www.who.int/europe/publications/i/item/WHO-EURO-2025-11827-51599-79106 (accessed on 20 February 2026).
38. Sommerville, I. *Software Engineering*, 10th ed.; Pearson Education: Harlow, UK, 2016.
39. Pressman, R.S.; Maxim, B.R. *Software Engineering: A Practitioner's Approach*, 8th ed.; McGraw-Hill: New York, NY, USA, 2015.
40. Benjumea, J.; Ropero, J.; Rivera, O.; Dorronzoro-Zubiete, E.; Carrasco, A. Privacy and security of mHealth applications: A scoping review. *Int. J. Environ. Res. Public Health* **2020**, *17*, 787.
41. Martínez-Pérez, B.; de la Torre-Díez, I.; López-Coronado, M. Privacy and security in mobile health apps: A review and recommendations. *J. Med. Syst.* **2015**, *39*, 181. [CrossRef]

42. OWASP Foundation. OWASP Mobile Application Security Testing Guide (MASTG). 2023. Available online: https://mas.owasp.org/MASTG/ (accessed on 5 December 2025).
43. *ISO/IEC 27001:2022*; Information Security, Cybersecurity and Privacy Protection—Information Security Management Systems—Requirements. ISO: Geneva, Switzerland, 2022.
44. National Institute of Standards Technology. *Security Privacy Controls for Information Systems Organizations*; NISTSpecial Publication 800-53, Rev. 5; NIST: Gaithersburg, MD, USA, 2020. [CrossRef]
45. OWASP Foundation. OWASP Software Assurance Maturity Model (SAMM), v2.0. 2023. Available online: https://owaspsamm.org/docs/tools/pdf/ (accessed on 13 November 2025).
46. Sasse, M.A.; Brostoff, S.; Weirich, D. Transforming the 'weakest link'—A human/computer interaction approach to usable and effective security. *BT Technol. J.* **2001**, *19*, 122–131. [CrossRef]
47. Alahmari, A.; Duncan, B. Cybersecurity risk management framework for mobile health (mHealth). In *Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*; IEEE: Piscataway, NJ, USA, 2020.
48. Moro Visconti, R. Multilayer networks and public digital health scalability. *Int. J. Comput. Sci. Mob. Appl.* **2024**, *12*, 56–83.
49. Chaturvedi, I.; Pawar, P.M.; Muthalagu, R.; Tamizharasan, P.S. Zero trust security architecture for digital privacy in healthcare. In *Information Technology Security: Modern Trends and Challenges*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 1–23.
50. Zaki, J.; Islam, S.R.; Alghamdi, N.S.; Abdullah-Al-Wadud, M.; Kwak, K.S. Introducing cloud-assisted micro-service-based software development framework for healthcare systems. *IEEE Access* **2022**, *10*, 33332–33348. [CrossRef]
51. WeCare. WeCare Digital Health Platform. 2019. Available online: https://wecare-app.net/ (accessed on 15 December 2025).
52. WeCare. WeCare Seeker Application. 2019. Available online: http://l.ead.me/bb3ejG (accessed on 15 December 2025).
53. WeCare. WeCare HCP Application. 2019. Available online: https://l.ead.me/bb3rOU (accessed on 15 December 2025).
54. Aldhyani, T.H.; Nair, R.; Alkahtani, H.; Khalaf, O.I. AI-powered security-aware reconfiguration in cyber-physical systems for smart healthcare and energy domains. *Int. J. Adv. Soft Comput. Appl.* **2026**, *18*, 156–176. [CrossRef]
55. Hammad, R.; Abeer, A.S. A deep learning approach for predicting hospital length of stay for people with diabetes using electronic health records (EHRs). *Int. J. Adv. Soft Comput. Appl.* **2026**, *18*, 192–206. [CrossRef]
56. Raza, A.; Ali, A.K.S.; Hussain, A.A. AI-driven approaches to cyber and information security: Machine learning algorithms for threat prediction and anomaly detection. *Spectr. Eng. Sci.* **2024**, *2*, 565–573.
57. Shah, T.; Sampangi, R.; Siegel, A. Mobile application security risk score: A sensitive user input-based approach. In *Proceedings of the IEEE 3rd International Conference on AI in Cybersecurity (ICAIC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–10.
58. Rajaprakash, S.; Sujatha, G.; Kavitha, S.; Darsi, V.; Kurakalva, M.S.; Palapati, S.R. Cyber security for mobile applications using artificial intelligence. In *Proceedings of the World Conference on Information Systems for Business Management*; Springer: Cham, Switzerland, 2025; pp. 516–527.
59. Al-Rahayfeh, A. A context-aware and efficient method for handling missing data imputation in IoT systems. *Int. J. Adv. Soft. Comput. Appl.* **2025**, *17*, 55. [CrossRef]
60. OWASP Foundation. *OWASP Mobile Application Security Verification Standard (MASVS)*, Version 2.1.0; OWASP Foundation: Wilmington, DE, USA, 2024. Available online: https://mas.owasp.org/MASVS/ (accessed on 2 October 2026).
61. Turugare, R.; Shava, F.B.; Chitauro, M. An OWASP-Based Security Vulnerabilities Evaluation of Solo and Small Team Developed mHealth Applications. In *Proceedings of the International Conference on Data Engineering and Communication Technology*; Springer Nature: Singapore, 2024; pp. 125–138. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.