

RESEARCH

Open Access



Proposed framework for medication delivery system in the Jordanian public health sector

Ayah Elshebli^{1,2*}, Ghaleb Sweis², Ahmad Sharaf¹ and Ghaith Al Jaghbeer¹

Abstract

Background In Jordan, the confluence of traffic congestion and overcrowding in public hospitals poses a significant challenge for patients to collect their medications timely. This challenge was further intensified during the COVID-19 pandemic. Recognizing this issue, the Ministry of Health (MOH) and Electronic Health Solutions (EHS) intend to establish a Medication Delivery System (MDS), designed to provide patients with home delivery of medications and ensure proper treatment. This paper outlines a comprehensive framework to guide requirements engineers in devising an effective MDS framework, with a focus on expediting the development and testing processes and mitigating the risks associated with constructing such a system.

Method The proposed methodology entails a robust, structured approach to requirements development for an MDS that integrates an electronic health record system, billing system, pharmacy application, the patient-oriented My Hakeem app, and a delivery tracking system. The requirements elicitation and analysis processes were undertaken by a multidisciplinary committee from MOH and EHS teams, ensuring a diverse understanding of stakeholder needs and expectations. The requirement specifications were meticulously documented via a data dictionary, unified modeling language (UML), and context diagrams. The quality and accuracy of the requirements were verified through an extensive validation process, involving thorough review by various EHS teams and the MOH committee.

Results The MDS was implemented across numerous MOH facilities within a timeline that was a third of the original projection, leveraging the same level of resources and expertise. Post the requirements development phase, there were no changes requested by any stakeholders, indicating a high level of requirement accuracy and satisfaction.

Conclusion The study illustrates that our proposed methodology significantly results in a comprehensive, well-documented, and validated set of requirements, which streamlines the development and testing phases of the project and effectively eliminates requirement errors at an early stage of the requirements development process.

Keywords Requirements development, Medication delivery system, Electronic health record, My hakeem

*Correspondence:

Ayah Elshebli

ayah.alshebly@ehs.com.jo; ayah.alshebly@gmail.com

¹Software department, Electronic Health Solutions, Amman, Jordan

²Engineering Faculty, University of Jordan, Amman, Jordan



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

Background

Severe traffic congestion is caused by a significant increase in the number of vehicles on the streets of urban cities [1]. Similarly, roads in Amman, the capital of Jordan, are becoming increasingly congested [2], making it difficult and time-consuming for commuters to reach their destinations [3]. In addition, Jordan's public hospitals are overcrowded, causing frustration among the medical staff and negatively affecting patient satisfaction [4]. Consequently, the time required for patients to reach the hospital has increased.

During the COVID-19 pandemic, it was necessary to limit human interaction [5], and limiting patient visits to hospitals was a mitigating plan to reduce the risk of contracting the virus [6]. Accordingly, the demand for purchases from online pharmacies and self-medication significantly increased during the pandemic [5, 7]. Patients with multiple prescriptions may be particularly at risk of drug interactions and require special attention from pharmacists [8]. Therefore, patients should be more aware of the risks of buying medicines without consulting doctors [9]. Consequently, there is a need for a system that is not only concerned with drug availability but also provides the appropriate medication that should be assessed by the patients' physicians [10].

The Ministry of Health (MOH), in cooperation with Electronic Health Solutions (EHS), an innovative technology-driven company that provides automated solutions to improve the quality and efficiency of Jordanian public healthcare services [11], intends to implement a medication delivery system (MDS) in MOH public hospitals to deliver medications requested by specialized physicians for chronic disease patients without leaving their homes.

Risks associated with MDS development

The development of medication delivery system is fraught with its own set of challenges. The domain of healthcare presents an ecosystem distinct from other industries, characterized by its inherent complexities and unique challenges [12]. One such complexity arises from the convoluted hospital workflows and the varying nature of different departmental procedures. The challenge here lies in achieving seamless integration of diverse hospital information systems, which often becomes a substantial barrier to their effective implementation [13].

Further amplifying the complexity is the dynamic nature of the healthcare industry, with numerous components continuously interacting, leading to high ambiguity [14]. This reality, combined with the inherently intricate process of decision-making in environmental health policy — even under the most favorable conditions — underscores the difficulties faced in healthcare

decision-making due to the prevalent ambiguity and uncertainty [15].

Unfortunately, the percentage of software project failures is concerning. Around 19% of projects fail outright, while 52% experience creeping scope and budget overruns, predominantly attributable to inadequate business analysis [16]. One of the significant obstacles in software development is the integration of MDS components, primarily due to the need to tackle software integration issues [17].

Several risks and challenges arise during the integration of software system components. To begin with, inconsistencies in data represent a significant problem in data integration. These inconsistencies, more often than not, emerge in heterogeneous environments during information sharing, leading to concerns regarding data quality, accuracy, and integrity. Such discrepancies invariably impact end-users across both analytical and operational applications [18].

Effective creation of modules can be severely hampered when their objectives are not lucidly articulated. Without a clear vision of how they contribute to the larger scheme of things, integration, rework, and alignment issues inevitably ensue [19]. The ripple effect of this lack of clarity is that the time designated for system integration testing gets consumed in completing the integration process, thereby disrupting the project timeline [20].

Further challenges emerge when there's a common understanding gap regarding requirements and architecture among developers. When precise requirement documentation for the product is missing, it becomes difficult for developers to identify necessities and fill in requirement gaps [19]. This lack of clarity often leads to the creation of subsystems based on vague requirements, resulting in variable assumptions about the components of other subsystems. Consequently, this can cause poor design, incompatibility among components, and significant delays in the project's integration phase, especially when discrepancies unveiled during unit testing manifest during integration testing [17, 21].

Moreover, as software complexity and size increase, the need for informal communication becomes crucial [22]. Absence of this form of communication hampers the project participants' ability to stay updated on the developments at other sites, compromising coordination, causing integration issues, and potentially leading to system defects. This, in turn, prolongs the project's completion time [19].

The "big bang" integration approach typically seen in waterfall life cycle projects further amplifies the technical risk, given that the underlying assumptions in the design are not tested until late in the project [23]. This pitfall can be mitigated by opting for an incremental integration

approach that amalgamates software components at different levels, leading to frequent deliverables [24].

In order to ensure successful system development and integration, it is imperative to have detailed and precise system requirements and design [25]. Moreover, active involvement and collaboration during the requirements analysis phase fosters a deeper understanding and shared interpretation of requirements, leading to the creation of software components with optimal designs and smooth integration of their subcomponents [21]. Regular and early user feedback promotes crucial stakeholder involvement in a project, thereby bolstering the system's integrity [26].

State-of-the-art in requirements development

The requirements development phase is a critical step in the life cycle of software system design. It encompasses several activities, namely eliciting, analyzing, documenting, and validating requirements, resulting in a formal, structured document that serves as a guideline for the necessary system requirements [27]. This phase sets the foundation for the entire project, underpinning the development and testing stages, and ultimately determining the system's success or failure. Each of these activities represents an integral aspect of requirements development and will be expanded upon below.

Requirements elicitation

Requirement elicitation is the first step in the requirements development phase. It is the process of extracting user needs, desires, and constraints from various perspectives to gain an understanding of the problem at hand and derive potential solutions [28]. A comprehensive understanding of stakeholder demands is vital during this stage. Without it, there is a risk of developing a product that fails to meet user expectations [29].

This stage not only involves gathering initial requirements but also facilitating clear and feasible project vision, thereby ensuring that everyone involved is focused on the project's essential aspects and limiting the risk of scope creep [30]. A variety of elicitation techniques can be used, including interviews, observations, workshops, scenarios, and goal identification [31]. Choosing the most suitable methods often depends on the specific context, nature of the system, and stakeholders involved [31].

Requirements analysis

Post-elicitation, the gathered requirements undergo thorough scrutiny in the analysis phase [32]. This involves critically examining and refining the raw requirements and negotiating with stakeholders to reach a consensus, as these requirements may contain conflicts or be too ambiguous [33].

This is also the phase where requirements are prioritized. The MoSCoW technique, which categorizes requirements into "must have," "should have," "could have," and "won't have," is widely used for this purpose [34]. A gap analysis may be conducted to identify discrepancies between the current state and desired outcomes, laying the groundwork for devising a comprehensive plan to bridge these gaps [35].

Requirements specification

The term "requirements specification" or "requirements documentation" refers to the articulation and documentation of the requirements in an understandable and accessible manner [36]. Various models and diagrams are used for this purpose. For instance, context and data flow diagrams are employed to generate a process model of the system [37].

The context diagram, showing all external entities interacting with the system and the data flows between them, delineates the system's scope [38]. Additionally, the data dictionary fully defines the data structure [39]. The Unified Modeling Language (UML) is another popular tool that standardizes the system model's visualization [40]. It emphasizes design abstraction and highlights the consistency in systems and implementation through components and subcomponents [41]. It incorporates thirteen diagramming techniques to depict the problem domain from various angles [42]. These include sequence diagrams, state diagrams, activity diagrams, and use-case diagrams, each serving a unique purpose in describing system behaviors, interactions, and functionalities [43].

Requirements validation

Requirements validation is the final phase, ensuring that the gathered, analyzed, and documented requirements truly reflect user expectations [44]. This phase culminates in a formal document that provides a baseline for the agreed-upon software requirements [45]. This validation process includes checks for validity, realism, verifiability, consistency, and completeness [46].

Several validation techniques, such as requirement reviews with stakeholders and prototyping, can be used in isolation or in combination. However, these two techniques are often favored due to their effectiveness in obtaining direct feedback and iteratively refining the requirements until they are satisfactory [27]. By validating requirements early in the project, costly changes and errors in the later stages of development can be avoided.

The central goal of this study is to present a robust framework intended to guide requirements engineers in crafting an optimized MDS model. This model is anticipated to enhance the efficiency of both the development and testing phases, leading to a streamlined, effective solution for medication delivery. The proposed

framework will be fundamental in overcoming the challenges of system integration and minimizing the occurrence of errors during the initial stages of development, thereby paving the way for a more cost-effective and expedient delivery of healthcare services.

Method

Proposed framework for MDS development

The methodology proposed in this study strictly adheres to the well-established processes of software requirements development, as comprehensively described in [State-of-the-art in requirements development](#) This methodology introduces a framework that encompasses a meticulously curated collection of tools and techniques specifically designed to mitigate the risks and barriers identified in [Risks associated with MDS development](#). This mitigation strategy consequently facilitates the development and testing phases of the Medical Delivery System (MDS). To ensure effective requirement elicitation, analysis, and validation processes, a committee was formed. This committee is composed of key Ministry of Health (MOH) stakeholders possessing a wide range of knowledge backgrounds, including but not limited to technical, financial, and clinical expertise.

Table 1 Sample of the prioritized MDS requirements

ID	User Story	Priority
1	As a delivery courier, I want the patient to determine the delivery address when submitting the request so that I know the exact desired delivery address.	Must have
2	As an accountant, I want the patient to determine the coverage plan (insurance) when submitting the request so that it will be considered when calculating the medications' bill	Must have
3	As an accountant, I want to provide patient the ability to pay online for medication delivery requests without having to leave their home.	Must have
4	As an accountant, I want to allow patients to view the billing report so that they know the bill details.	Must have
5	As a pharmacist, I want to see the patient's medications so that I can rule out any interactions.	Must have
6	As an accountant, I want to be notified of the validated medications so that I can calculate the bill.	Must have
7	As a pharmacist, I want the medication delivery requests to be sorted by default based on the request data time so that I can start working with the oldest request	Could have
8	As a pharmacist, I want to have "Terms and Conditions" in the pharmacy application so that I know the product rules.	Could have
9	As a pharmacist, I want the option to decline delivery requests so that I can decline those that are somehow wrong or missed.	Should have
10	As a pharmacist, I want the last time I was logged into the app to appear in a notification when I'm logged into the pharmacy app so that I can highlight the last time I was logged in.	Won't have

Requirements elicitation in the MDS

To gather the necessary requirements and data required for constructing the MDS, a series of semi-structured interviews, workshops, and both active and passive observation sessions were conducted with the MOH committee and EHS teams. The interview questions used in these sessions are available as [Supplementary File 1]. These activities were performed during the requirement elicitation process. Furthermore, prototypes were utilized to materialize and visualize the system's requirements, thus aiding in eliciting further requirements that satisfied the committee's expectations.

Requirements analysis in the MDS

To evade potential requirement conflicts and scope creep risks, collaborative effort was employed by the MOH committee in conducting requirements analysis. This collaborative process was designed to further analyze and prioritize the raw requirements that were acquired during the aforementioned elicitation process.

The prioritization process utilized the MoSCoW technique, where requirements were divided into "Must have", "Could have", and "Should have" categories, along with "Won't have" for the current iteration. Table 1 provides a representative sample of the MDS system requirements, prioritized using MoSCoW technique, based on the requirements gathered during the elicitation process mentioned in [Requirements elicitation in the MDS](#).

Following the analysis, a workflow was prepared to plan how to address the gaps between the current state and the desired state that needs to be achieved. This workflow, illustrated in Fig. 1, demonstrates the cross-functional workflow of the envisioned MDS, taking into account the multiple participants in the workflow, including patients, pharmacists, accountants, and delivery couriers, and their interactions with the medication delivery request (MDR) from its inception by the patient until delivery or cancellation.

Requirements specification in the MDS

Once analyzed, requirements were further detailed using appropriate representation methods. For instance, a context diagram was created to investigate the environment and boundaries of the proposed system. Figure 2 depicts this context diagram of the proposed MDS, created using Microsoft Visio.

For a more in-depth depiction of the system architecture, Fig. 3 presents a detailed representation of the MDS architecture. The following section provides a description of the system components depicted in the Fig. 3:

- **The Electronic Health Information System:** consists of two key components associated with the MDS: The Veterans Health Information Systems

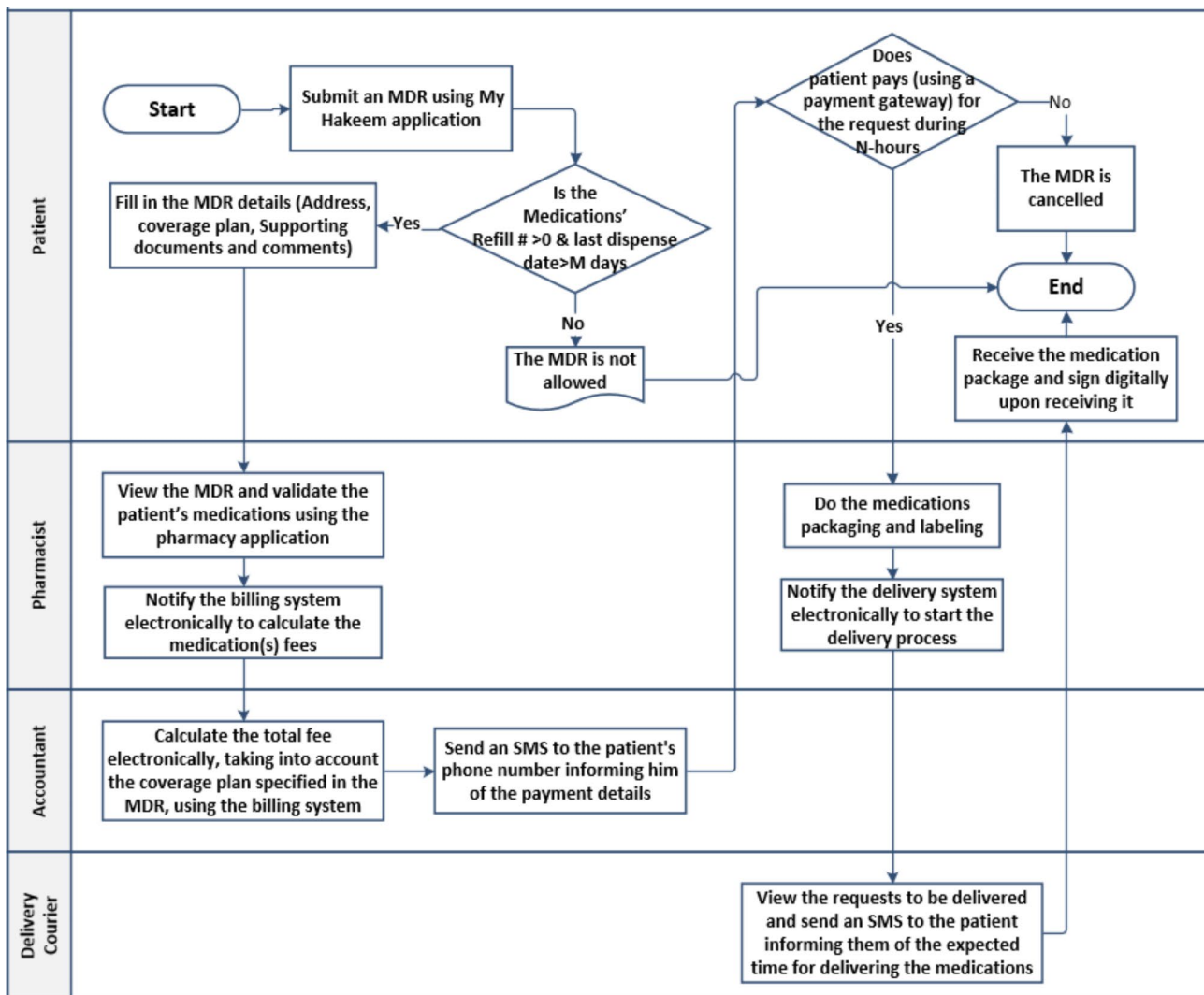


Fig. 1 Cross-functional flowchart of the MDS future state

and Technology Architecture (VistA) system and the Computerized Patient Record System (CPRS) application. VistA is an open-source electronic health record system created by the U.S. Department of Veterans Affairs (VA). VistA seamlessly integrates over a hundred software modules catering to outpatient and inpatient pharmacy, radiology, laboratory, and various clinical and administrative systems. To ensure a tailored implementation, the Electronic Health System (EHS) has customized VistA to align with Jordanian regulations, standards, language, and needs. VistA is integrated with a MUMPS non-relational database engine, known as GT.M (Greystone Technology M) [47]. CPRS is an application developed by the VA to offer healthcare providers convenient access to patients' health records that are stored in VistA. Through a user-friendly graphical interface, providers can easily

review comprehensive clinical information such as active problems, allergies, medications, clinical reminders, laboratory, vital signs, appointments, and clinical notes. Additionally, the application supports the entry of consults, notes, and various orders such as laboratory tests, radiology procedures, and medication prescriptions—all of which are saved on VistA [48].

- **My Hakeem application:** is the side of the system used by the patient, which is a web-based and mobile app that enables patients to view their medical information, such as medication orders, scheduled appointments, and lab test results that are saved in VistA. Furthermore, the My Hakeem app allows patients to submit a medication delivery request (MDR) to start the medicine delivery process. The details of these submitted MDRs are stored in the MyHakeem SQL database.

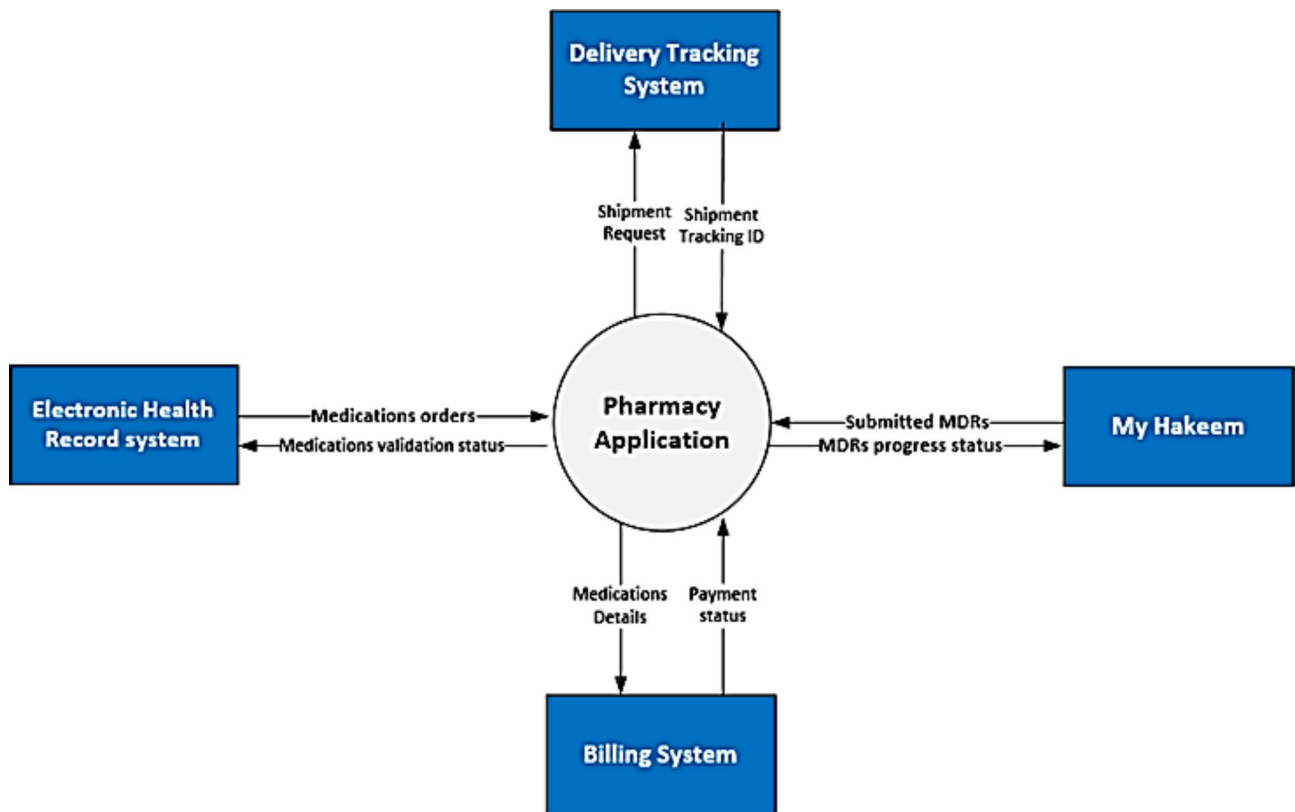


Fig. 2 MDS context diagram illustrating the data flow and boundaries of the MDS

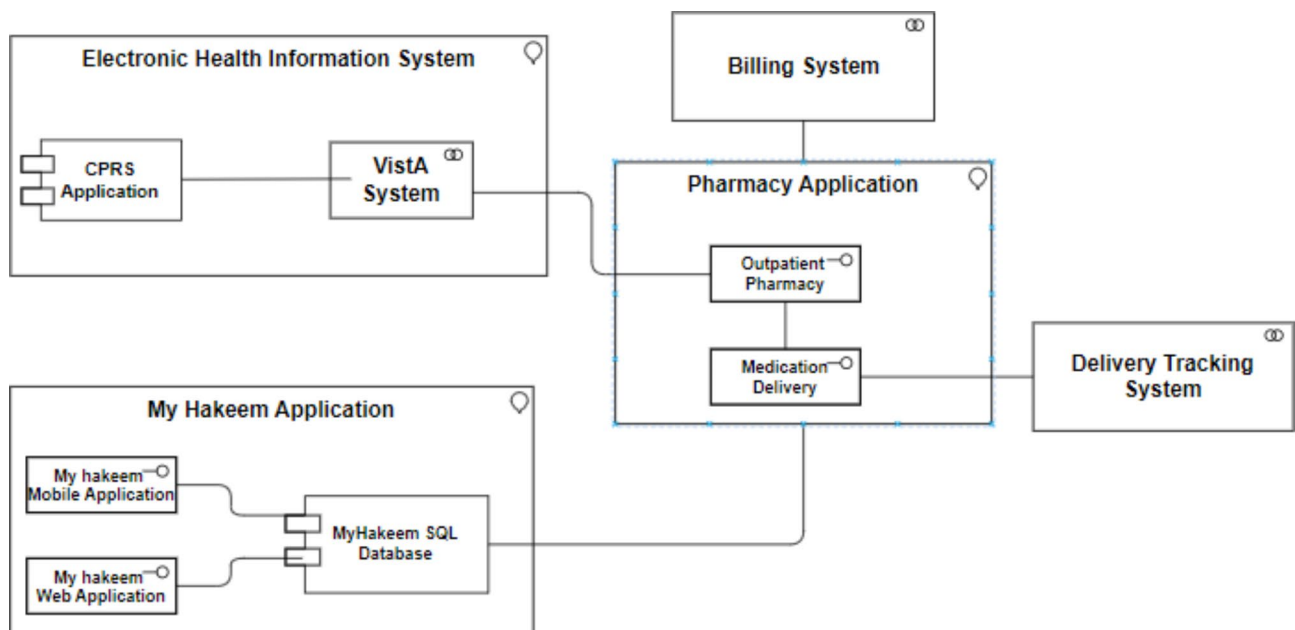


Fig. 3 MDS architecture overview

- Pharmacy Application:** is a web-based application, developed by EHS, consists of two modules: The Outpatient Pharmacy (OP) and the Remote Monthly Medication Refills (RMMR). The OP module,

developed by EHS, provides pharmacists with the capability to view, verify, validate, and dispense medication orders designated for outpatients. These orders were initially requested by physicians through

the CPRS application and are stored on VistA. Any actions taken on these orders are also recorded in VistA. Following the completion of the validation process by pharmacists, the details of the validated medications are then reflected in the billing system. The RMMR module, developed as part of MDS, enables pharmacists to examine the submitted MDRs stored in the MyHakeem SQL database. Pharmacists carry out the validation and verification process using the OP for prescriptions associated with patients who have an MDR. Subsequently, the pharmacist proceeds with the MDR within the medication delivery module to reflect it in the billing system. Upon completion of payment, the details related to the delivery address for the request are then reflected in the delivery tracking system.

- **Billing system:** is a system, developed by EHS, enables accountants to view comprehensive details of the validated medication orders received from the pharmacy application, such as the brand and quantity. Additionally, accountants can review request details retrieved from the medication delivery, which may include coverage plans if a patient wish to consider a new one instead of the previously saved plan in the billing system. The total fees are calculated based on the request details

and communicated to the patient via SMS. Patients then proceed with the payment through the online payment gateway.

- **Delivery Tracking System:** is a system was developed by Jordan Post (Jo Post) a licensed delivery company authorized by the Jordan Food and Drug Administration (JFDA), this system integrates seamlessly with the pharmacy application to track the MDR to the desired address. Upon completion of the medications packaging by pharmacists, the delivery tracking system is notified to begin the shipment process to the designated address chosen by the patient during the submission of the delivery request through the My Hakeem application.

For larger and more complex projects, context diagrams often struggle to simultaneously represent multiple perspectives. Therefore, use case diagrams, as they enable developers to better understand system requirements, are essential in these scenarios [49]. Figure 4 illustrates a comprehensive use case diagram, also drawn using Microsoft Visio, that depicts the primary actions that actors can perform on the system. This use case diagram illustrates that the initiation of medication delivery begins with the patient submitting a Medication Delivery Request (MDR) through the My Hakeem application, as

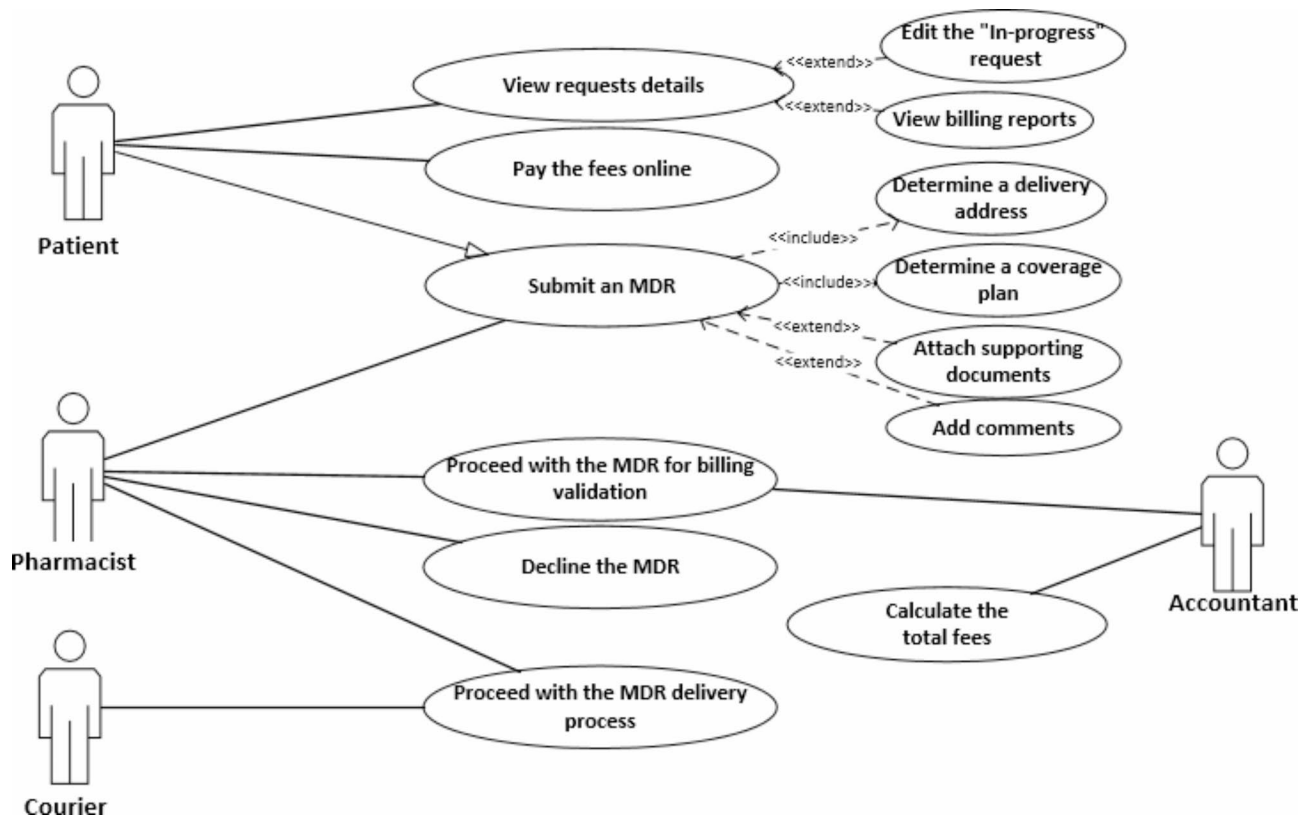


Fig. 4 Overall MDS functionalities use case diagram

previously mentioned in this [Requirements specification in the MDS](#). When submitting the request, the patient must specify the delivery address and select a coverage plan. Additionally, the patient has the option to include extra attachments or comments if necessary. The patient can access and review all details of their requests, edit ongoing requests, and view billing report information. Online payment of fees can also be completed through the integrated online payment gateway. The pharmacist can access the submitted Medication Delivery Request (MDR) through the pharmacy application, as highlighted earlier in this [Requirements specification in the MDS](#). They have the option to either reject the request or advance it to the billing system, providing the accountant with the necessary details to compute the total fees using the billing system. Subsequently, the pharmacist moves forward with the MDR into the delivery process, conveying essential delivery information to the delivery tracking system.

Each use case in the general diagram is subsequently described in detail from different perspectives using a use-case description table, sequence diagram, and a data dictionary table. An example of this is the “Submit an MDR” use case shown in Fig. 4, which is described in detail in Table 2.

Figure 5 presents an equivalent sequence diagram of the “Submit an MDR use case”, providing a clear depiction of the timing sequence of the interactions and operations involved in submitting an MDR.

A data dictionary was used to define the structure of data items in a standardized and clear manner. Table 3

displays the data dictionary for the fields in the “Request_details” table.

Validation process in the MDS

The requirements document underwent a rigorous and extensive validation process aimed at detecting any incompleteness, inconsistencies, or ambiguities. This process involved numerous teams within the EHS ecosystem including the development team, system integration team, pharmacy, and billing team, each of whom brought unique expertise to the process. The members of these teams were well-versed in both the functionalities of the system and the various workflows that were employed in the MOH.

Following this internal validation process, the requirements document was then subject to external validation by the MOH committee, whose role was to ensure that the defined requirements accurately reflected the expectations and needs of the client. This two-tier validation process not only helped to uncover any potential discrepancies but also provided a comprehensive perspective, balancing the technical intricacies of the system with the real-world applicability and relevance.

In order to facilitate the validation process and provide a more tangible representation of the requirements, the “Justinmind” application was employed to construct prototypes of the requirements. These prototypes served a dual purpose: firstly, they highlighted the strengths and weaknesses of the software, providing an additional layer of quality control. Secondly, they stimulated new ideas for potential requirements and opened up avenues

Table 2 Submit an MDR use case description

Actors	Trigger	Pre-conditions	Basic Path	Alternative Paths	Post-conditions	Exception Paths
Patient	The user clicks the “Submit a medication delivery request” button.	1. User is logged in “My Hakeem” application. 2. The patient’s monthly medications on the patient health record system meet the eligibility conditions for the request, which are: a) The previous date for dispensing medication was at least a month ago. b) The number of remaining refills is greater than zero.	1. Navigate to the “Medication Delivery” tab. 2. The user clicks on the “Submit a medication delivery request” button. 3. The system calls the “Get-Coverage-Plans” and “Get-DeliveryAddresses” APIs to view the patient’s coverage plans and addresses that are saved on the system. 4. The system displays the “Request Details” page and shows the patient’s address(es) in the “Your available addresses” section and the patients’ coverage plans in the “Insurance/Coverage Plan Information” section. 5. The user adds a new address and coverage plan 6. The user attaches the supporting document(s) and adds additional comments. 7. The user clicks the “Submit and confirm” button to submit the medication’s delivery request. 8. The user confirms the request.	A1: At step 3 of the basic path, if the patient has no previously saved addresses, the system displays “No saved addresses”. A2: At step 3 of the basic path, if the patient has no previously saved coverage plans, the system displays “No saved coverage plans”. A3: At step 5 of the basic path, if the user wants to consider any of the displayed coverage plans or addresses, they can mark them as checked.	1. A new request is inserted into the “MyHakeem” database. 2. The system displays the “The request has been submitted successfully” message.	E1: At step 1 of the basic path, if the patient doesn’t have a medication with a remaining refill number greater than zero and a last dispensed date greater than one month, the system prevents the user from making a request and displays the message “You have no due monthly medication listed on your medical record”. E2: At step 8 of the basic path, if the user cancels or doesn’t confirm the request, the request is not saved.

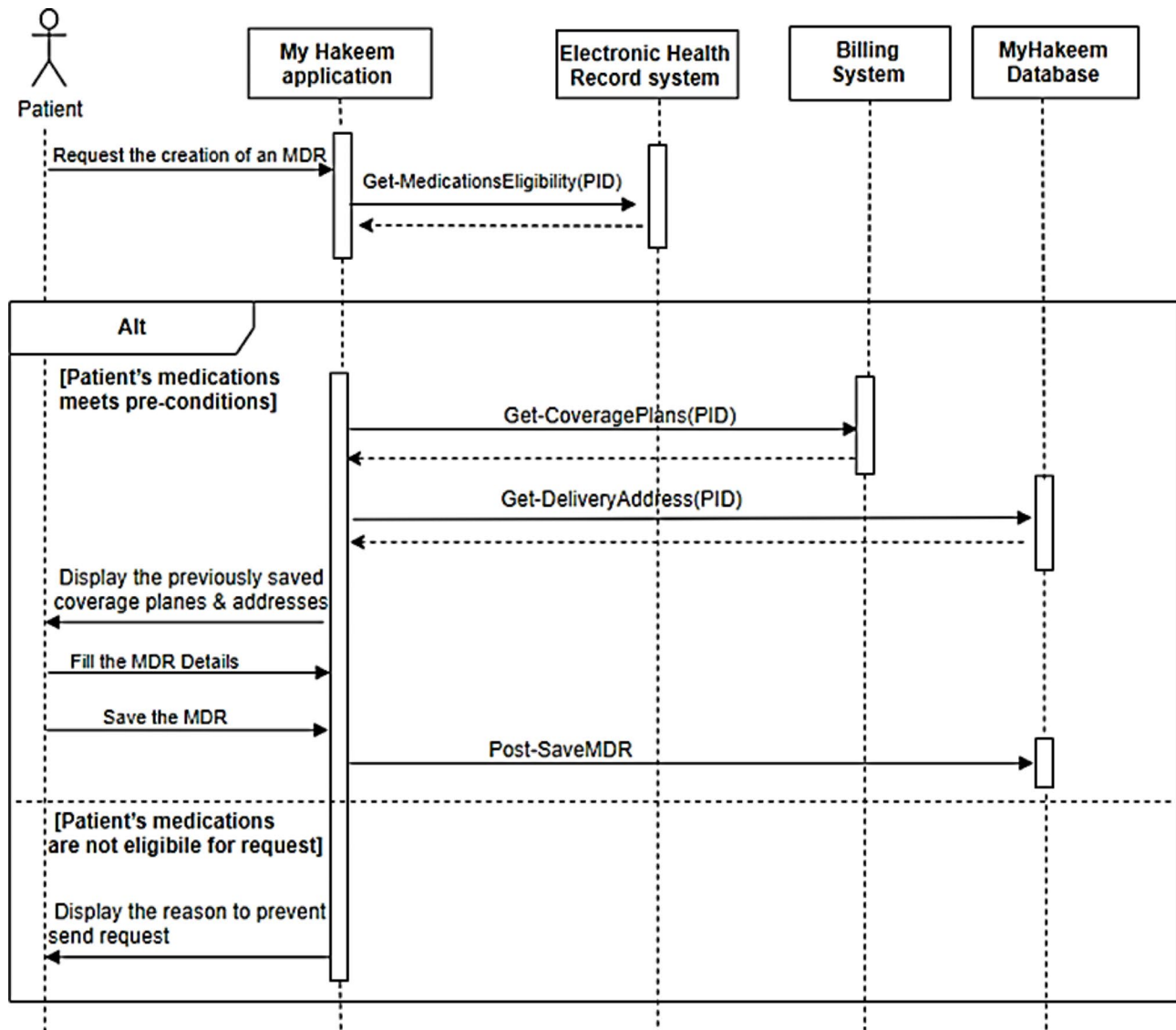


Fig. 5 MDR sequence diagram

for improvements. Figure 6 illustrates the prototype designed for the “Request Details” page, where patients enter their MDR details.

Throughout the validation process, a total of 132 requested changes in the requirements document were recorded, primarily due to incompleteness, ambiguities, and inconsistencies in the initial specifications. The process unfolded over three iterative rounds, each aimed at refining and improving the requirements. Table 4 offers a detailed breakdown of the progression of these changes, organized by the discovery round.

Upon the successful conclusion of the requirement validation process and the identification and rectification of all errors, the corrected requirements document was presented to the MOH committee. Their signature on the document symbolized their approval of the finalized

requirements and signaled the transition to the development phase, as per the documented requirements.

Requirements management in the MDS

An effective requirements management process was implemented to ensure the consistency, traceability, and ongoing management of the requirements throughout the development cycle of the MDS.

- **Traceability:** A traceability matrix was established to enable tracing each requirement from its origin through its development and testing, and finally to its deployment.
- **Change Management:** Given the dynamic nature of the healthcare industry, a process for managing changes to the requirements was established. This

Table 3 Data dictionary for the “Request_details” table

Field name	Description	Type	Unique	Mandatory	Default value	Additional information
ID	Identification code of the MDR	Int	Yes	Yes		The primary key (PK) of the table Length is 3–11 digits
PatientID	Identification code of the patient who submitted the MDR	Int	No	Yes		A foreign key (FK); reference to the “ID” field in the “Patient” table
AddressID	Identification code of the address at which the request will be delivered	Int	No	Yes	The last selected value	FK; reference to the “ID” field in the “Delivery_address_details” table
Coverage_Plan_ID	Identification code of the coverage plan that will be considered in the bill calculation	Int	No	Yes	The last selected value	FK; reference to the “ID” field in the “Coverage_plans_details” table
SupportingDoc_ID	Identification code of the supporting documents	Int	No	No		A maximum of three files are allowed to be attached.
Comment	Additional notes about the MDR	Varchar	No	No		The field length is 3-255 characters
Status_ID	Identification code of the MDR status	Int	No	Yes		FK; Reference to the “ID” field in the “Request_status” table
Created_at	The date and time of submitting the MDR	Datetime	No	Yes		

Request Details

Fig. 6 “Request details” page prototype

Table 4 Progress of changes in requirements by discovery round

Round Number	Number of changes in requirements
1	105
2	21
3	6

included an assessment of the impact of changes, documentation of changes, and communication of changes to all stakeholders.

- **Version Control:** The version control system is used to manage different versions of the requirement

documents. This ensured that every version of a document was available for review and reference.

Implementation and evaluation of requirements

Building upon the robust and well-defined requirements established in the development phase, the team initiated the implementation process for the MDS. This section will delve into the design, development, testing, and deployment phases, providing a detailed account of the activities performed and results obtained.

Design phase

The design phase kicked off with the creation of high-level system architecture, identifying main modules and their interactions. The architecture was guided by the context diagram (Fig. 2) and the general use case diagram (Fig. 4), providing a global view of the MDS system.

The system was segmented into five main modules: The Pharmacy Application, the billing system, the Electronic Health Record System, the Patient Application (My Hakeem), and the Delivery Tracking System. Interface and database design were carried out with due consideration to usability and data integrity, respectively. Use case diagrams, along with their descriptions (Table 2), guided the design of system functionalities.

The MDS was designed to be both web-based and mobile-based to cater to different user preferences. The interface designs were reviewed and approved by the MOH committee, and any necessary revisions were made based on their feedback.

Development phase

The development phase saw the transformation of designs into a functioning system. The MDS was developed using a modern and robust tech stack that combined Yii2 and flutter for both frontend and backend development. Database management was handled by MariaDB and GT.M, an open source MUMPS database engine.

The software development process followed an iterative model based on agile methodology where each iteration focused on implementing a specific subset of requirements, prioritizing those marked as “Must have” in the MoSCoW prioritization (Table 1). This allowed early feedback on system functionalities and facilitated necessary adjustments.

Testing phase

Testing was an ongoing process throughout the development phase, with unit tests conducted on individual components and integration tests performed to ensure seamless interaction between different components.

System testing was carried out to verify that the MDS met the specified requirements. Test cases were derived from use-case descriptions and linked to specific requirements in the traceability matrix, ensuring comprehensive coverage of all functional and non-functional requirements. Issues identified during testing were logged, prioritized, and addressed by the development team.

Acceptance testing was performed with the involvement of the MOH committee, with test scenarios designed to mimic real-world use of the MDS. The results showed that the system met the user needs and expectations as outlined in the requirements document.

Deployment phase

Upon successful acceptance testing, the MDS was deployed for live operation. It was launched in a phased manner, initially made available to a select group of users for beta testing. Feedback from these users helped identify and resolve any unforeseen issues before a full-scale launch.

Training sessions were organized for end-users, and comprehensive user manuals were developed and distributed. The MDS was made available on both web and mobile platforms, ensuring wide accessibility. Sample screenshots of the MDS applications in live operation are presented in Figs. 7 and 8.

Evaluation

Post-deployment, an evaluation was conducted to assess the effectiveness of the MDS against the initial goals and requirements. The system was found to be efficient, user-friendly, and robust, with positive feedback received from various stakeholders.

Performance metrics such as system response time, request processing time, and success rate of medication deliveries were within the specified requirements. The MDS also demonstrated high reliability and security, handling large volumes of transactions without any significant issues and maintaining stringent data security measures such as integrity, confidentiality, authentication and authorization [50].

All in all, the implementation and evaluation of the MDS requirements were successful, resulting in a well-designed and highly effective system. The thorough and meticulous requirements development process, as well as the user-centric approach adopted during design and development, played a crucial role in this success.

Results

The detailed requirements elicitation, analysis, specification, and validation processes in [Implementation and evaluation of requirements](#) yielded a comprehensive list of system requirements. A total of 657 requirements were identified, each addressing a specific user need or system functionality. These requirements were classified into different categories, such as system functionality, performance, usability, reliability, and security requirements.

The MOH committee played an active role throughout the requirements development process. The committee's diverse background was instrumental in the elicitation process, ensuring that all relevant perspectives were considered. The elicitation techniques, such as semi-structured interviews, workshops, and observation sessions, provided a rich understanding of the system's context and stakeholders' needs.

The requirements analysis process helped identify potential conflicts and avoid scope creep. The use of the

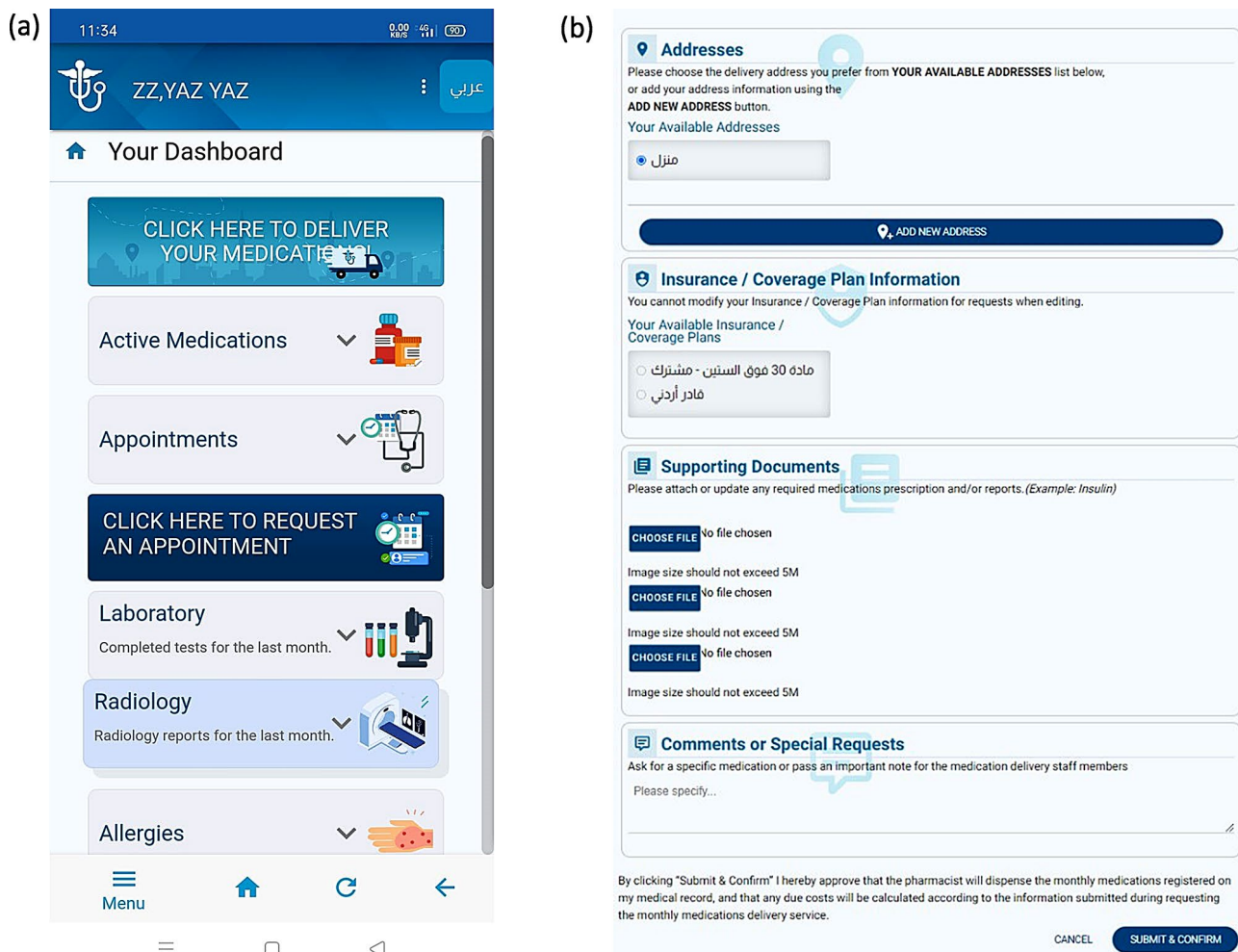


Fig. 7 (a) The My Hakeem mobile application dashboard for a testing patient (b) The “Request details” page on the My Hakeem web application

MoSCoW prioritization technique (Table 1) facilitated the selection of the most important functionalities for inclusion in the initial product release.

The requirements specification process resulted in a well-defined context diagram, use case diagrams, and use-case description tables. These specifications provided a clear understanding of the MDS boundaries, interactions, and functionalities. The data dictionary provided additional clarity on the data structures and relationships within the system.

The validation process involved multiple rounds of reviews and testing. The initial round revealed 105 changes, reflecting the inherent complexities of the MDS. Further rounds refined the requirements, yielding 21 and 6 changes respectively (Table 4). The prototypes created using Justinmind software helped the committee visualize the system, leading to insightful feedback and suggestions for improvement.

The result of the validation process was a refined set of requirements that effectively communicated the stakeholders’ needs and expectations. This rigorous process

helped improve the quality of the requirements document, reducing potential problems in the later stages of system development.

The implementation of a robust requirements management process helped ensure consistency and traceability of the requirements. The traceability matrix effectively linked each requirement to its corresponding source, design elements, and test cases. The change management process was instrumental in handling requirement changes, ensuring that all changes were assessed, documented, and communicated to stakeholders. Version control systems ensured the availability of all versions of the requirement document for future reference.

In conclusion, the requirement development process of the proposed MDS framework led to a comprehensive, well-documented, and validated set of requirements. This rigorous process has set a solid foundation for the subsequent phases of system development, paving the way for a successful implementation of the MDS.

(a)

#	Rx Number	Generic Name	Rt/Ri	Issue Date	Route	Sched	Dosage	Qty	Days/S	Ref#	Last Fill Date	Rem. Ref.	Patient Instructions	Brand Name	Billing Status
1	1012570233	METRONIDAZOLE 125/SPIRAMYCIN 750K TAB[W]		JAN 02, 2024	ORAL	Q8H	1 TABLET	22	7	0	JAN 02, 2024	0		DENTAGYL	Not Found
2	1012570235	ATORVASTATIN 20MG TAB,ORAL		JAN 02, 2024	ORAL	QDAY	1 TABLET	30	30	3	JAN 02, 2024	3		ADITOR 20MG ...	Not Found
3	1012570238	ASPIRIN BUFFERED 325MG TAB,ORAL		JAN 02, 2024	ORAL	QTUESDAY	0.5 TABLET	15	30	2	JAN 02, 2024	2		SALISAL PLUS ...	Not Found
4	1012570234	FUSIDIC ACID 20MG/GM GEL, TOP		JAN 02, 2024	TOPICAL	BID	A THIN LAYER	1	7	0	JAN 02, 2024	0		DERMOPUCIN G...	Not Found
5	1012570237	GLIMEPIRIDE 2MG TAB,ORAL		JAN 02, 2024	ORAL	QDAY	2 TABLETS	30	30	3	JAN 02, 2024	3	ON EMPTY STOMACH - For Diabetes	GLIMERYL 2MG...	Not Found
6	1012570236	INSULIN ISOPHANE 70/30 100 IU/ML VIAL(R)		JAN 02, 2024	SUBCUTANEOUS	AC		10	30	4	JAN 02, 2024	4		MIXTARD VIAL ...	Not Found

(b)

#	National ID	Document ID	Patient Name	Order Status	Division	Medical reports	Billing Report	Request Date/Time	Actions
1	333333333		TEST,PHU TEST PHU	Waiting payment	KARAK COMPREHENSIVE CENTER			2023-11-14 16:13:17	PROGRESS VIEW REQUEST
2	444444444		Test,Test test test3	Waiting payment	ZARQA NEW HOSPITAL			2023-11-14 23:01:02	PROGRESS VIEW REQUEST
3	555555555		TESTING,TEST TEST TEST	Payment collected	PRINCESS RAHMEH HOSPITAL			2023-11-14 13:45:18	PROGRESS VIEW REQUEST
4	888888888		TEST,PATIENT TEST TEST	Payment collected	AL RAMTHA COMPREHENSIVE CLINIC			2023-11-14 09:07:28	PROGRESS VIEW REQUEST
5	111111110		Test,TEST testing	In progress-Packaging	JARASH COMPREHENSIVE CLINIC			2023-11-14 15:55:19	UNPROGRESS VIEW REQUEST
6	444500477		ZZ,TEST TEST	In progress-Packaging	AL BASHEER HOSPITAL			2023-11-14 07:37:35	UNPROGRESS VIEW REQUEST

(c)

وزارة الصحة
Ministry of Health
العيادة العصبية للمريض

الرقم الوطني: 222222222
اسم المريض: تجربة تجربة تجربة تجربة

العمر: 25 سنة
رقم القاتورة: 13716053

العيادة: 1234567890
رقم المرجح: 1234567890

التاريخ	رمز الخدمة	الخدمة	الكمية	الوحدة	السعر	مساهمة المريض	مساهمة التأمين	المجموع
11/01/2023	006010102020001	Mixtard 70/30- (10 MI) 100 IU/M	3	VL	2.04	0	6.12	6.12
11/01/2023	006010103003006	Novofine Needles G31	30	BX	0.024	0	0.72	0.72
المجموع الكلي						0	6.84	6.84
المطلوب من المريض و واردات العلاج						0.556		0.556
واردات علاج						0.556		0.556

حالة دفع رسوم التوصيل: مدفوع
سيتم اضافة رسوم توصيل الأدوية: 2 دينار

Fig. 8 (a) The active order tab within the OP module in the pharmacy application (b) The “Processed Requests” tab within the RMMR module in the pharmacy application (c) The billing details, as presented on both the My Hakeem application and the pharmacy application, originate from the billing system

Discussion

The quality of the requirements is pivotal in the success of any system development project, and the implementation of the MDS was no exception. The early detection and rectification of errors in the requirement stage were instrumental in the efficient and effective development of the MDS, affirming the efficacy of the validation process employed.

Notably, two factors were identified to gauge the quality of requirements: changes in requirements and development time. The former quantifies the level of modifications required post-requirement analysis, whereas the latter measures the efficiency of the development process, relative to the quality of the initial requirements [51].

In the case of the MDS, the development and testing phases were completed within two months - a timeline significantly reduced to a third of the expected and planned duration. The successful condensation of the development period can be largely attributed to the quality and clarity of the initial requirements.

Moreover, the system has not encountered any significant errors or requirement changes since the implementation of its first version in February 2022. This underscores the success of the validation process and confirms the robustness of the requirement analysis and development phases. The stability of the system and its consistency with initial requirements suggest a high degree of accuracy and thoroughness in the requirement elicitation and validation process.

The system's iterations added new features and enhanced the system without the need for significant alterations to the initial requirements, demonstrating flexibility and scalability. These iterations were part of the system's evolution, signifying not a deficit in the original requirement specification, but a normal growth process in response to changing needs and advancements in technology.

The implementation of the MDS at 117 MOH facilities as of December 2023 and the successful delivery of

25,015 MDRs offer tangible evidence of the system's efficacy. Figure 9 illustrates the distribution of the MDRs successfully delivered across each quartile during the two years of the system's operation from 2022 to 2023. A continuous uptrend is observed in the number of the delivered MDRs; this upward trajectory is attributed to the growing number of facilities that are in operation as well as the satisfaction experienced by patients availing of the service. Remarkably, 74.6% of patients who initially requested MDRs have chosen to resubmit their MDR in the subsequent months to obtain their monthly medications. In addition, 71% of the requested MDRs were successfully delivered, with an average delivery time of 1.9 days from the time the MDR was requested. The MDRs that weren't delivered were either declined due to administrative reasons, such as missing documents or expired medication reports, or because the patients' payment process was not completed. The operational success of the MDS, in conjunction with the efficient development process and lack of significant errors, validates the requirements process that was undertaken.

In essence, this discussion asserts the crucial role that a well-structured requirement phase plays in system development. Early error detection and rectification, rigorous validation processes, and high-quality requirement specification can significantly impact the efficiency of the development phase and the effectiveness of the resulting system. These factors, as demonstrated in the MDS implementation, not only ensure timely project completion but also contribute to the system's longevity, flexibility, and adaptability to the evolving healthcare landscape.

Implications and limitations

Implications

The successful implementation of the MDS has significant implications, not only within the context of the MOH facilities but also for the broader healthcare sector. Firstly, it demonstrated the potential of digital healthcare services in enhancing patient care and providing solutions tailored to their specific needs. This suggests further opportunities to digitize healthcare services, with a focus on user-centered design.

The effectiveness of the rigorous requirements process used in this project highlights the potential benefits of adopting similar approaches in other system development projects. This study serves as a roadmap, guiding future projects in developing robust, error-free systems within shorter timeframes. The success of the MDS underlines the importance of carefully constructed and validated requirement phases in system development.

Additionally, this research provides a basis for future studies to examine the impact of similar systems in different healthcare settings. The scalability and flexibility of the MDS can be investigated further to determine

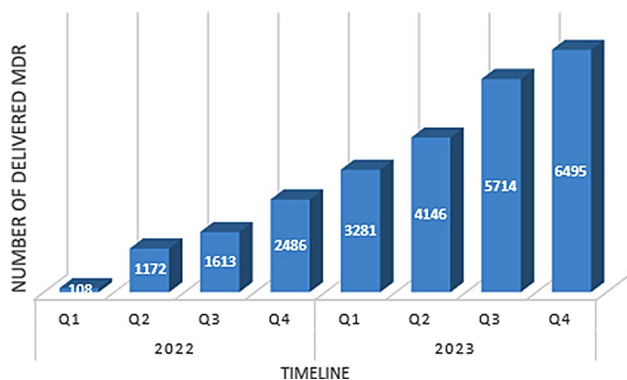


Fig. 9 Number of the delivered MDRs per quartile from 2022 to 2023

its adaptability in various environments and different healthcare systems.

Limitations

Despite the significant findings, this study is not without limitations. Firstly, the study was conducted within the specific context of MOH facilities, which may limit the generalizability of the findings to other settings. Healthcare systems vary across regions, and what worked within the MOH may not be directly applicable in other healthcare systems.

Secondly, while the MDS has not required significant changes since its inception, the system's long-term maintenance needs and adaptability to changing healthcare environments are not fully known. The current study did not include a thorough investigation of these aspects, which are important for understanding the system's longevity and future relevance.

These limitations, however, provide fertile ground for future research, offering opportunities to investigate these areas further, with a view to refining our understanding of the MDS and its potential application in diverse healthcare contexts.

Conclusion

This paper presented the development and implementation of the Medication Delivery System (MDS) at MOH facilities, aiming to streamline the medication delivery process and improve patient care. The study affirmed the critical importance of a robust requirements process in system development, with particular emphasis on requirement elicitation, analysis, and validation.

The well-structured requirements phase ensured early error detection and rectification, minimizing the time and resources needed for revisions during the later development stages. It also allowed for the MDS's successful development and testing within a compressed timeline, demonstrating efficiency gains achievable with meticulous requirements planning.

Further, the MDS has shown operational success since its first iteration in February 2022. The lack of significant errors or requirement changes is indicative of the quality and precision of the initial requirements. The successful delivery of 25,015 MDRs to patients without requiring them to leave their homes, across 117 MOH facilities as of Dec 2023, provided tangible evidence of the system's efficacy. Also, the ongoing increase in the number of submitted MDRs coupled with the statistics that 74.6% of patients have submitted MDRs more than once, alongside an impressive 71% success rate in delivering the requested MDRs with an average delivery time of 1.9 days, underscore the high level of satisfaction among patients utilizing the service.

The system's iterations, aimed at adding new features and enhancing the system, showed the scalability and flexibility of the MDS, underlining the importance of future-proofing during the requirements phase.

In summary, the MDS project demonstrates that rigorous requirement processes can significantly impact project timelines, resource management, system efficiency, and, most importantly, end-user satisfaction. Future system development projects can glean valuable insights from the MDS implementation, emphasizing the need for a thorough, precise, and forward-thinking requirements phase in the broader context of system development.

As the healthcare landscape continues to evolve, so will the demands on the systems designed to support it. Thus, the lessons learned from the MDS project are not only applicable to current system development endeavors but will remain relevant as we forge ahead in our mission to leverage technology in the service of healthcare.

Key Findings and Insights

Prior Knowledge on the Topic:

- The process of building integrated medical systems inherently presents unique challenges due to the specificities of the healthcare domain and the associated risks of integrating various subsystems.
- In many software projects, failure, scope creep, and budget overruns are frequently attributed to inadequate development of project requirements.

New Insights from This Study:

- This research demonstrates that a well-structured framework for requirements development can significantly enhance the efficiency and efficacy of development and testing phases in systems that involve integration of diverse components.
- The study also reveals the strategic importance of employing suitable tools and techniques during the early stages of requirements development. This proactive approach helps to identify and rectify errors and ambiguities in the requirements, mitigating future complications and unnecessary costs.

Abbreviations

MOH	Ministry of Health
EHS	Electronic Health Solutions
MDS	Medication Delivery System
UML	Unified Modeling Language
MDR	Medication Delivery Request
VistA	Veterans Health Information Systems and Technology Architecture
CPRS	Computerized Patient Record System
VA	Veterans Affairs
GT.M	Greystone Technology M
OP	Outpatient Pharmacy

RMMR	Remote Monthly Medication Refills
Jo Post	Jordan Post
JFDA	Jordan Food and Drug Administration
PK	Primary Key
FK	Foreign Key

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s12911-024-02673-2>.

Supplementary Material 1 : Supplementary File 1:[Research_Interview_ Questions], [Ayah Elshebli], [2022], [BMC Medical Informatics and Decision Making]

Author contributions

Authors AE, AS, and GA prepared the framework design and conducted its implementation. AE and GS contributed to writing the manuscript, incorporating input from all authors. All authors reviewed the manuscript.

Funding

This research did not receive any specific grant or funding from any agencies in the public, commercial, or not-for-profit sector.

Data availability

The underlying data essential to this research, supporting the study's findings, is accessible upon reasonable request from the corresponding author. The provided data encompasses the system requirements and software engineering documentation utilized during the development of the described system. However, please note that the comprehensive data specifically related to the requested medication delivery orders cannot be shared, as it contains sensitive patient information. This data is securely stored in compliance with privacy and confidentiality regulations. Researchers seeking access to the available data for purposes of replication or further investigation are welcome to contact the corresponding author at Ayah.Alshebli@ehs.com.jo.

Declarations

Ethics approval and consent to participate

The research conducted for this study, which involved gathering the necessary requirements and data for constructing the MDS, strictly adhered to the following ethical parameters: Compliance with Relevant Guidelines and Regulations: The study design and execution followed the ethical principles outlined in the "Declaration of Helsinki". Institutional and/or Licensing Committee Approval: Prior to the commencement of any experimental protocols, the research was subjected to approval by the Institutional Review Board (IRB) for the Ministry of Health in Jordan, known as Al-Bashir Hospital - Ethics Committee, to ensure conformity with the highest ethical standards. Informed consent was obtained from all subjects before their participation in the study.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 17 July 2023 / Accepted: 6 September 2024

Published online: 10 October 2024

References

1. Kumar A, Sing RR. Traffic congestion and possible solution in urban transport system, in 4th International Conference on Emerging Trends in Engineering Technology, Science and Management, 2017, pp. 603–607.
2. Tawil M, Reicher C, Ramadan KZ, Jafari M. Towards more pedestrian friendly streets in Jordan: the case of Al Medina Street in Amman. *J Sustain Dev*. 2014;7(2):144.
3. Al Tal R, Theodory R, Bazlamit SM. Assessing the intersected relationship between Land Use and Transportation Planning. *Geogr Environ Sustain*. 2023;15(4):80–9.
4. Aqel A. Jordan Hospital's ED overcrowding: experience and recent initiatives to address the problem. *Australas Emerg Nurs J*. 2011;14:S24.
5. Al-Zaidan M, Mohamed Ibrahim MI, Al-Kuwari MG, Mohammed AM, Mohammed MN, Abdulla SA. Qatar's Primary Health Care Medication Home Delivery Service: a response toward COVID-19. *J Multidiscip Healthc*, pp. 651–7, 2021.
6. Jairoun AA, Al-Hemyari SS, Abdulla NM, El-Dahiyat F, Jairoun M, Al-Tamimi SK. Online medication purchasing during the Covid-19 pandemic: potential risks to patient safety and the urgent need to develop more rigorous controls for purchasing online medications, a pilot study from the United Arab Emirates. *J Pharm Policy Pract*, 14, 2021.
7. Arias F, et al. A cross-sectional analysis of self-medication patterns during the COVID-19 pandemic in Ecuador. *Med (B Aires)*. 2022;58(11):1678.
8. Koehler JA, Abramowitz PW, Swim SE, Daniels CE. Indicators for the selection of ambulatory patients who warrant pharmacist monitoring. *Am J Heal Pharm*. 1989;46(4):729–32.
9. Desai KR, Chewning B, Wilcox A, Safdar N. Mail-order pharmacy experience of veterans living with AIDS/HIV. *Res Soc Adm Pharm*. 2018;14(2):153–61.
10. Hafner T, et al. Integrating pharmaceutical systems strengthening in the current global health scenario: three 'uncomfortable truths'. *J Pharm Policy Pract*. 2020;13:1–4.
11. Electronic Health Solutions - EHS | Electronic Health Solutions. (n.d.). Electronic Health Solutions. <https://ehs.com/jo/electronic-health-solutions-ehs>. Accessed 2 Feb,2023.
12. Kahraman C, Topcu YI. Operations research applications in health care management. Springer; 2018.
13. Sagirolu O, Ozturan M. Implementation difficulties of hospital information systems. *Inf Technol J*. 2006;5(5):892–9.
14. Carter M. Diagnosis: mismanagement of resources. *OR MS TODAY*. 2002;29(2):26–33.
15. Reis J, Spencer PS. Decision-making under uncertainty in environmental health policy: new approaches. *Environ Health Prev Med*. 2019;24(1):1–8.
16. Project Management Institute (PMI). Pulse of the profession overview. Newtown Square, PA: Project Management Institute; 2017.
17. Herbsleb JD, Grinter RE. Splitting the organization and integrating the code: Conway's law revisited, in Proceedings of the 21st international conference on Software engineering, 1999, pp. 85–95.
18. Wang X, Huang L-P, Zhang Y, Xu X-H, Chen J-Q. A solution of data inconsistencies in data integration—designed for pervasive computing environment. *J Comput Sci Technol*. 2010;25(3):499–508.
19. Zafar A, Ali S, Shahzad RK. Investigating integration challenges and solutions in global software development, in 2011 Frontiers of Information Technology, 2011, pp. 291–297.
20. Cusumano MA. Managing software development in globally distributed teams. *Commun ACM*. 2008;51(2):15–7.
21. Kommeren R, Parviainen P. Philips experiences in global distributed software development. *Empir Softw Eng*. 2007;12:647–60.
22. Cataldo M, Herbsleb JD. Communication networks in geographically distributed software development, in Proceedings of the 2008 ACM conference on Computer supported cooperative work, 2008, pp. 579–588.
23. Wang L, Tan KC. Software testing for safety critical applications. *IEEE Instrum Meas Mag*. 2005;8(2):38–47.
24. Battin RD, Crocker R, Kreidler J, Subramanian K. Leveraging resources in global software development. *IEEE Softw*. 2001;18(2):70–7.
25. Gotel O, Kulkarni V, Scharff C, Neak L. Integration starts on day one in global software development projects, in 2008 IEEE International Conference on Global Software Engineering, 2008, pp. 244–248.
26. Bosch J, Bosch-Sijtsema P. From integration to composition: on the impact of software product lines, global development and ecosystems. *J Syst Softw*. 2010;83(1):67–76.
27. Pandey D, Pandey V. Importance of requirement management: a requirement engineering concern. *Int J Res Dev Manag Rev*. 2012;1(1):66–70.
28. Iqbal T, Suaib M. Requirement elicitation technique: a review paper. *Int J Comput Math Sci*. 2014;3(9):1–6.
29. Pandey D, Suman U, Ramani AK. An effective requirement engineering process model for software development and requirements management, in 2010 International Conference on Advances in Recent Technologies in Communication and Computing, 2010, pp. 287–291.
30. Christenson D, Walker DHT. Understanding the role of 'vision' in project success. *Proj Manag J*. 2004;35(3):39–52.

31. Zowghi D, Coulin C. Requirements elicitation: a survey of techniques, approaches, and tools. *Eng Manag Softw Requir*, 2005.
32. Kuloor C, Eberlein A. Requirements engineering for software product lines, 2002.
33. Parviainen P, Tihinen M, Lormanms M, van Solingen R. Requirements engineering: dealing with the complexity of Sociotechnical Systems Development. in *Requirements engineering for sociotechnical systems*. IGI Global; 2005. pp. 1–20.
34. Cohn M. *Agile estimating and planning*. Pearson Education; 2005.
35. Kim S, Ji Y. Aug., Gap Analysis, pp. 1–6, 2018, <https://doi.org/10.1002/9781119010722.iesc0079>
36. Jarzębowicz A, Polocka K. Selecting requirements documentation techniques for software projects: a survey study, in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2017, pp. 1189–1198.
37. Ibrahim R. Formalization of the data flow diagram rules for consistency check. *arXiv Prepr. arXiv1011.0278*, 2010.
38. Sakra AA, Mosa DT. Data Flow diagrams of an Electronic Medical Record System in Mansoura Hospital. *Int J Comput Technol*. 2016;15(7):6885–97.
39. DeMarco T. *Structured analysis and system specification*. Inc., New York, New York: Yourdon; 1978.
40. Pender T. *UML 2 Bible*. Wiley; 2003.
41. Friedenthal S, Moore A, Steiner R. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann; 2014.
42. Booch G, Rumbaugh J, Jacobson I. *The unified modeling language reference manual*, 1999.
43. Booch G. *The unified modeling language user guide*. Pearson Education India; 2005.
44. Sommerville I, Sawyer P. *RE: a good practice guide*. John Wiley Sons. 1997;113:114.
45. Gilb T. *Competitive engineering: a handbook for systems engineering, requirements engineering, and software engineering using Planguage*. Elsevier; 2005.
46. Sommerville I. *Software engineering 9th Edition*, ISBN-10, vol. 137035152, p. 18, 2011.
47. U.S. Department of Veterans Affairs. <https://www.va.gov/health>. Accessed 19 Dec,2023.
48. Fletcher RD, Dayhoff RE, Wu CM, Graves A, Jones RE. Computerized medical records in the Department of Veterans affairs. *Cancer*. 2001;91:1603–6.
49. Mule SS, Waykar Y, Mahavidyalaya SV. Role of USE CASE diagram in s/w development. *Int J Manag Econ*, 2015.
50. Chandra M, Dhamija P. A Survey of Security Testing techniques in Software systems. *Int J Comput Appl*. 2016. <https://doi.org/10.1016/bs.adcom.2015.11.003>.
51. Raju S, Uma GV. A quantitative measurement of software requirement factors using goal question metric (gqm) approach. *IOSR J Comput Eng*. 2014;16(3):1–15.

Publisher's note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.