INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND COMPUTER SYSTEMS
E-ISSN: 2180-0650
VOLUME 11, ISSUE 1, 2025, pp. 1 – 15
DOI: https://doi.org/10.15282/ijsecs.11.1.2025.1.0133

RESEARCH ARTICLE

An Effective Approach for Eliciting Requirements and Ensuring Traceability: A Case Study of Automated Teller Machine

Habiba Azrin Mim

Department of Computer Science and Engineering, East West University, Dhaka 1212, Bangladesh

ABSTRACT – Requirement Elicitation plays a great role in the successful completion of a software project. Elicitation is a productive effort to extricate project-associated information from the stakeholders. To gather complete, concise, and clear requirements, the concept of requirement elicitation allows various analytics and techniques. Requirement elicitation is significant due to the lack of efficiency in accurately articulating their demands towards the intended system. Therefore, requirement engineers realize the need to perform elicitation to ensure that the requirements produced are easily accessible, useable, and applicable according to the client's demands. A software project can be considered successful if it satisfies all the requirements of the client. In certain situations, changes in requirements from the client side may arise during the development of the intended software. This research focuses on the challenges of this context mentioned as follows-many stakeholders are unable to clearly express their needs toward the intended system and the client-side requirements may change often. In this case, traceability plays a vital role in making the necessary changes in the components of the corresponding requirements. Therefore, the impact of any change can be easily visible & the changes can be applied correspondingly. After all, a project's success can be achieved by ensuring traceability among the components used in eliciting requirements. So, this research proposes an effective approach for eliciting functional and non-functional requirements and ensuring traceability among the corresponding components used in eliciting requirements. In addition, a tool has been designed to ensure traceability among functional requirements, non-functional requirements, and other corresponding components. If any change request arises to any of the requirements, all the corresponding components associated with that requirement can be traced with the help of this tool. In overall, this proposed approach has been applied successfully to a case study of an Automated Teller Machine using this tool.

ARTICLE HISTORY
Received : 28 July 2024
Revised : 6 March 2025
Accepted : 25 March 2025
Published : 28 March 2025

KEYWORDS
Requirement Elicitation
Requirements
Techniques
Traceability
Functional
Non-functional

1.0 INTRODUCTION

Requirements provided by the clients can be considered as the primary pillars for building the software. Software should be tailor-made to fit the client's requirements. According to the language of software engineering, the expectations of the users from the software product can be said as requirements. Requirement engineering includes the collection, analysis, and documentation of the software requirements from a client in a process. The features and functionalities of the target software are called the software requirements. Requirements can be categorized into two groups: functional and non-functional. The goal of requirement engineering is to be able to interpret the exact requirements of the clients so that the intended software product will get appreciation and satisfaction from its clients. Elicitation can be regarded as the procedure of collecting information from the end-user/stakeholder/client [1]. A successful software cannot be built without the proper elicitation of requirements. Requirement elicitation can be regarded as the very first step of software development life-cycle (SDLC). It is also known as the requirement gathering or collection phase. The software can be considered as successful if it meets all the expectations of its client.

Requirement Engineering (RE) contributes significantly to developing the exact software according to the stakeholder's needs [2]. Proper elicitation of Functional Requirements (FR) holds both the commitment of the stakeholders and reduces the chance of failure at an advanced stage. Similarly, proper elicitation of Non-Functional Requirements (NFR) increases the satisfaction of the stakeholders and future scopes for the industry [2]. Usually, the main priority is given to the system features during the software development process which are documented as functional requirements (FRs) [3]. To be specific, the FRs can be regarded as the behaviors/actions that the system supports [4]. To develop new software, elicitation of functional requirements specifies the need of the client, structures the communication between the stakeholders, and provides substantial evidence that controls the progress of the projects in software development related fields [5]-[10]. Modeling of functional requirements can be regarded as the medium between the stakeholders of the project to expedite communication. Thus, it is inevitable that the stakeholders as well as the end-users

*CORRESPONDING AUTHOR | H.A. Mim | ✉ azrinmim@gmail.com

narrate requirements unequivocally and the stakeholders such as requirement engineers interpret them correctly to the Developer team and QA team of the corresponding project [11]. On the contrary, the non-functional requirements (NFRs) connect to the system functionalities which do not infect the features of the system directly [12]. Functional requirements may vary due to the diversity of the functionalities of different software products. The representation of the NFRs can be done in different ways based on the justifications of their usage and the level of the project [13].

Requirements are extracted by applying different existing requirement elicitation techniques. There are four categories of requirement elicitation techniques according to their nature of communication: (i) Traditional Techniques, (ii) Collaborative Techniques, (iii) Cognitive Techniques, and (iv) Observational Techniques. The traditional techniques are as follows: Interview, Questionnaire, Data gathered from existing system, and survey. The collaborative techniques are as follows: Focus Group, Brainstorming, JAD, Prototyping, Workshop, Storyboarding, Models, Use-case/Scenarios and Viewpoint. The cognitive techniques are as follows: Document Analysis, Card Sorting, Protocol Analysis, Laddering, and Repository Grid. The observational techniques are as follows: Observation and Ethnography/Social Analysis. Software Project Team uses any of these techniques according to their need and expertise to elicit requirements.

Software requirements continuously vary for different reasons, frequent changes in software requirements are inevitable [14]. In the case of necessitated changes in any requirement, it is essential to ensure traceability to update its associations along with that requirement. Traceability of requirements provides the commitment to ensure the synchronization between the client's requirements and system evolution [15]. Software products should be developed according to the client's requirements so that they will gain a high level of quality as well as client satisfaction [16]. Association as well as communication among stakeholders is very significant for achieving success in eliciting software requirements and ensuring traceability [17]. From the above discussion, it is observed that the proper selection and implementation of requirement elicitation techniques and ensuring traceability among the corresponding components used in eliciting requirements can be considered as a significant option for performing this research. In addition, a tool has been developed to perform a case study on an automated teller machine (ATM) to verify the validity of this proposal.

2.0 RELATED WORKS

2.1 Functional Requirement

Functional requirements (FR) are the basic facilities as well as demands of the client that the system should offer. All the essential functionalities expected to be incorporated into the intended software as a part of the contract can be known as functional requirements.

2.2 Viewpoint

The method of Viewpoint (VP) is regarded as a technique for eliciting requirements. Viewpoints (VP) can be taken based on the types of users using the system. Under the viewpoints, the specific system requirements will be described correspondingly. Viewpoints are introduced at the very beginning of the requirement engineering process to structure the mechanism for eliciting requirements [18]. Identification of viewpoints and organization of system requirements under them at this earliest stage eliminates the possibility of missing requirements and provides a mechanism to ensure traceability for linking requirements with their viewpoints correspondingly [18]. Here, viewpoints are the end-users of the system who perform various activities in the system that can be regarded as the system requirements as well as the functional requirements of the system.

2.3 Operation Criteria

Operation Criteria (OC) can be considered as the criteria that should be maintained as prerequisites to perform any kind of operation in the system. These can also be considered as terms and conditions or rules that are mandatory to perform any sort of operation in the system. Some important factors need to be kept in mind while defining OC for any system:

- The environment where the system will be used.
- The type of data on which the system will work to perform any kind of operation.
- The type of audience/end-users who will use the system.

2.4 Non-functional Requirement

Non-functional requirements (NFR) can be considered as the quality factors that the system should be able to satisfy according to the project contract. These factors should be prioritized according to the nature and needs of the project. Different non-functional requirements are used in different software projects. Various non-functional requirements are as follows: Availability, Portability, Accuracy/Correctness, Security, Maintainability, Reliability, Scalability, Performance, Reusability, Flexibility, etc.

2.5 Traceability

Traceability is the ability to keep track of every single change performed to the requirement and connect them to the overall process of requirement management [19]. It reduces redundancy and increases the efficiency of the requirement management system. With the help of traceability, the effect of change in the requirement can be analyzed [20]. If the components involved in eliciting requirements are traceable, then if any change is performed in one component, the other components that are associated with it can be changed accordingly with the help of traceability. Proper indexing and referencing of the components are required to perform tracing. If any kind of error is found in the indexing and referencing of the components, then the probability of facing failure will increase. Ramamoorthy et al. addressed traceability as the ability of tracing from one entity to another on the basis of provided semantic relations in 1986 [21]. According to IEEE 1984, a software requirements specification can be traceable if the root of every requirements is distinct and if it expedites the referencing of each requirement in further development or enhancement documentation [21]. As stated by Carlshamre et al. in 2001, requirements are inter-linked with each other and they are dependent on other artifacts. Traceability provides the ability to trace these relationships of interdependency [22]. Successful implementation of tracing leads the way to success within a shorter time period. Thus, traceability plays a great role in managing requirements. The traceability among VP, FR, OC, and NFR is ensured in this research by applying the proposed methodology as shown in Figure 1:

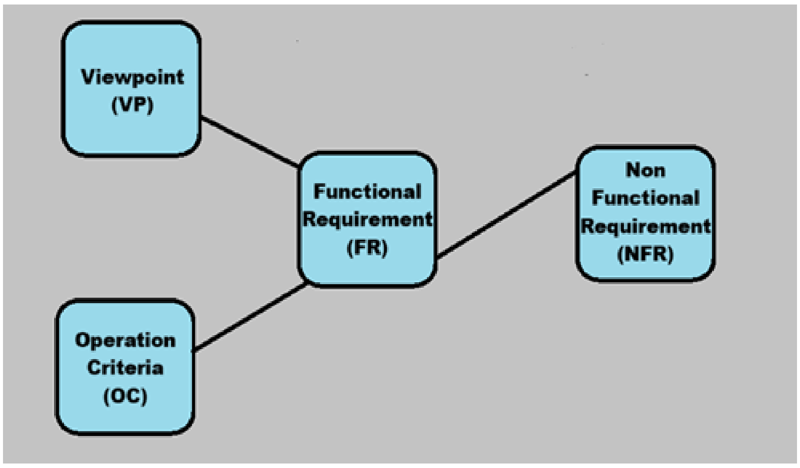

Figure 1. Traceability among VP, FR, OC, and NFR

Requirement elicitation techniques and associated tools can be considered as the key factors of requirement engineering. A standard outline of requirement engineering approaches by P. Rempel et al. was visualized in [23] where a multi-view approach was proposed including a profound differentiation of tools and techniques that were utilized in managing requirements. The comparison was based on 5 items: requirement traceability, requirements integration, requirement prioritizing, requirement status, and customer satisfaction. Several research issues, such as requirement ontology and traceability, were sparked by this study. By ensuring requirement traceability, the authors hoped to resolve the issue of requirements elicitation, verification, and validation. It is critical to collect requirements from stakeholders, but it is difficult to do so without first recognizing the needs of the end user. E. Scharifi et al. presented a synopsis of the requirement elicitation techniques in [24] and there the authors also highlighted the key issues faced by the industry, despite the numerous types of techniques that were suggested for the requirement elicitation process. According to this study, there is no single method that can meet all requirements elicitation requirements because there are so many problems, such as insufficiency in clarity regarding the purview of the project, inadequate stakeholder participation, inadequate communication and negotiation skills, ineffective techniques, tight deadlines, inadequate documentation, inadequate requirements management, and inconsistent requirements. In addition, there are some issues that were faced while implementing the elicitation techniques which must be resolved so that software can be developed as per the client's requirements. This research featured various issues and challenges of requirement elicitation techniques in the industry & it was contingent on the analysis of the previous data. In addition, this research also disclosed the need for creative techniques that are used to upgrade the existing methods/statistical analysis: a literature review method was adapted in this regard. This literature search was performed in view of several keywords, namely, requirements elicitation, requirements challenges, and validation techniques. Eventually, a critical analysis of the available literature was performed through the approach of thematic analysis. The results achieved were also analyzed using various parameters. The future work of this research was mentioned as the development of a systematic strategy for selecting the best and appropriate techniques for collecting software requirements. To utilize the efficacy of the requirement elicitation process, an approach was presented by. S. Tiwari et al. and S. S. Rathore et al. in [25] where the selection of a subset of requirement elicitation techniques to deliver an ideal outcome in the requirement elicitation process was shown. Their strategy consisted of three steps. In the first, they identified various attributes in three important dimensions: the project, the people, and the software development process, which could create an impact on the outcome of the elicitation process. In the subsequent step, they developed a three-p framework (3 PM) independently for each aspect, that imagines a connection between the elicitation methods and three elements of a product. In the third step, at last, they gave a planning model and used it in the determination of an elicitation method from the subset of elicitation procedures. Case studies were performed to evaluate and spread the circumstantial knowledge in the selection of requirement elicitation techniques according to their proposed methodology. Basically, they attempted to show significant discernments in the properties of various kinds of requirement elicitation methods. They expressed their future plan regarding the incorporation of industry data in the form of 3P matrices and the analysis of these matrices using an organization-centric case study in selecting elicitation techniques.

Proper requirements change management can be regarded as a dire need to manage continuously changing requirements. Traceability is significant for managing these changes and analyzing their effects in this regard. M. F. Bashir et al. presented a synopsis of existing techniques in the field of traceability in [26]. According to the authors' opinion, current traceability techniques were inadequate so problems were faced in managing changing requirements in a proper manner. So, they classified techniques into three classes and combined them to recognize the demerits of each other so that maximum benefits from traceability could be ensured. This paper focused on the complications regarding the definitions and techniques of traceability in the literature. In a nutshell, a general categorization of techniques was proposed so that practitioners could easily select their desired technique which would give the best results according to their needs from the existing techniques. A software traceability approach was proposed in [27] to support the impact of changes for object-oriented software. In this approach, the ability to incorporate the high-level software models with the low-level software models along with the requirements, test cases, design, and code were shown. This research supported both top-down and bottom-up traceability along with the provision of a direct connection between a component at one level to other components at various levels. The authors initiated a software prototype entitled Catia to provide assistance to C++ software performed a case study on an embedded system and discussed the results. In order to incorporate requirements into the low-level components, they combined dynamic and static analysis methods. In a powerful examination, they gave an interconnection between the prerequisites and experiments to the execution code. They provided connections between design model components and code components in static analysis. Because an artifact perspective can provide direct access to the impacted artifacts, the strategy of incorporating both top-down and bottom-up impacts supported provision for efficiency. That's what it's referenced, by applying this proposed strategy in a huge framework, the exhibition of the framework could corrupt. In large systems, maintenance personnel were typically more interested in acknowledging which classes or methods required modification than in the code's specific statements [28]. In conclusion, this prototype offered traceability infrastructures that included a number of measurements to balance the impact of changes. Future work was mentioned as the improvement of supporting GUI (Graphical User Interface) views linked to program views in order to improve user interface performance.

Many research studies have been performed on requirement elicitation or traceability. Proper selection of requirement elicitation techniques is significant as inappropriate requirements may cause high-level user disappointment and decrease the quality of the software [29]. Keeping proper documentation of elicited requirements according to the client's needs is very useful [30]. It contributes to processing further changes to any requirement. As requirement elicitation can be considered as the catalyst that can affect the success of the software product as well as the entire software project and traceability can be an effective solution to the problems in this regard, a link has been created between requirement elicitation and traceability through this research work. Four components and a procedure of six steps have been introduced for eliciting requirements and ensuring traceability among the corresponding components used in eliciting requirements in this research. The four components are as follows: viewpoints, functional requirements, operation criteria, and non-functional requirements. Direct Stakeholders can affect the functionalities of a system, so the method of viewpoint is used for identifying the stakeholders of a system as the viewpoints and eliciting their corresponding functional requirements as the functionalities of a system in this research work [31]. Identification of viewpoints, elicitation of functional requirements, identification of operation criteria, elicitation of non-functional requirements, building relationship between functional requirement and viewpoint, functional requirement and operation criteria and functional requirement and non-functional requirement, and eventually the traceability among viewpoint, functional requirement, operation criteria, and non-functional requirement are performed correspondingly. A tool has been implemented to ensure traceability among these corresponding components. In addition, a case study is applied to an Automated Teller Machine according to the proposed approach using the newly developed tool. This research ensures transparency among the stakeholders in accessing the system. Similarly, it increases the reliability of the client on the software project team. The ease of accessibility and maintainability of traceability are provided according to the proposed approach. It reduces redundancy and increases the efficiency of the components used in ensuring traceability.

3.0 METHODS AND MATERIAL

According to the proposed approach, there are six steps to perform the requirement elicitation along with traceability analysis using four components (VP, FR, OC, and NFR). The description and Overview Diagram (Figure 2) of the proposed methodology are given below:

Step-1: Identifying the VPs of the corresponding system.

Step-2: Eliciting the FR from the activities of the corresponding VPs accordingly.

Step-3: Identifying and set some OC for the corresponding system.

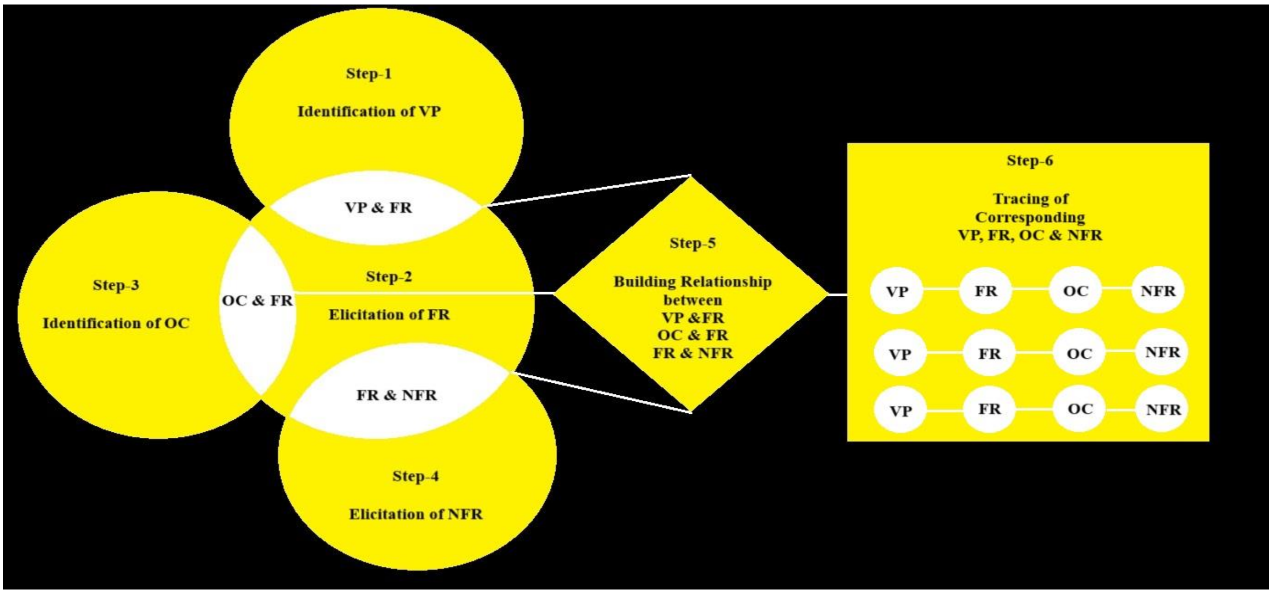

Figure 2. Overview Diagram

Step-4: Eliciting the NFR for the corresponding system and set some questions to determine the status of the NFR in the following system.

Step-5: After identifying VP and OC and eliciting FR and NFR, building the relationship between VP and FR, OC and FR and FR and NFR. Correspondingly, VP, OC, and NFR have separate relationships with FR.

Step-6: Finally, tracing the corresponding VP, FR, OC, and NFR using the relationship between VP and FR, OC and FR and FR and NFR.

3.1 Eliciting FR

In this research, the method of Viewpoint is used for eliciting functional requirements. Different VPs have different FRs correspondingly shown in Figure 3 below:

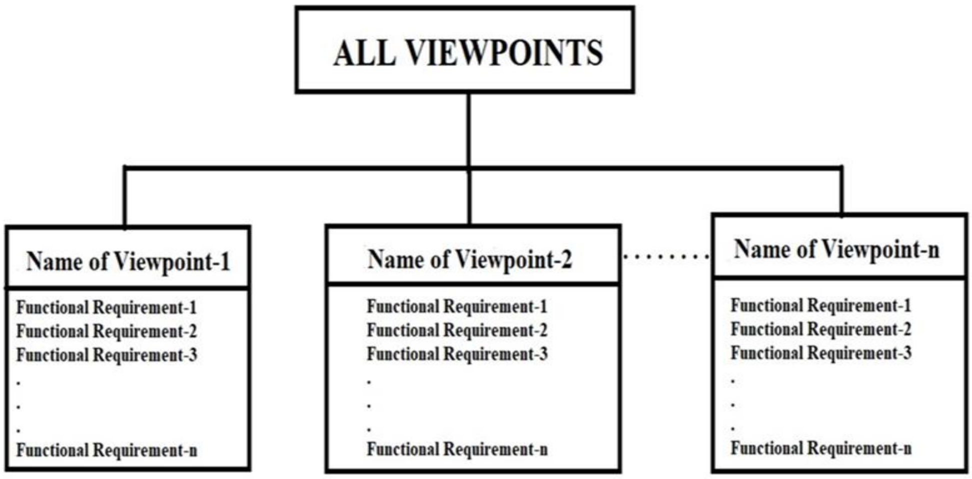

Figure 3. Generalized Diagram of Viewpoint Method for eliciting Functional Requirements

3.2 Eliciting NFR

According to the proposed approach, the author has selected the following 5 non-functional requirements as the standard NFRs that have to be ensured constantly at various extents in all sorts of projects:

- Availability
- Performance
- Security
- Accuracy/Correctness

- Reliability

To ensure the status of these non-functional requirements in a system, a pre-defined set of dichotomous questions on NFR (NFRQ): Availability, Performance, Security, Accuracy or Correctness, and Reliability will be asked to the stakeholders of the system as shown in Table 1:

Table 1. Generalized NFRQ/A for eliciting NFR

NFR ID	NFR Name	NFR Question(NFRQ)	Answer
NFR-1	Availability	Is this service available?	Yes/No
NFR-2	Performance	Does the service process the user's request according tothe pre-defined time?	Yes/No
NFR-3	Security	Are the credentials of thecorresponding user secured while accessing theservice?	Yes/No
NFR-4	Correctness/Accuracy	Does the service function correctly?	Yes/No
NFR-5	Reliability	Can the users rely on this system to access the service?	Yes/No

3.3 Building Relational Tables

According to the proposed approach, each viewpoint has single/multiple functional requirements as shown in the Overview Diagram (Figure 2). So, there is a relationship between VP and FR. The corresponding FRs of a VP can be viewed from the following sample of a table:

Table 2. Sample of a Relational Table showing the Relationship between VP and FR

VP ID	FR ID
V1	FR-1.x
	FR-1.n
.	.
.	.
Vn	FR-n.x
	FR-n.n

According to the proposed approach, each functional requirement has a relationship of dependency with single/multiple operation criteria as shown in the Overview Diagram (Figure 2). The OC of a dependent FR can be viewed from the following sample of a table:

Table 3. Sample of a Relational Table showing the Relationship between OC and FR

FR ID	Dependent on 'OC ID'
FR-1.x	Select from OC-x to OC-n
FR-1.n	Select from OC-x to OC-n
.	.
.	.
FR-n.x	Select from OC-x to OC-n
FR-n.n	Select from OC-x to OC-n

According to the proposed approach, each functional requirement has a relationship with single/multiple non-functional requirements as shown in the Overview Diagram (Figure 2). The corresponding NFRs of a FR can be viewed from the following sample of a table:

Table 4. Sample of a Relational Table showing the Relationship between FR and NFR

FR ID	NFR ID
FR-1.x	Select from NFR-1 to NFR-5
FR-1.n	Select from NFR-1 to NFR-5
.	.
.	.
FR-n.x	Select from NFR-1 to NFR-5
FR-n.n	Select from NFR-1 to NFR-5

3.4 Ensuring Traceability

In this research, VP, FR, OC, and NFR can be traced correspondingly using the relationship between VP and FR, OC and FR and FR and NFR as shown in the Overview Diagram above (Figure 2). According to the proposed approach, the corresponding VP, FR, OC, and NFR can be viewed from the following sample of ensuring traceability in a tabular form:

Table 5. Traceability among VP, FR, OC and NFR

VP ID	FR ID	OC ID	NFR ID
V1	FR-1.x	Select from OC-x to OC-n	Select from NFR-1 to NFR-5
	FR-1.n	Select from OC-x to OC-n	Select from NFR-1 to NFR-5
.	.	.	.
.	.	.	.
Vn	FR-n.x	Select from OC-x to OC-n	Select from NFR-1 to NFR-5
	FR-n.n	Select from OC-x to OC-n	Select from NFR-1 to NFR-5

3.5 Case Study

A case study is applied to an Automated Teller Machine to perform requirement elicitation and traceability analysis according to the proposed approach.

ATM: An ATM (Automated Teller Machine) is an electronic machine that was invented for performing financial transactions. According to the term ‘automated’, no banking representative, teller, or human cashier is required for the banking platform ATM. It can be considered a self-service banking outlet [32].

Components of ATM:

The automatic teller machine consists of four input devices and four output devices, which are described and shown in Figure 4 below:

Input Devices:

- Keypad: The keypad allows the user to enter the secret identification number of the card for authentication.
- Card Reader: After providing the correct secret identification number, the card reader reads the information on the card of the user and passes it to the host server.
- Screen Buttons: It takes the desired service input from the user and processes the user’s request.
- Deposit Slot: It receives cash/cheque from the user for deposit purposes.

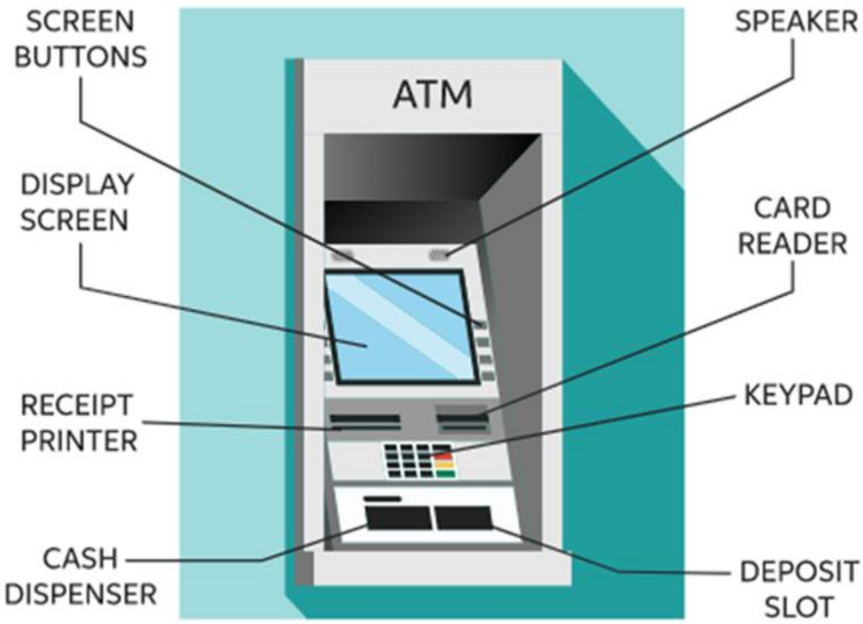

Figure 4. ATM

Output Devices:

- Speaker: It provides audio instruction to the user when the user does any action in the ATM.
- Display Screen: It displays the main menu, each step, and information about all the services to allow the user to carry out the transaction and other activities.
- Receipt Printer: It prints the transaction records and statements of the user.
- Cash Dispenser: It dispenses the amount of money that the user wants to withdraw.

Features of ATM:

Automated Teller Machine provides 24×7 access. It reduces the hassle of the customers to stand in long queues at the bank for simpler transactions and the workload of the bank officials. Some important features of an ATM are described and shown in Figure 5 below-

- Checking balance
- Withdrawing balance
- Transferring balance
- Depositing cash
- Depositing cheque
- Ordering bank-statement
- Ordering cheque
- Issuing Mini-statement

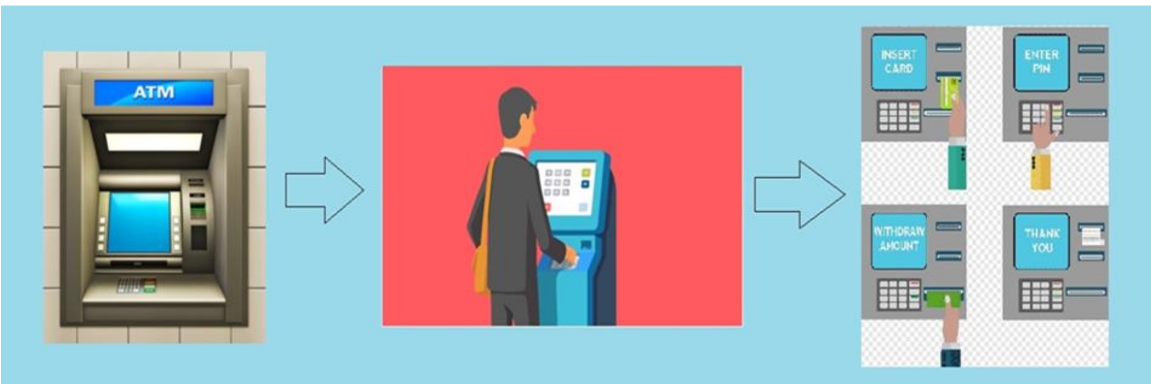

Figure 5. Steps of performing a transaction using an ATM

Connection with the Database Management System (DBMS): ATM uses a DBMS from where all the information about all the account holders of the bank can be found. A connection between the ATM and the database management system has to be established for performing any kind of transaction as well as processing the user's request. No action will have any effect on the ATM without establishing this connection. After withdrawing and depositing cash using the

ATM, the new balance gets updated in the database management system. Thus, the database management system be considered as the back-end and the ATM interface can be considered as the front-end as shown in Figure 6 below:

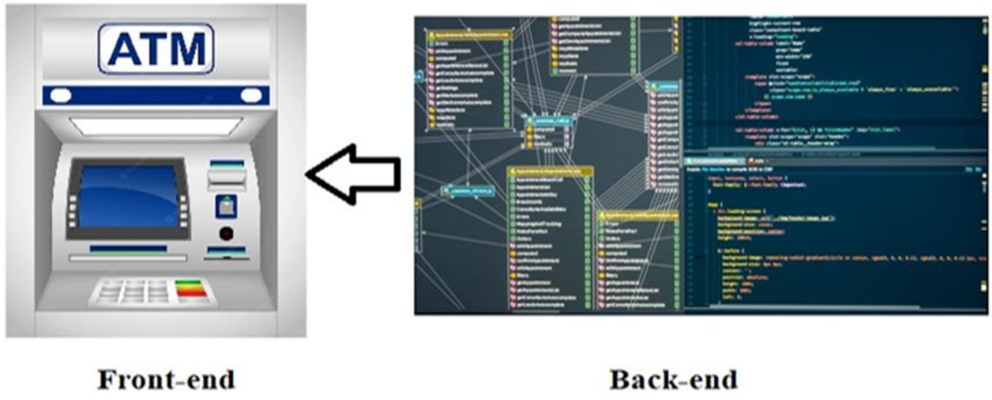

Figure 6. Connection between ATM and Database Management System

3.6 Implementation

A case study is applied to an Automated Teller Machine for eliciting requirements and ensuring traceability according to the proposed approach. As discussed in the proposed approach, the 6 steps that are required for eliciting requirement and ensuring traceability among the corresponding components used in eliciting requirements of ATM are performed below:

According to the proposed approach, following Steps- 1 and 2, the viewpoints for ATM have been identified as shown below: (1) ATM Operator, (2) Account Holder, (3) Other Customer, and (4) System.

The numbering of the VPs for ATM has been performed as shown in the table below:

Table 6. Definition Table for VP

Viewpoint ID	Name
V1	ATM Operator
V2	Account Holder
V3	Other Customer
V4	System

Following Step-1 and Step-2, the viewpoint diagram for ATM is shown in Figure 7 below where the FRs of all the VPs have been elicited correspondingly:

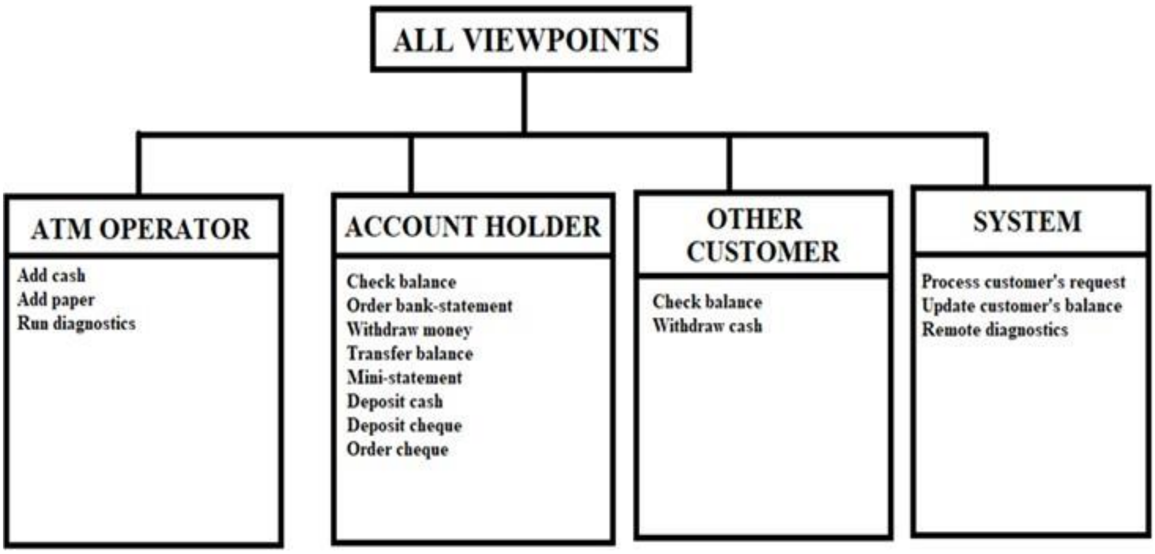

Figure 7. Viewpoint Diagram for eliciting FRs

The numbering of the FRs of the corresponding VPs for ATM has been performed as shown in the table below:

Table 7. Definition Table for FR

Functional Requirement ID (FR ID)	Description
FR-1.1	Add cash
FR-1.2	Add paper
FR-1.3	Run diagnostics
FR-2.1	Check balance
FR-2.2	Order bank-statement
FR-2.3	Withdraw money
FR-2.4	Transfer balance
FR-2.5	Mini-statement
FR-2.6	Deposit cash
FR-2.7	Deposit cheque
FR-2.8	Order cheque
FR-3.1	Check balance
FR-3.2	Withdraw cash
FR-4.1	Process customer’s request
FR-4.2	Update the customer’s balance
FR-4.3	Remote diagnostics

According to the proposed approach, following Step-3, the OC for ATM have been identified and set. In addition, the numbering has been performed as shown in the table below:

Table 8. Definition Table for OC

Operation Criteria ID (OC ID)	Description
OC-1	Receipt paper must be present in the machine.
OC-2	Money must be present in the machine.
OC-3	The ATM machine must be properly connected to the database.
OC-4	All the services that the ATM machine is displaying must be valid and active.
OC-5	The credentials of all the customers must be present & updated in the database.
OC-6	A pre-defined minimum amount of balance must be present in the customer’s account.

According to the proposed approach, following Step-4, the NFRs for ATM are selected and the corresponding questions are set. Thus, the NFRs have been elicited. In addition, the numbering has been performed as shown in the table below:

Table 9. Definition Table for NFR

Non-functional Requirement ID (NFR ID)	Name	Question
NFR-1	Availability	Is this service available?
NFR-2	Performance	Does the service process the user's request according to the pre-defined time?
NFR-3	Security	Are the credentials of the corresponding user secured while accessing the service?
NFR-4	Correctness/Accuracy	Does the service function correctly?
NFR-5	Reliability	Can the users rely on this system to access the service?

According to the proposed approach, following Step-5, the relationship between VPs and FRs has been built correspondingly for the case study of ATM as shown in the table below:

Table 10. Relational Table of VP & FR

VP ID	FR ID
V1	FR-1.1
	FR-1.2
	FR-1.3
V2	FR-2.1
	FR-2.2
	FR-2.3
	FR-2.4
	FR-2.5
	FR-2.6
	FR-2.7
	FR-2.8
V3	FR-3.1
	FR-3.2
V4	FR-4.1
	FR-4.2
	FR-4.3

According to the proposed approach, following Step-5, the relationship between FRs and OC has been built on the basis of dependency for the case study of ATM as shown in the table below:

Table 11. Relational Table of FR and OC

FR ID	Dependent on 'OC ID'
FR-1.1	OC-2
FR-1.2	OC-1
FR-1.3	OC-4
FR-2.1	OC-3, OC-4, OC-5
FR-2.2	OC-3, OC-4, OC-5
FR-2.3	OC-2, OC-3, OC-4, OC-5, OC-6
FR-2.4	OC-3, OC-4, OC-5, OC-6

FR-2.5	OC-1, OC-3, OC-4, OC-5
FR-2.6	OC-3, OC-4, OC-5
FR-2.7	OC-3, OC-4, OC-5
FR-2.8	OC-3, OC-4, OC-5
FR-3.1	OC-3, OC-4, OC-5
FR-3.2	OC-2, OC-3, OC-4, OC-5, OC-6
FR-4.1	OC-3, OC-5
FR-4.2	OC-3, OC-5
FR-4.3	OC-3, OC-4

According to the proposed approach, following Step-5, the relationship between FRs and NFRs has been built correspondingly for the case study of ATM as shown in the table below:

Table 12. Relational Table of FR and NFR

FR ID	NFR ID
FR-1.1	Null
FR-1.2	Null
FR-1.3	NFR-1, NFR-2, NFR-4
FR-2.1	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
FR-2.2	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
FR-2.3	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
FR-2.4	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
FR-2.5	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
FR-2.6	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
FR-2.7	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
FR-2.8	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
FR-3.1	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
FR-3.2	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
FR-4.1	NFR-2, NFR-3, NFR-4
FR-4.2	NFR-3, NFR-4
FR-4.3	NFR-1, NFR-4

4.0 RESULT AND DISCUSSION

According to the proposed approach, following Step-6, the traceability among VP, FR, OC, and NFR has been ensured correspondingly using the relationship between VP and FR, OC and FR, and FR and NFR for the case study of ATM as shown in the table below:

Table 13. Final Output (Traceability among VP, FR, OC and NFR)

VP ID	FR ID	OC ID	NFR ID
	FR-1.1	OC-2	Null

V1	FR-1.2	OC-1	Null
	FR-1.3	OC-4	NFR-1, NFR-2, NFR-4
V2	FR-2.1	OC-3, OC-4, OC-5	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
	FR-2.2	OC-3, OC-4, OC-5	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
	FR-2.3	OC-2, OC-3, OC-4, OC-5, OC-6	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
	FR-2.4	OC-3, OC-4, OC-5, OC-6	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
	FR-2.5	OC-1, OC-3, OC-4, OC-5	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
	FR-2.6	OC-3, OC-4, OC-5	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
	FR-2.7	OC-3, OC-4, OC-5	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
	FR-2.8	OC-3, OC-4, OC-5	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
V3	FR-3.1	OC-3, OC-4, OC-5	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
	FR-3.2	OC-2, OC-3, OC-4, OC-5, OC-6	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5
V4	FR-4.1	OC-3, OC-5	NFR-2, NFR-3, NFR-4
	FR-4.2	OC-3, OC-5	NFR-3, NFR-4
	FR-4.3	OC-3, OC-4	NFR-1, NFR-4

Microsoft Visual Studio 2022 has been used to implement the tool to achieve the goal of the proposed approach for the case study of ATM as shown below:

Figure 8. Final Output from the Tool (Traceability among Corresponding VP, FR, OC, and NFR

To summarize, the proposed approach comprising six steps has been applied successfully through the help of a newly developed tool on the following four components: VP, FR, OC, and NFR as shown above. In the first step, the VPs are identified. In the second step, the elicitation of FRs is performed. In the third step, the OC are identified. In the fourth step, the elicitation of NFRs is performed. In the fifth step, the relationship between FR and VP, FR and OC, and FR and NFR are built correspondingly. Finally, in the sixth step, traceability among VP, FR, OC, and NFR is ensured correspondingly. In addition, a case study has been performed on a small-scale project entitled Automated Teller Machine which requires low space complexity and low time complexity using the newly developed tool.

5.0 CONCLUSIONS

Finally, an effective approach for eliciting requirements and ensuring traceability among the components used in eliciting requirements has been performed through this research work. Traceability among VP, FR, OC, and NFR prevents the risk of software project failure at an advanced stage. In this research, a tool has been developed to provide automated traceability. It has shown the desired result in terms of a small-scale project of an Automated Teller Machine in the case study. However, the scenario will be different for large-scale projects that require high time complexity and high space complexity. So, future work can be regarded as the implementation of a large project requiring high time complexity and high space complexity. Moreover, inputs are provided manually to the system according to the proposed approach which creates a possibility of errors occurring in indexing and referencing. So, another future can be considered as the creation of an automated system for providing input to the system in an automated manner.

ACKNOWLEDGEMENTS

The author expresses gratitude to Professor Dr. Shamim H Ripon from East West University, the Editors and Reviewers of IJSECS for their guidance, support and cooperation regarding this research and publication.

This research did not receive any financial support from funding organizations in the public, private, or nonprofit sectors.

AUTHORS CONTRIBUTION

Habiba Azrin Mim (Conceptualization; Formal Analysis; Data curation; Visualization; Supervision; Investigation; Methodology; Validation; Writing - original draft; Software; editing; Resources.)

CONFLICT OF INTEREST

The author declares no conflicts of interest.

REFERENCES

[1] P. Zave, “Classification of Research Efforts in Requirements Engineering,” 1997.

[2] T. Shah and S. Patel, “A Novel Approach for Specifying Functional and Non-functional Requirements Using RDS (Requirement Description Schema),” *Procedia Comput. Sci.*, vol. 79, pp. 852–860, 2016, doi: 10.1016/J.PROCS.2016.03.083.

[3] A. Jarzebowicz and P. Weichbroth, “A Qualitative Study on Non-Functional Requirements in Agile Software Development,” *IEEE Access*, vol. 9, pp. 40458–40475, 2021, doi: 10.1109/ACCESS.2021.3064424.

[4] J. D. Summers, S. Joshi, and B. Morkos, “Requirements evolution: Relating functional and non-functional requirement change on student project success,” *Proc. ASME Des. Eng. Tech. Conf.*, vol. 3, 2014, doi: 10.1115/DETC2014-35023.

[5] “User Stories: An Agile Introduction.” https://agilemodeling.com/artifacts/userstory.htm (accessed Dec. 20, 2024).

[6] D. Avison and G. Fitzgerald, “Information systems development: methodologies, techniques & tools,” *Inf. Syst. Ser.*, p. 645, 2006, Accessed: Dec. 20, 2024. [Online]. Available: https://books.google.com/books/about/Information_Systems_Development.html?id=xk9dPwAACAAJ

[7] H. F. Hofmann and F. Lehner, “Requirements engineering as a success factor in software projects,” *IEEE Softw.*, vol. 18, no. 4, pp. 58–66, Jul. 2001, doi: 10.1109/MS.2001.936219.

[8] “Loucopoulos, P. and Karakostas, V. (1995) System Requirements Engineering. McGraw Hill, London. - References - Scientific Research Publishing.” https://www.scirp.org/reference/referencespapers?referenceid=3428127 (accessed Dec. 20, 2024).

[9] E. Mumford, “Effective systems design and requirements analysis : the ETHICS approach,” p. 188, 1995.

[10] “Nuseibeh, B. and Easterbrook, S. (2000) Requirements Engineering A Roadmap. Proceedings of the Conference on the Future of Software Engineering, 35-46. - References - Scientific Research Publishing.” https://www.scirp.org/reference/referencespapers?referenceid=1706167 (accessed Dec. 20, 2024).

[11] A. Benfell, “Modeling functional requirements using tacit knowledge: a design science research methodology informed approach,” *Requir. Eng.*, vol. 26, no. 1, pp. 25–42, 2021, doi: 10.1007/s00766-020-00330-4.

[12] S. Kugele, W. Haberl, M. Tautschnig, and M. Wechs, “Optimizing automatic deployment using non-functional requirement annotations,” *Commun. Comput. Inf. Sci.*, vol. 17 CCIS, pp. 400–414, 2008, doi: 10.1007/978-3-540-88479-8_28.

[13] M. A. Gondal, N. A. Qureshi, H. Mukhtar, and H. F. Ahmed, “An engineering approach to integrate Non-Functional Requirements (NFR) to achieve high quality software process,” *ICEIS 2020 - Proc. 22nd Int. Conf. Enterp. Inf. Syst.*, vol. 2, no. Iceis, pp. 377–384, 2020, doi: 10.5220/0009568503770384.

[14] B. J. M. Abma and I. Kurtev, “Evaluation of requirements management tools with support for traceability-based change impact analysis,” 2009.

[15] B. Ramesh and M. Jarke, “Towards Reference Models for,” *Source*, vol. 1, no. 1, pp. 1–54, 1999, [Online]. Available:

http://www.springerlink.com/index/6q31q7556n440211.pdf%5Cnhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:TOWARDS+REFERENCE+MODELS+FOR+REQUIREMENTS+TRACEABILITY#2%5Cnhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Towards+Refer

[16] A. A. Attia and M. A. Soliman, "EIJEST A NEW MEHODOLOGY USING QFD FOR TRACING REQUIREMENTS IN," vol. 14, no. 3, pp. 288–303, 2011.

[17] "D. Duarte, C. Farinha, M. M. da silva, and A.R. da silva, 'Collaborative Requirement Elicitation with visualization Techniques,' IEEE, pp. 343 - 348 2012.".

[18] I. Sommerville, P. Sawyer, and S. Viller, "Viewpoints for requirements elicitation: A practical approach," *Proc. IEEE Int. Conf. Requir. Eng.*, pp. 74–81, 1998, doi: 10.1109/ICRE.1998.667811.

[19] M. Ahmed Abbasi, J. Jabeen, Y. Hafeez, D.-B. Batool, and N. Fareen, "Assessment of Requirement Elicitation Tools and Techniques by Various Parameters," *Softw. Eng.*, vol. 3, no. 2, pp. 7–11, 2015, doi: 10.11648/j.se.20150302.11.

[20] F. Khursheed and M. Suaib, "A Survey on Importance of Requirement Traceability in Software Engineering," *Int. J. Eng. Innov. Technol.*, vol. 5, no. 4, pp. 109–112, 2015, doi: 10.17605/OSF.IO/ZPVSK.

[21] O. C. Z. Gotel and A. C. W. Finkelstein, "Analysis of the requirements traceability problem," *Proc. Int. Conf. Requir. Eng.*, pp. 94–101, 1994, doi: 10.1109/icre.1994.292398.

[22] R. Torkar, T. Gorschek, R. Feldt, M. Svahnberg, U. A. Raja, and K. Kamran, "Requirements traceability: A systematic review and industry case study," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 22, no. 3, pp. 385–433, 2012, doi: 10.1142/S021819401250009X.

[23] P. Rempel and P. Mader, "Preventing defects: The impact of requirements traceability completeness on software quality," *IEEE Trans. Softw. Eng.*, vol. 43, no. 8, pp. 777–797, Aug. 2017, doi: 10.1109/TSE.2016.2622264.

[24] E. Scharifi, A. Danilenko, U. Weidig, and K. Steinhoff, "Influence of plastic deformation gradients at room temperature on precipitation kinetics and mechanical properties of high- strength aluminum alloys," *J. Eng. Res. Appl.*, vol. 9, no. 1, pp. 24–29, 2019, doi: 10.9790/9622.

[25] S. Tiwari and S. S. Rathore, "A Methodology for the Selection of Requirement Elicitation Techniques," 2017, [Online]. Available: http://arxiv.org/abs/1709.08481

[26] M. F. Bashir and M. A. Qadir, "Proceedings, 10th IEEE International Multitopic Conference, INMIC 2006 : 2006, 23-24 December, 2006, Islāmābād, Pakistan.," p. 506, 2006.

[27] K. Lumpur, F. Technology, and I. System, "A REQUIREMENTS TRACEABILITY TO SUPPORT CHANGE IMPACT ANALYSIS Suhaimi Ibrahim , Norbik Bashah Idris Centre For Advanced Software Engineering , Malcolm Munro Department of Computer Science , University of Durham , United Kingdom Aziz Deraman," pp. 1–12.

[28] "M. Lindvall and K. Sandahl, 1998. 'Traceability Aspects of Impacts Analysis in Object-Oriented System', Journal of Software Maintenance Research And Practice, vol. 10, pp. 37-57.".

[29] H. Dar, M. I. Lali, H. Ashraf, M. Ramzan, T. Amjad, and B. Shahzad, "A Systematic Study on Software Requirements Elicitation Techniques and Its Challenges in Mobile Application Development," *IEEE Access*, vol. 6, pp. 63859–63867, 2018, doi: 10.1109/ACCESS.2018.2874981.

[30] E. S. M, "Ingeniería Un modelo para documentar la elicitación de requisites," no. 2, pp. 1–24, 2023.

[31] I. Study and F. Shahbeklu, "Requirement Elicitation from Diverse Sources for Software Projects," no. May, 2024.

[32] "What is ATM - Definition, Uses and ATM Full Meaning." https://paytm.com/blog/atm/what-is-atm-full-form-definition-uses/ (accessed Dec. 21, 2024).