Elly anak Stephen

Software Metric for Assessing the Quality of Software Requirements

Elly anak Stephen

Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak
2020

Master of Science

2020

Software Metric for Assessing the Quality of Software Requirements

Elly anak Stephen

A thesis submitted

In fulfillment of the requirements for the degree of Master of Science

(Software Engineering)

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK
2020

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

……

Signature

Name: Elly anak Stephen

Matric No.: 15020302

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date :

ACKNOWLEDGEMENT

I would like to thank everyone that has given their contribution toward the success of my thesis. I would like to express my gratitude towards my supervisor AP. Dr. Edwin Mit for his patient, advice and guidance throughout this research.

In addition, I would like to thank both of my parents for giving me a chance and encouragement towards the completion of this research.

Furthermore, I would like to thank Universiti Malaysia Sarawak (UNIMAS) for the opportunity in giving me the job as Research Assistant and grant under FRGS/2/2014/ICT07/UNIMAS/02/1.

.

ABSTRACT

SRS document composes of FR and NFR. However, due to the heterogeneous domain development environment, the quality of the produced SRS document is in question. Software development depends on the quality of the SRS document. Poor quality of the SRS document results in poor software production. A study on the sample of the SRS document shows a lack of standardization of the document structure as well as its FR. A recent researcher manually evaluates the SRS document which is time consuming. Furthermore, the unstructured way in writing FR leads to ambiguity. A recent researcher shows that the ambiguity of natural language can be reduced by restricting the term used in writing FR. Numbers properties for each quality had been introduced to evaluate the written FR. Two methods are adopted in order to standardize the structural and FR. The first method is by comparison with the IEEE 830 ToC. The topic from the ToC is extracted and used to assess the structure of the document. The second method is the RB. It is used to standardize the way of defining the FR. The structural and FR are assessed based on four quality properties which are completeness, consistency, correctness and preciseness. The completeness quality is intended to assess the SRS structure meanwhile the others are for FR. A framework is designed based on quality properties to be assessed. The quality properties are formalized by proposing the equation for measurement. The framework is implemented and the prototype is tested to evaluate its capability. Case study is applied to the prototype. This resulted in two outputs. The first output is by comparison against the IEEE 830 which results in the percentage similarity of structure. Meanwhile, second output resulted from prototype analyst the user input. The overall measurement of the case study is produced by a prototype in the degree of percentage by adding both from the first and second outputs. From the results, the developer can evaluate the maturity of the SRS document. The resulting

measurement of the SRS structure allows the developer to reorganize the document to increase readability. The result from the measurement of FR allows the developer to ensure the consistency, validating and avoiding unspecific usage of the term used in writing the FR.

Keywords: Quantitative measurement, software quality, software requirement specification

Metrik Perisian untuk Menilai Kualiti Keperluan Perisian

ABSTRAK

Dokumen SRS mengandungi FR dan NFR. Walau bagaimanapun, disebabkan oleh persekitaran pembangunan domain yang heterogen, kualiti dokumen SRS yang dihasilkan agak meragukan. Pembangunan perisian bergantung pada kualiti dokumen SRS. Lemahnya kualiti dokumen SRS akan menghasilkan pengeluaran perisian yang lemah. Kajian sampel dokumen SRS menunjukkan kekurangan standardisasi struktur dokumen serta FR. Penyelidik baru-baru ini secara manual menilai struktur dokumen SRS yang memakan masa. Selain itu, cara yang tidak tersusun dalam penulisan FR membawa kepada ketidakpastian. Seorang penyelidik baru-baru ini menunjukkan bahawa ketidakpastian dalam bahasa tabii dapat dikurangkan dengan menyekat istilah yang digunakan dalam penulisan FR. Ciri-ciri nombor bagi setiap kualiti telah diperkenalkan untuk menilai FR yang ditulis. Dua kaedah digunakan untuk menyeragamkan struktur dan FR. Kaedah pertama adalah dengan perbandingan dengan IEEE 830 ToC. Topik dari ToC diekstraksi dan digunakan untuk menilai struktur dokumen. Kaedah kedua ialah RB. Ia digunakan untuk menyeragamkan cara mendefinisikan FR. Struktur dan FR dinilai berdasarkan empat sifat kualiti iaitu kelengkapan, konsistensi, kebenaran dan ketepatan. Kualiti kelengkapan ini bertujuan untuk menilai struktur SRS sementara yang lain adalah untuk FR. Rangka kerja direka berdasarkan sifat-sifat kualiti yang akan dinilai. Ciri-ciri kualiti diformalkan dengan mencadangkan persamaan untuk pengukuran. Rangka kerja dilaksanakan dan prototaip diuji untuk menilai keupayaannya. Kajian kes diaplikasikan ke prototaip. Dua hasil dikeluarkan. Keluaran pertama adalah hasil persamaan perbandingan struktur berbanding dengan IEEE 830. Sementara keluaran kedua dihasilkan daripada penganalisis prototaip input pengguna. Pengukuran keseluruhan kajian kes dihasilkan oleh prototaip dalam tahap

peratusan hasil pemenambahkan dari hasil pertama dan kedua. Daripada keputusan, pemaju boleh menilai kematangan dokumen SRS. Pengukuran hasil struktur SRS membenarkan pemaju menyusun semula dokumen itu untuk meningkatkan kebolehbacaan. Hasil dari pengukuran FR membolehkan pemaju memastikan konsistensi, mengesahkan dan mengelakkan penggunaan istilah yang tidak spesifik digunakan dalam penulisan.

Kata kunci: *Kualiti perisian, spesifikasi keperluan perisian, pengukuran kuantitatif*

TABLE OF CONTENTS

		Page
DECLARATION		i
ACKNOWLEDGEMENT		ii
ABSTRACT		iii
ABSTRAK		v
TABLE OF CONTENTS		vii
LIST OF TABLES		xi
LIST OF FIGURES		xii
LIST OF ABBREVIATIONS		xiv
CHAPTER 1: INTRODUCTION		1
1.1	Background	1
1.2	Problem Statement	2
1.3	Objectives	4
1.4	Hypothesis	4
1.5	Scope of Project	5
1.6	Thesis Structure	6
1.7	Significant of Project	7
1.8	Summary	8
CHAPTER 2: LITERATURE REVIEW		9

2.1 Introduction 9

2.2 Software Quality 9

2.3 Software Requirement Specification 12

2.3.1 Structural 15

2.3.2 Functional Requirement 17

2.3.3 Natural Language Processing for Requirement Analysis 23

2.3.4 Challenges of Current Approach 24

2.4 Summary 27

CHAPTER 3: RESEARCH METHODOLOGY 28

3.1 Introduction 28

3.2 Phase 1: Review on problem and solutions 28

3.3 Phase 2: Defining Problem 29

3.3.1 Propose Rule and Equation 35

3.4 Phase 3: Developing Prototype 44

3.5 Phase 4: Evaluation of Formalize Problem 47

3.6 Summary 48

CHAPTER 4: IMPLEMENTATION 50

4.1 Introduction 50

4.2 Prototype Architecture 50

4.2.1 Activity Diagram 51

4.3 Prototype Design 54

4.4 Prototype Sample Output 64

4.5 Summary 81

CHAPTER 5: DISCUSSION 82

5.1 Introduction 82

5.2 Case-control Study 82

5.2.1 Structural Assessment 83

5.2.2 Functional Requirement Assessment 86

5.3 Actual Case Study 98

5.3.1 Structural Assessment 99

5.3.2 Functional Requirement Assessment 101

5.3.3 Overall Measurement 108

5.3.4 Discussion 110

5.4 Summary 114

CHAPTER 6: CONCLUSION 115

6.1 Introduction 115

6.2 Achievement 115

6.3 Contribution 118

6.4 Limitation 119

6.5 Future Work 120

6.6 Conclusion 121

REFERENCES 124

APPENDICES 134

LIST OF TABLES

		Page
Table 2.1	Requirement Boilerplate template	18
Table 2.2	User Story vs Use Case Testing	21
Table 3.1	Frequency of propose properties to be assessed in 93 gathered SRS	32
Table 3.2	Sample assessment of FR	40
Table 3.3	Hardware minimal requirement	45
Table 4.1	Sample code sanitization structural assessment	55
Table 4.2	Example pattern-matching denotation result	56
Table 4.3	Sample code sanitization functional requirement assessment	57
Table 4.4	Example of sentence sanitization	59
Table 5.1	List of the functional requirement	86
Table 5.2	Actual result refinement functional requirement	87
Table 5.3	Possible similar functional requirement based on percentage	90
Table 5.4	Result comparison and tokenization of *[F5]* and *[F8]*	91
Table 5.5	Result comparison and tokenization of *[F7]* and *[F1]*	92
Table 5.6	Present of datatype in functional requirement	96
Table 5.7	Summary of structural measurement	100
Table 5.8	Summary of consistency properties measurement	102
Table 5.9	Summary of correctness properties measurement	103
Table 5.10	Summary of preciseness properties measurement	105

LIST OF FIGURES

		Page
Figure 2.1	Mental model of SRS document	14
Figure 2.2	IEEE 830 standard table of content	15
Figure 3.1	Research Question	29
Figure 3.2	Frequency present of IEEE 830 topic	31
Figure 3.3	Quality framework to measure SRS document	34
Figure 3.4	Component Diagram	46
Figure 4.1	Prototype activity diagram	51
Figure 4.2	Sanitization of software requirement specification document	54
Figure 4.3	Sample of pattern-matching code for structural assessment	55
Figure 4.4	Sanitization of Functional Requirement Document	57
Figure 4.5	Sample code to identify stakeholder	58
Figure 4.6	Sample code to identify verb in sentence	58
Figure 4.7	Sample code for similarity checking	60
Figure 4.8	Sample code to detect present of stakeholder	61
Figure 4.9	Sample code to detect present of possible design	61
Figure 4.10	Sample code to check the present of vague word and datatype	62
Figure 4.11	Entity relationship diagram	63
Figure 4.12	Sample of add new project	65
Figure 4.13	Sample portion SRS document contain ToC	65
Figure 4.14	Sample result of text sanitization	66
Figure 4.15	Sample result pattern-matching for structural assessment	66
Figure 4.16	Graph presentation of detected topic	67
Figure 4.17	Sample suggested synonym topic to IEEE 830 topic	67

Figure 4.18 Sample of text in FR document 69

Figure 4.19 Sample on how to add stakeholder to the prototype 69

Figure 4.20 Sample result FR refinement with similarity check 70

Figure 4.21 Sample result consistency check 71

Figure 4.22 Sample contain of Consistent Function page 71

Figure 4.23 Sample contain of Inconsistent Function page 72

Figure 4.24 Sample of test case 73

Figure 4.25 Sample input in test case 74

Figure 4.26 Sample of correctness assessment result 75

Figure 4.27 Sample of preciseness assessment result 77

Figure 4.28 Sample of datatype identification result 77

Figure 4.29 Star graph represent the overall quality of assessed SRS document 80

Figure 5.1 Sample of structural assessment and its result 83

Figure 5.2 Graph preview structural assessment result 85

Figure 5.3 Correct test case 94

Figure 5.4 Incorrect test case 95

Figure 5.5 Graph result of structural assessment 99

Figure 5.6 Graph summarization quality properties assessment for first case study 107

Figure 5.7 Graph summarization quality properties assessment for second case study 108

Figure 5.8 Summary degree of each quality assessment for first case study 109

Figure 5.9 Summary degree of each quality assessment for second case study 110

LIST OF ABBREVIATIONS

CRUD	Create, Read, Update and Delete
FR	Functional Requirement
NFR	Non-Functional Requirement
RE	Requirement Engineering
SDLC	Software Development Life Cycle
SRS	Software Requirement Specification
ToC	Table of Content

CHAPTER 1

INTRODUCTION

1.1 Background

Software engineering is a software process, which commonly consists of four fundamental activities. Those fundamental activities are software specification, software development, software validation and software evolution. This research concerns the RE process in software specifications.

RE process is including four phases which are feasibility study, requirement elicitation and analysis, requirement specification and requirement validation (Sommerville, 2016). The feasibility study focuses on the usefulness of the proposed system towards the organisation. The study included the system contribution towards the organisation, the capability to engineer the proposed system within budget and the capability of the proposed system to integrate with the others. The activities of requirement gathering, requirement classification and organisation, prioritization and negotiation and requirement documentation are the requirements of elicitation and the analysis. The requirement specification is the complete specified of the user and system specification after the requirement gathering process. While the requirement validation consists of the requirement checking process either it met the client's needs.

SRS is a documentation used to describe the user requirement and a detailed specification of the developed system. All aspects of the details must be specified before the project commerce (Pressman & Maxim, 2015). All of the detail mentions is refer to the produced SRS document. The SRS document should compose at least FR, NFR, level

prioritization of FR, target stakeholder, interface design and database design. A complete structure of the SRS document can be seen based on the topic suggested by IEEE 830 standard. All of the requirements were gathered through requirement gathering technique and they were divided into user and system requirements. Both of those requirements will be documented in the SRS.

This research concerns on the measurement of the SRS document quality. There are four qualities to be assessed; correctness, completeness, consistency and preciseness. The qualities were applied efficiently to measure the quality of SRS.

1.2 Problem Statement

Requirements gathered from the client were in a raw state and they were needed to undergo requirement elicitation process. RE is a practice of where the raw requirement is converted into useable FR. According to Zafar et al. (2018), neglecting the RE practice may impact the software development process. A survey done by Zafar et al. (2018) indicated that the main factors which can cause neglection of RE practices are limited budget, time, dedicated team and communication.

Besrour et al. (2016) created a survey to identify the challenges in the context of RE practices. The survey highlights one of the challenges which is undocumented FR and NFR, which can affect poor communication either between the developer team itself or between developer and client. Other than that, the survey is very crucial in order to collect vague and ambiguous requirements and poorly defined specifications. The usage of unstructured NL in defining the FR can cause ambiguity and poor understanding of the complex requirement.

Saito et al. (2014) and Takoshima and Aoyama (2015) highlight the issue regarding the structure of the SRS document. Those researchers arise the issue on how the FR and NFR are organized in the SRS document. Included also is the issue regarding the incapability of the current IEEE 830 standard topic to coop with heterogeneous domain in software development. However, they did express that the current IEEE 830 standard topic must be complied in order to organize the written FR and NFR.

There are two main problems that were highlighted in this research; unstructured structural of SRS and unstructured FR.

SRS document contains a number of elements as stated in IEEE 830 (IEEE Std. 830-1998, 1998). IEEE 830 standard had suggested a document structure that should be complied by the developer in order to produce a proper SRS document. The structure of the document shows the availability inside the content. Each SRS document structure may be different based on its domain study. The produced SRS document is based on the developer's understanding of the study domain. Due to no common standardisation which is compulsory to be followed, most of the documents are ill-defined, which can affect their quality.

Defining the FR is one of the main objectives of producing the SRS document. In order to write the FR, a clear specification needs to be defined. Clear specification resulted in consistency, correctness and preciseness. Commonly, FR is defined in NL. Unstructured FR is prone to inconsistency, ambiguity and incorrectness. NL is the most ambiguous model. There are numerous types of ambiguity (e.g., lexical ambiguity, syntactic ambiguity, semantic ambiguity, pragmatic ambiguity, vagueness and generality, language error ambiguity). Ambiguity causes multiple interpretations between the writer and reader of the document. This situation may affect the preciseness of the specified FR.

1.3 Objectives

The objectives of this research project are:

i. To identify the properties of the SRS document that affected the quality of the document. The properties are attribute of the quality which is needed in order to be assessed the SRS document structure and its FR. Assessment of this document's property is expected to give impacts to the quality of completeness, consistency, correctness and preciseness.

ii. To propose rules and equations based on the quality properties to be assessed. Each of the assessed quality has its own proposed properties. The completeness quality properties are proposed to assess the structure of the SRS document meanwhile the others are the FR. The rules and equations are proposed to comply with the software quality metric measurement for a quantitative study.

iii. To evaluate the proposed framework based on the software artefacts (e.g., SRS document). Based on the rules and equations, the quality framework was formulated. The framework is converted into a prototype. Train data as well as case study were used to evaluate the framework.

1.4 Hypothesis

The hypothesis for this research project are:

i. If the topic suggested in the IEEE 830 standard is followed, then the structure of the SRS document is complete and readable. The increment level of complexity in the software process can give an impact on the time for

developing the software product, meanwhile, the differences in the software production domain can cause the IEEE 830 standard to be ignored. IEEE 830 standard has been suggested as a standard guideline, which must be followed by any developer in order to generate the SRS document. As the SRS document is commonly written in NL, it is the developer's freedom to focus on their topic and ignore the standard. An automated document structure evaluation is proposed in order to ensure the developer follows the standard guideline.

ii. If the writing follows the RB template for the FR, it can reduce the ambiguity, inconsistency and incorrect conditions. One of the impacts of using NL in writing the FR is varied based on the interpretation of one individual to another. RB template restricts the usage of the word which can reduce the possible number of interpretations. It is also suggested that evaluation of the refined FR by proposing quality properties, can increase the FR quality.

1.5 Scope of Project

The scopes of the study can be simplified as below:

i. To assess the completeness structure of the SRS document based on the IEEE 830 topic as a standard guideline. 25 topics from IEEE 830 standard are extracted as a corpus to assess the document.

ii. To assess the structure of FR by restructuring it based on the RB template for the balance of its consistency, correctness and preciseness. There are some quality properties for consistency, correctness and preciseness which are

proposed and they are consisted of general properties. The proposed general properties are applied to assess the quality SRS document regardless of its domain.

iii. To develop the framework based on the assessed quality included its properties. The framework then converted into a prototype. The prototype is developed in a web-based environment. The prototype will only accept .docx format document as an input.

1.6 Thesis Structure

This thesis consists of six chapters. The first chapter is the introduction of the study, which consists of the background study, current issue with SRS document, objectives of the research, methodology used, scopes of study and the thesis structure.

The second chapter is the literature review, as it is very important in the study on the quality involve in SRS. The method used in assessing the structural and FR were extracted and evaluated. The chapter also includes the challenges of the current approach and the method which was adopted into the study.

The third chapter focuses on the proposed framework. The quality framework was proposed and discussed on the flow of the prototype, starting with the input until the production of results. The rules and equations for each quality were also proposed in this section.

The fourth chapter is about the implementation of the proposed framework. This section focuses on prototype architecture and workflow.

The fifth chapter focuses on the discussion. Two case studies were included in the prototype. The first case study is a case-control study. The input for the case-control study is predefine. The second case study involved the actual case study gather from the web site. The result from both case studies is clearly explained in the discussion section.

The last chapter is the conclusion. This conclusion section focuses on the achievement, contribution, limitation and future works.

1.7 Significant of Project

Throughout the project, there are numbers of benefits that can be expected, as stated as the following:

i. **Minimize human interference in the quality evaluation of the SRS document.** The evaluation of the structure is fully automated while as for FR, human interference is needed in certain areas.

ii. **Study on the relationship between the properties inherits by each quality assessed.** There was a study done on ISO 25010, which showed mutual influences between each of the quality assessed. Those qualities were supported by the properties proposed toward them as such as performance quality which owned the properties of time, resources and capacity. This is an advantage for the researcher to evaluate the produced SRS document based on the properties stated.

iii. **Increase client and developer acceptance.** This study only involves the developer side which means there is no involvement of clients within the assessed project. The idea is to increase the degree of acceptance from the

clients. The produced SRS document must achieve a certain degree of quality before presenting it in front of them. By using the prototype, the SRS document is evaluated into a certain degree of quality before it is submitted to the client. The result from the evaluation shows the maturity of the produced SRS document as well as an understanding of the developer toward the written FR.

iv. **Minimal time constraint**. Triple constraints; time, budget and quality are the main issues when developing a system. This prototype can minimize the time needed in order to understand the complexity of the written FR.

v. **Promote good RE practice.** RE practices are included in a crucial phase in software development. Good RE practices can produce mature documentation of FR and it is important as the quality of software production depends on it. This prototype promotes the completeness quality of the document structure. The quality of FR can refine unstructured FR into structured FR by adopting the RB template. Structured FR resulted in the quality of consistency, correctness and preciseness.

1.8 Summary

Thus, this chapter introduced the area of study which was the assessment on the quality of the SRS document. The current problem with assessing the SRS document has been identified which is the lack of standardization is the main issue. The objectives and methodology of the research are already defined. The scope of the study focuses on two elements; structural and FR. There are four qualities that were applied to assess the SRS document which is completeness, consistency, correctness and preciseness.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses the literature review on the work to assess the software quality. Included also is the SRS document attribute that is chosen to be studied. There will be four types of software quality which are completeness, consistency, correctness and preciseness. Each of the quality will be assessed based on its own properties chosen. Those quality properties will be explained in the next section.

2.2 Software Quality

Software quality is a degree to which a software product meets established requirements; however, quality depends on the degree to which those established requirements accurately represent stakeholder needs, wants and expectations (ISO/IEC/IEEE 24765, 2010). The concern of the software quality is to ensure the product fits its purpose (Sommerville, 2016).

In 1977, the McCall model was introduced with 3 main quality attribute characteristics of a software product which are product revision, product transition and product operation (McCall et al., 1977). McCall also introduces the quality metric by using the format range of value to consider the presence of quality attributes to be assessed. A year later, the Boehm model was introduced. Compared to the McCall model, the Boehm model was based on 3 quality attribute which is as-is utility, maintainability and portability.

International Organization for Standardization in 2011 introduce the ISO 9126. ISO 9126 was an evolution of software quality where it consists of 4 studies which are quality

model, external metrics, internal metrics and quality in use metrics. This ISO 9126 is a compilation study done by McCall, Boehm and other researchers in the quality software field.

The evolution quality model from the McCall model until ISO 9126 shown that some of the attributes became sub characteristic for certain attributes (Singh & Kannojia, 2013). In 2011, ISO 25010 had been proposed to replace the ISO 9126. Mutual influences between each attribute have been documented and studied. The mutual influence of eight quality attributes of ISO 25010 has been proposed in the ontology approach (Hovorushchenko & Pomorova, 2016).

There is a dilemma in software quality that should be addressed such as choosing the appropriate quality properties to be assessed and concluding the software system meets its specification. It is often quite impossible to ensure the software system created to meet its requirements due to certain factors such as usage of natural language to express the FR and improper defining level of stakeholders (Sommerville, 2016). The concern on choosing proper quality to be focused on varies between one domain to another (Su et al., 2016).

The degree of software quality is the key factor to ensure the effectiveness of using the developed software and client basic needs (Hovorushchenko & Pomorova, 2016). It might be difficult to choose the focused quality, but the relationship between each software quality allows the study to choose which quality to focus on. Even if the focused quality properties are different between each domain, a standard has been implemented to standardize the SRS (IEEE Std. 830-1998, 1998; IEEE Std. 1233, 1998) document.

One of the important qualities of software development is usability. Usability is a study which related to the satisfaction of the client toward a product or services (Abuqaddom

et al., 2019). The satisfaction of the client is based on the developer understanding of the written FR. The usability requirement is the study of converting the written FR to possible interface design. The term used in writing an FR must be specific and representable (Omoronyia & Stålhane, 2017). Studies were done by Ghazi and Glinz (2018) show the possibility of converting the written FR to possible interface design by the term that is used.

The study of this research is focused on four qualities which are completeness, consistency, correctness and preciseness. In ISO 25010, consistency and correctness quality is sub characteristic under functional suitability attribute. Preciseness quality is added to consider the presence of ambiguity during defining the FR. Meanwhile, the completeness quality is chosen due to adopting the IEEE 830 standard.

In order to evaluate the proposed quality, software quality metrics are used. According to Farbey (1990), the SRS document is best evaluated using a checklist method which is written based on the experience and current good practices. However, Farbey (1990) point of view is different from Galin (2018) which focuses on the evaluation of software quality metrics. Galin (2018) express the objective of software quality metrics is to assist management to monitor and control the development and maintenance of software system. The monitoring process can be done by controlling the data source. Galin (2018) express the importance of identification of data source in order for the improvement process.

The rule for each assessed quality in this thesis study is proposed based on its quality properties. Each quality properties act as a data source which denoted with a range of metrics. Once the weight for each property is determined, the equation will be proposed. The equation for each assessed quality act as a tool for monitoring.

2.3 Software Requirement Specification

Successfulness of software production depends on the quality of the produced SRS. Survey on RE practices shows that incomplete requirement is the main cause of software project failure (Fernández & Wagner, 2015).

There are two criteria that are needed to be considered in writing the SRS (Hull et al., 2011); the SRS must be easily readable and the set of requirements is processable. According to Aurum and Wohlin (2005), the successfulness of the documentation design is based on four factors

i. First are the knowledge and skill of a person performing analysis. Requirement analysis should be able to pinpoint the problematic requirement that causes the requirement changes in the system.

ii. Second is the availability of the documentation. This happens during the requirement gathering process where the client may have 'hidden' documentation that might impact the quality of the requirement gathered.

iii. The third is the amount of information conveyed in the documentation. This happens due to the failure of the person who wrote the document in conveying enough information.

iv. Fourth is clear and consistent documentation. Unclear or ambiguous documentation is open for interpretation.

The quality of the SRS document has been in question since the increasing complexity of software production. The complexity of software production is resulted in an increase in demand and needs by the user. According to The Standish Group report in 1994

(Standish Report, 1994), three main factors cause project failure is the lack of user input which cause by the incapability of the developer to identify target user. The second factor is the incomplete requirement which causes by ignoring the use of formal methodology such as the Waterfall model, Agile model and so on. The third factor is the changes in requirement which is caused by a lack of requirement verification during the early stage of software development.

Furthermore, according to The Standish Group report in 2010 (Standish Group, 2010), the report stresses that the success of a software project depends on the strength of user involvement. This is due to the different interpretation of user and developer on the written requirement in SRS document lead to frustration. Aside from that, proper identification of the target user will minimize the stress on both parties. Mention also in the report that tool support which helps in requirement elicitation should be considered in order to lead the requirement to mature stage. Usage of similar vocabulary by team member of the developer also lead to a mutual understanding of the written requirement specification. In short, it is beneficial for the software developer to write up the SRS document in the early phase of software development. Aside from the document became an agreement between developer and client it also can be used to measure the maturity of written FR. The maturity of the written FR will ensure the success of software development.

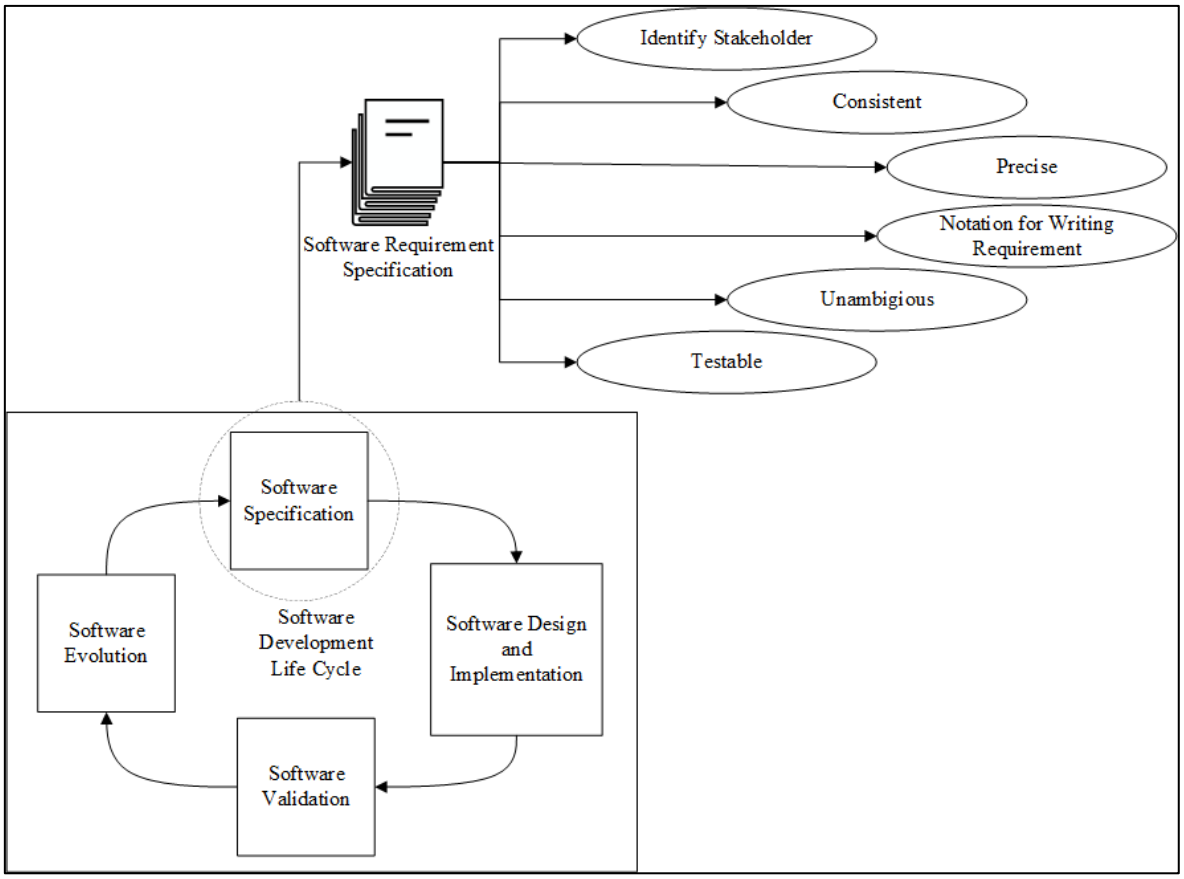


Figure 2.1: Mental model of SRS document

Figure 2.1 shows the mental model of the SRS document. According to Sommerville (2016), SDLC consists of 4 phase which are Software Specification, Software Design and Implementation, Software Validation and Software Evolution. SRS document is produced during the Software Specification phase. Mention in the mental model in Figure 2.1 is the attribute that suggested by Sommerville (2016) in order to write the FR. Those attributes are the identification of target stakeholders so that the written FR is consistent and precise. Usage of natural language in writing the FR is often ambiguous. Word use in writing the FR should be restricted. The way for writing the FR is part of the attribute stated in the mental model to counter the ambiguity in natural language. The testable attribute is to ensure the written FR is correct. Some ways in order to test the written FR are to generate test case, user story and behaviour diagram.

2.3.1 Structural

Even though the SRS document is mainly focused on the requirement specification, its structure plays an important role in organizing the content. It is crucial to ensure that the document structure is completed in an organized way to ensure the readability of the reader. A standard guideline had been proposed in IEEE 830 in 1998 (IEEE Std. 830-1998, 1998). Numerous researchers are still depending on this standard as a guideline (Georgiades & Andreou, 2010; Saito et al., 2014; Kamalrudin & Sidek, 2015; Takoshima & Aoyama, 2015; Thitisathienkul & Prompoon, 2015; Slhoub et al., 2017). The guideline in assessing the SRS document structure is based on the present of topic from ToC in IEEE 830. IEEE 830 standard is still widely used due to the similarity of structure in the IEEE 29148 standard with supported common good SRS characteristics. Figure 2.2 shows a structural guideline to build an SRS document.

1. Introduction
    1.1. Purpose
    1.2. Scope
    1.3. Definition, Acronym and Abbreviations
    1.4. References
    1.5. Overview
2. Overall Description
    2.1. Product Perspective
    2.2. Product Functions
    2.3. User Characteristics
    2.4. Constraints
    2.5. Assumptions and Dependencies
    2.6. Apportioning of Requirements
3. Specific Requirements
    3.1. External Interfaces
    3.2. Functions
    3.3. Performance Requirements
    3.4. Logical Database Requirements
    3.5. Design Constraints
    3.6. Software System Attributes
    3.7. Organizing the specific requirements
    3.8. Additional comments
4. Supporting Information
    4.1. Table of Content and Index
    4.2. Appendixes

Figure 2.2: IEEE 830 standard table of content

The completeness quality properties are used to measure the percentage of similarity between the standard structure stated in IEEE 830 with the provided SRS document (IEEE

Std. 830-1998, 1998). Completeness quality property is one of the elements in the IEEE 830 standard (IEEE Std. 830-1998, 1998). IEEE 830 table of content is used as part of the measurement.

Takoshima and Aoyama (2015) proposed the use of a translation matrix to solve the structural issue in the SRS document. This approach is done manually with the involvement of the third-party inspector. There are twenty-five topics that became a standard number of topics each assessed SRS must achieve (Takoshima & Aoyama, 2015). Thitisathienkul and Prompoon (2015) and Anil and Moiz (2017) also propose an almost similar method where the structure (eg: Completeness reflect the quality of the structure) of the SRS document can be assessed by using cross-reference method. Both of them added a few new properties toward the assessment of the SRS structure. But it limited only to the domain they assessed. The cross-reference is done by checking the presence of the topic similar which is denoted as 1 meanwhile non-present as 0. The differences between both researchers with Takoshima and Aoyama (2015) have third-party involvement in assessing the structure.

Research done in the automotive domain shows that IEEE 830 standard (IEEE Std. 830-1998, 1998) is not enough to show a complete structure for this domain (Takoshima & Aoyama, 2015). Additional quality properties may have to be implemented to accommodate the required domain. However, according to Takoshima & Aoyama (2015), the implementation of the IEEE 830 standard may become a minimal requirement that every SRS should follow.

The latest study by Asif et al. (2019) focuses only on the present important topic which are Scope, Introduction, Module, FR and Non-Functional Requirement. The topic that

had been the focus on is also a part of topic from the IEEE 830 standard even though it did not follow the standard structure by IEEE 830.

2.3.2 Functional Requirement

RE value increasingly important, due to the complexity of capturing the requirement proposed by the client. Studies done by Mund et al. (2015), show the importance of FR as a part of the SRS document. Normally, the FR is written in natural language. It is due to the written FR need to be verified by the client and became agreement between client and developer. It is proven that the usage of natural language prone to ambiguity. According to Kocerka et al. (2018), only 21% out of 100% FR are written in structured natural language. The reason for emphasizing the structured way in writing FR is an attempt to restrict the term used to construct the sentence. Those restrictions allow the developer to focus only on the defined stakeholder needs and expectations.

RB has been introduced in the SRS to overcome the complexity and ambiguity by providing the uniformity for FR (Anuar et al., 2015). RB method has been adopted by another researcher (Audytra et al., 2016). Evaluation done by Ahmed et al. (2018), proof that writing the FR in the RB template contribute toward improving the SRS quality.

According to Hull et al. (2011), improving the requirement means improving the quality of the product. RB method is adopted in the system due to the semi-formal template used. The RB promotes consistent language across the requirement due to the standardizing of the language (Hull et al., 2011). RB help in expressing the requirement in term of capability(functional) and constraint on capability(quality). Table 2.1 is the example of the RB template based on stakeholder and system as an actor (Hull et al., 2011; Ibrahim et al., 2014).

Table 2.1: Requirement Boilerplate template

Actor	Category	Requirement Boilerplate template
Stakeholder	Capability	The <*stakeholder type*> shall be able to <*capability*>.
	Constraint – Performance	The <*stakeholder type*> shall be able to <*capability*> within <*performance*> of <*event*> while <*operational condition*>.
System	Capability	The <*system*> shall be able to <*function*>.
	Constraint – Performance	The <*system*> shall <*function*> not less than <*quantity*> <*object*> while <*operational condition*>.

RB is divided into two categories. First is the capability which only addresses the FR. Meanwhile, the second is the constraint which addressed the additional detail for the FR. Initially, the capability requirement is addressed first before adding it with the constraint.

The <*capability*> and <*function*> segment from Table 2.1 is mandatory to be describe. Without it, the FR defined will be meaningless. That segment should provide information for the process verb (eg: <capability>) assist with an object (eg: <stakeholder>) which describes the segment. Library of verb vocabulary from WordNet is gathered as part of verb identification.

RB is adopted in this research due to its uniformity and the capacity of it expressing the use of the constraint. The FR defined will be easy to be tracked along with the document as well as to proceed to the next phase of development. The redundancy or conflicting requirement is easy to be trace and actors are properly defined. The advantage of having constraint ensures that define FR is more precise and concise.

The first quality assessment for the FR is consistency quality. Consistency is defined as the degree of uniformity, standardization and freedom from contradiction among the

documents or parts of a system or component (ISO/IEC/IEEE 24765, 2010). The consistency properties measure to the non-similarity of each defined FR assist with the presence of the stakeholder (Wiegers & Beatty, 2013). Each of the FR must be unique from one another (Laplante, 2007; Pohl & Rupp, 2015).

Identifying the goal of stakeholders has become increasingly complex due to the diverse range of backgrounds (Aurum & Wohlin, 2005). Inconsistency happens due to various ways of stakeholders to express themselves (Aurum & Wohlin, 2005). Understanding multiple stakeholder viewpoints is part of requirement elicitation in order to ensure consistency. Identifying stakeholders is one of the important elements in software development (Audytra et al., 2016; Ali et al. 2018; Ferreira et al. 2018). It is important to completely understand and clarify each of the stakeholder involvement in the development (Maguire & Bevan, 2002). Ali et al. (2018) proposed 4W's questions (Who, Where, When and Why) ontology approach to assessing the FR. The first question 'Who' show the importance to indicate the correct and consistent term for stakeholder.

Assessing the <*capability*> segment depends on the technique used. The ontology approach by Ali et al. (2018) represents the 4W's question on the purpose of the identified stakeholder. Meanwhile, Ferreira et al. (2018) create an empathy profile of identified stakeholders in order to identify requirements based on assessed stakeholder needs and expectations. Audytra et al. (2016) proposed the use of the Levenstein method in order to assess the FR. The similarity of the FR is assessed based on the word order similarity. Audytra et al. (2016) divide the FR sentence into a segment based on RB. Each of the segment is denoted with range zero to twenty-five. The total up of all segments is one hundred. Assessed FR sentence split up and fill into the respective segment. Levenstein

method applied to increase the reliability of the similarity check between segmentation. Ali et al. (2018) express that the consistency calculation for each FR is based on the acceptance of the client during the meeting. Accepted FR will be denoted as one otherwise zero.

Correctness quality is defined as the capability to meet satisfactory needs (Sarmiento et al., 2015). There are few techniques to measure the correctness of the FR. One of them is by using the requirement validation technique (Harmon & Youngblood, 2008; Sommerville, 2016). Requirement validation is the process of checking the requirement based on customer needs.

A survey done by Kocerka et al. (2018) shows that more than 60% of companies prefer to model and test the FR. Those include the use of CASE and test case generation. Test case generation is one of the required validation techniques (Sommerville, 2016). The requirement analyst should be able to come out with the test cases based on the defined FR in the first place. In other words, if the test case cannot be developed, then the FR is considered as ambiguous (Wiegers & Beatty, 2013). The generation of the test case can minimize the uncertainty in assessing the written FR (Zhang et al., 2018).

Use case testing is part of the test case generation. This technique helps to identify test cases where it exercises transaction by transaction execution (Graham et al., 2006). Use case testing is more challenging for users compares to the user story. However, use case testing provides more organized information. Studies were done by Medeiros et al., (2016), showing that the SRS document developed based on the user story is brief, vague, ambiguous and insufficient for capturing the complexities of the up-front design. The example of the technicality level of a user story and test case can be seen in Table 2.2. (Hull et al., 2011; Ibrahim et al., 2014).

Table 2.2: User Story vs Use Case Testing

Validation Technique	Sample
User Story	As a user, I can click on viewed picture so that I can view picture info.
Use Case Testing (Normal Flow)	1. The user view the picture. 2. The user click on the picture. 3. The system search the database based on the picture id. 4. The system view the picture info in model-dialog.

User story is user-friendly towards clients where it is much easier to be understood. Use case testing is much more technical compared to the user story. The technical level of language to define use case testing may be hard to be interpreted by the client. Aside from that, the technicality of use case testing allows a better understanding of the FR. Those resulted in the written FR much reliable and easier to be trace. The use case testing consists of few properties. The properties of use case testing are discussed further in Chapter 3.

Ahmad et al. (2018) indicate a survey done to evaluate the correctness of the test case generated. Likert Scale analysis used by denotation of range between zero to five. There is no clear indication that there is exist an internal measurement used by the researcher to evaluate the generated test case.

The third quality to be assessed is preciseness quality. Preciseness properties are described to measure the presence of the vague word and datatype in each of the FR. If the requirements do not reflect the needs of the client precisely; thus, allow for several interpretations on the system, then, the result is often a system that does not meet the expectations of the client or the users (Pohl & Rupp, 2015; Bhatia et al., 2016).

Usage of natural language to define the FR had cause several interpretations. To meet the client's needs, the FR must be precise without any vague details (Ali, 2006). According to Saito et al. (2014) and Thitisathienkul and Prompoon (2015), there are numbers of characteristic or properties which lead to the ambiguity of the FR. One of the criteria for requirement writing stress on avoiding the usage of the vague word (Hull et al., 2011; Bano, 2015; Kocerka et al. 2018). Saito et al. (2014) listed several criteria in order to assess the FR. One of the criteria that are adopted into this research which is the presence of the non-equivocal word. Non-equivocal represents the term that is used without any further explanation. Additional information to define the term used is required to increase the preciseness of the FR. The additional information is as such as defining the required data type or even the invariant value (Omoronyia & Stålhane, 2017).

Possible vague words had been discussed as a word that may have either general or ambiguous meaning (Bhatia et al., 2016). An example is the following sentence:

I am coming to the bank.

The word '*bank*' can have more than one interpretation, as in riverbank or money bank. Even the word '*coming*' also has several interpretations which did not indicate the actual time of arrival at the bank.

There are six types of ambiguities which are lexical ambiguity, syntactic ambiguity, semantic ambiguity, pragmatic ambiguity, vagueness and generality and language error ambiguity (Bhatia et al., 2016). Thitisathienkul and Prompoon (2015), Bhatia et al. (2016) and Sabriye and Zainon (2017) raised the issue of language error ambiguity in assessing the FR. Thitisathienkul and Prompoon (2015) evaluate the ambiguity by denoting it as one if ambiguity is present otherwise zero. Bhatia et al. (2016) introduce an ontology approach to

assess the ambiguity through the reasoning phase. Sabriye and Zainon (2017) evaluate the ambiguity of the sentence by counting the number of occurrences and divide it with the number of sentences.

2.3.3 Natural Language Processing for Requirement Analysis

This section discusses the usage of NLP in order to analyst the FR. NLP is used to extract information from the raw data (Nazir et al., 2017). According to Riaz and Butt (2019), there are three kinds of information that can be extracted from the SRS document; requirement sentence with tags, repeated requirement and conflict between requirement. There are many techniques in order to extract information from the raw data which are tokenization, POS tagger, text chunking, parsing, VSM and TF-IDF.

NLP is used in this thesis to automate the extraction element of concern from raw data. The raw data in this research refer to the inputted structure of the SRS document as well as its FR. Before the information gathered from the raw data, it must split into the individual sentence. That process is known as sentence boundary detection. A similar process is used by Kuchta and Padhiyar, 2018. The boundary between each sentence needs to be declared and it is totally based on the text editor used. Then only the information from the sentence can be gathered. The next NLP process is to tokenize the sentence into individual word and run pattern-matching with the knowledge-based. Riaz and Butt (2019) acknowledge knowledge-based as an approach to be used in the pattern-matching process because it may contain a large set of informational data.

2.3.4 Challenges of Current Approach

This section discusses the approach used by the researcher to overcome the issue regarding the measurement quality of the SRS document. The discussion will only focus on the structure of the SRS document and its FR as mention in the scope of this thesis.

Assessment of completeness quality may be classified into the structure of the SRS document. The challenge of the current approach in assessing the structure of the SRS document is the lack of automation. The translation matrix used by Takoshima and Aoyama (2015) indicates the lack of automation in locating the similarity of the IEEE 830 table of content topic.

Thitisathienkul and Prompoon (2015) and Anil and Moiz (2017) added a new topic to compliment the domain they assessed on. The evaluation of the structure is similar to the translation matrix used by Takoshima and Aoyama (2015) which also lack of automation. Asif et al. (2019) focus only on certain topics which are scope, introduction, module, FR and non-functional requirements. The evaluation is based on the presence of the topic indicated. However, due to Asif et al. (2019) only focus on a certain topic, it can be concluded that the completeness of the SRS document structure will not be achieved.

Research done by Takoshima and Aoyama (2015), Thitisathienkul and Prompoon (2015), Anil and Moiz (2017) and Asif et al. (2019), did not indicate any tool to automate the evaluation of the SRS document structure. However, Takoshima and Aoyama (2015), Thitisathienkul and Prompoon (2015) and Anil and Moiz (2017) did come out with a framework that indicates the measurement of the SRS document structure need to be based on topic suggested in the IEEE 830 standard.

Based on the argument above, the automated measurement process is proposed to evaluate the structure of the SRS document. Those resulted in the increased reliability of the result. An additional advantage of automation is it minimizes the involvement of a third-party inspector which is costly and time-consuming.

The usage of natural language without restriction in defining the FR is the main cause of ambiguity. The current approach in writing the FR shows a lack of restriction on the usage of natural language (Kocerka et al., 2018). Adoption of the RB template help in restricting the usage of natural language.

RB can be categorized into two which are <*stakeholder*> and <*capability*> segment. Stakeholder involvement needs to be clearly defined. There is no clear indication by the researcher on how the stakeholder is defined or captured (Ali et al., 2018; Ferreira et al., 2018). Due to varies levels of stakeholders, it is hard to properly identify which stakeholder to focus on. However, the percentage of defined stakeholders in each FR can be calculated to identify which stakeholder to focus on.

The <*capability*> segment consists of the role of each defined FR. The current approach to measuring the similarity of written FR is used by Audytra et al. (2016) which are the Levenstein method. Levenstein's method study on the degree of transformation needed to transform one word into another. However, the reliability of the result can be increased by the adoption of a similar_text() method which allows the comparison in terms of character compared to the Levenstein method.

Identification of the vague word decreases the ambiguity of the define FR. According to Kocerka et al. (2018) and Osman and Zaharin (2018), a number of vague words are collected and used as part of a library to detect the presence of a vague word in the written

FR. However, identification of the vague word in FR is not enough. It lacks a specific term which causes the written FR meaningful.

According to Omoronyia and Stålhane (2017), the term used in the FR must be specific and representable. Indication present of a vague word in written FR is not enough to conclude the assessed FR is ambiguous. The term used in writing the FR also should be considered to ensure the written FR are meaningful. This led to the proposal of specific terms used in order to express the FR. A list of the term represents possible datatype and graphical interfaces are collected and stored in knowledge-based. A pattern-matching technique used those lists of term to assess the FR.

Systematic literature was done by Anuar et al. (2015) shown the importance of designing the test case as early as the requirement phase. Graham et al. (2006) also proposed the use case testing which will be adopted in the study. Ahmad et al. (2018) adopted one of the test case generations which is use case testing method. However, his study only focused on the presence of the identified stakeholder. Furthermore, is no clear indication that each of the written FR should have at least 1 test case. Wiegers and Beatty (2013) express the importance of generating the test case as early as the requirement phase. Ansari et al. (2017) did attempt to automate the generation of test case. However, there is still a limitation which it only can be used with the presence of a conjunctive statement. The validation of each written FR can be improved by calculating the number of valid test case generated by the developer. The generated use case testing can be improved by checking the similarity of stakeholder stated in FR with the stakeholder indicated in the '*Normal Flow*' element.

Shah et al. (2017) evaluate existed tools that are used to evaluate the requirement specification. Twenty tools had been evaluated. Out of twenty, thirteen of the tools indicated

are paid licenses. However, the differences in the development platform for the tool resulted in difficulty for integration into a custom prototype.

2.4 Summary

Thus far, the chapter discussed on the related literature review. Based on the literature, all intended qualities are assessed. There are four qualities to be assessed: completeness, consistency, correctness and preciseness. All properties intended to assess each quality are discussed. First, the completeness quality is intended to assess the structure of the SRS document meanwhile others are for FR. Completeness quality properties are the assessment of the IEEE 830 topic in the table of content. Second, the consistency quality properties are the presence of relevant stakeholders and the assessment of similarity between each FR. Third, the correctness quality properties are intended to assess the use case testing *'Normal Flow'* element. The similarity of actor and term representing the possible interface design are the properties that must exist in order to assess the *'Normal Flow'* element of test case. Lastly, the preciseness quality properties are intended to assess the presence of vague words and term representing the possible datatype. Based on the properties for each quality mention, the rule will be proposed. Then the rule converted into an equation. A prototype is built to relive the equation. The end result generated by the prototype shows the maturity of the produced SRS document.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter discusses the research methodology process. This research methodology will be carried out in four phases. The first phase is a review of problems and solutions. The second phase is defining the problem. The third phase is developing a prototype. The last phase is the evaluation of formalizes problem.

3.2 Phase 1: Review on problem and solutions

There are three steps done in this phase. The first step focuses on learning the impact of SRS document in software production. The study focuses on learning which quality properties that should be inherited by the SRS document. The revolution of the software quality model indicates the presence of mutual influences between each of the quality. Numbers of the element in the SRS document are identified, to understand which element delivers the most impact on the quality assessment. This study will only focus on two elements of the SRS document which are the structure and FR.

The second step is to study the proposed solution. Literature review done in Chapter 2 indicate certain criteria must be considered in order to create an SRS document. Those criteria reflect the quality of the created SRS document. A number of solutions had been proposed by the researcher in order to solve the issue structure of the SRS document and FR.

The third step is to identify the solutions patterns for the problems. Two methods had been adopted as part of the solution. First is the similarity check with the topic from ToC of the IEEE 830 standard. Second is the refinement of written FR based on RB which is a semi-

formal template. The topic gathers from ToC of the IEEE 830 standard will be used to assess the completeness of the SRS document structure. Meanwhile, the RB template adopted to refine the original written FR. Refined FR will be assessed by the properties' quality of consistency, correctness and preciseness.

3.3 Phase 2: Defining Problem

This phase can be divided into four steps: Develop architecture for the problem, Determine the relationships between problems and solutions, Formalize the quality properties and Propose framework.

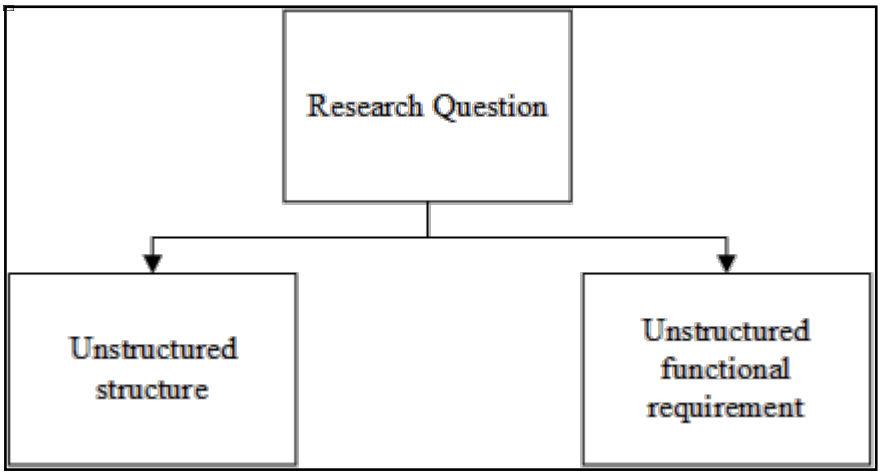


Figure 3.1: Research Question

Figure 3.1 shows the research question that will be the focus on. In short, there are two problems in SRS to be the focus on. First is the unstructured SRS document structure. Second is the unstructured FR. Based on the literature review done in Chapter 2, the impact of the SRS document on software production is high. The present of the complete SRS document structure is important where it impacts the completeness as well as readability of the produced document. Aside from that, the common understanding of one developer to another also can be increased. The benefit of structured FR had been discussed in Chapter 2. In short, structured FR benefits the developer in terms of understanding the written FR.

Further studies had been done to prove the presence of the problem as stated in Figure 3.1. A total of 93 SRS documents are downloaded from the available website and divided into nineteen domains (Appendix A). The process to gather an SRS document is quite simple. For example, in the search engine website type ‘sample srs document for e-commerce website’. Then the result of searching will appear. The SRS document can be downloaded from the list which resulted in 93 SRS documents. All of the collected SRS document is gathered using the same method. The collected SRS document then classified into its domains which are account, administration, agriculture, analysis, communication, education, entertainment, geolocation, management, marketing, medical, mobile application, multimodal, network, program tool, safety, society, software development and transportation. However, the classification of the SRS document based on its domain does not impact the studies. This is due to the evaluation of the SRS document that will not consider which domain it belongs to.

The relationships between problems and solutions are defined to understand the contribution between both of them. The challenges of the current approach had been discussed in Chapter 2 indicate a clear relationship between the problem in the SRS document with the assessed quality. Discuss also is the quality properties that are inherited by the quality to be assessed. Some of the property is adopted in the study and some are proposed in order to increase the reliability of the SRS document quality measurement.

The quality properties of each quality to be assessed can be summarized by; Completeness quality adopt topic suggested in IEEE 830 as standard to assess the structure of the SRS document. Consistency quality assesses the refine FR by proposing the use of a similar_text() method instead of the Levenstein method. The correctness quality of refined

FR is based on the validity of the generated test case. Lastly, the preciseness quality of refined FR is based on the presence of term represent possible datatype and non-present of a term representing the vague word.

The assessment quality of the SRS document is divided into two which are structure and FR. The result within the assessment will not impact each other. However, in the FR assessment, correctness and preciseness quality depend on the result produced by the prototype. The final confirmation on which assessed FR are consistent need to be done by the user. Those final conformations will impact the total number of FR used by the equation for correctness and preciseness quality. Due to the presence of a connection between each quality throughout its properties, it promotes traceability among the quality. In this assessment, the correctness and preciseness quality depends on consistency quality. The number of conformed non-similar FR by the <*user*> for the consistency assessment impact the number of FR to be assessed during the assessment for correctness and preciseness quality.

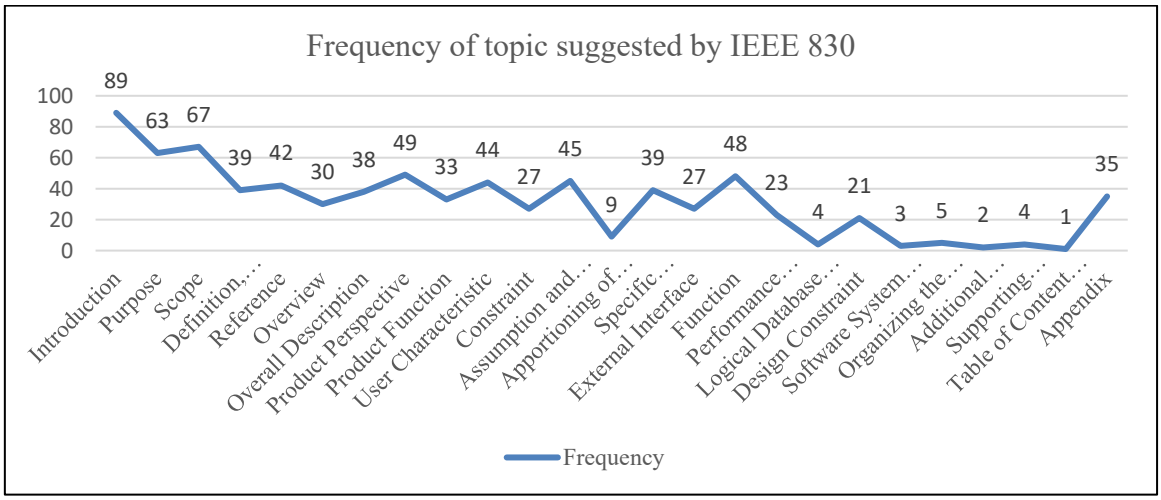


Figure 3.2: Frequency present of IEEE 830 topic

From the data tabulated in Figure 3.2, it shows a range of frequency from low until high for each topic. This is due to the different ways of organizing the SRS document. Each

of the developers from different companies and backgrounds may have a different way of setting up the SRS document. Variety frequency shows a lack of standard topic as suggested in IEEE 830 had been followed. The study is concerned about following the number of topics suggested in the IEEE 830 standard.

Table 3.1: Frequency of propose properties to be assessed in 93 gathered SRS

	Test Case	**Stakeholder**	**Datatype**	**Interface**	**Functional Requirement (FR)**	**Semi-formal FR**
Frequency	23	89	30	35	86	21

From the data tabulated in Table 3.1, it shows the frequency occurrence of each proposed quality properties to be assessed from ninety-three SRS document. Table 3.1 is composed of property that exists in the SRS document which are test case, stakeholder, datatype, interface, FR and semi-formal FR. Twenty-three SRS had implemented test case to further validate their proposed FR. Assessment of the FR based on test case shows the understanding of the developer on each function. This study had been further discussed in previous Chapter 2. High frequency on the stakeholder and FR property show the awareness of SRS document author toward the importance of defining target stakeholders for each of FR.

Meanwhile, semi-formal format represents the way of writing FR. Even though the frequency is as low as twenty-one, it shows the relevance of implementing a semi-formal format to define FR. The benefit of using the semi-formal format to define the FR also has been discussed in previous chapters.

Adopting the use case specification help in raising the number of possible interface design. Shown in Table 3.1, the frequency sketch design of the interface is thirty-five. The presence of datatype in the SRS document is also shown in Table 3.1. The other properties that had not been assessed in the collected SRS document are presence of vague words and redundancy. This is due to the possibility of FR to be redundant and the presence of vague words is based on the developer and client interpretation toward the written FR.

Each of the quality properties is then formalized. Each of the properties assesses in the previous step is assigned based on its qualities. The formalization of the quality properties is by assigning it with a range of zero to one based on its presence. Zero represents not present otherwise one represents present. The method used is called as Likert Scale method. The usage of the Likert Scale method in this study allows the conversion of a qualitative study to a quantitative study.

The final step in this phase is proposing a framework. A quality framework is proposed to better visualize each of the quality to be assessed. Rules are proposed for each of the assessed quality. Rules are proposed based on the literature review done and properties on the second step. Finally, the equation is proposed based on the proposed rule to measure the quality of the assessed SRS document. The mean equation adopted as a measurement. It is due to the mean is easy to be calculated. The final result of the calculation will be in percentage form to ease the prototype user to evaluate the quality of the assessed SRS document.

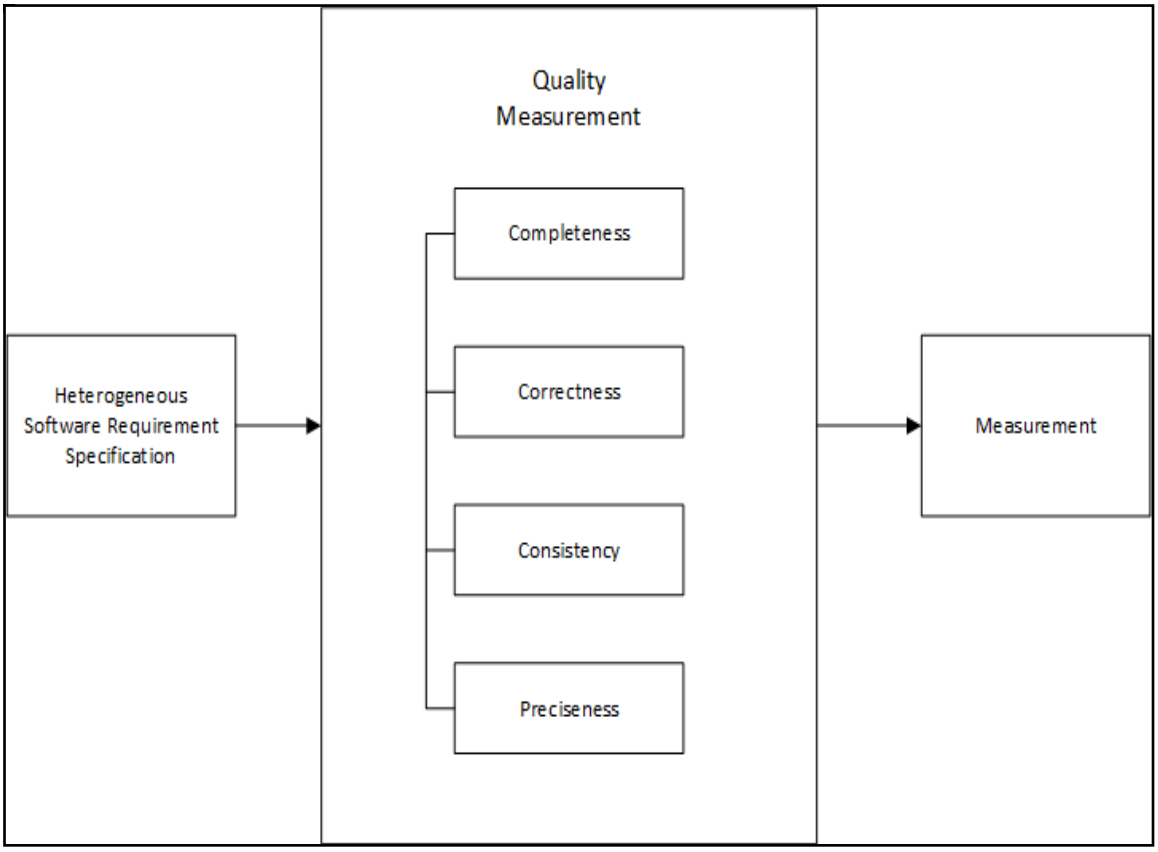


Figure 3.3: Quality framework to measure SRS document

Figure 3.3 shows the quality framework to measure the quality of the assessed SRS document. The term heterogeneous refers to diversity or dissimilarity. Heterogenous in SRS refers to the varieties domain of SRS. Based on Figure 3.3, the domain of the assessed SRS document will not give any impact on the measurement. The quality properties used to assess the SRS document has been chosen properly to ensure the integrity of the measurement.

Mention early in Chapter 1, the studies will focus on the quality measurement of the SRS document. There are numbers element in the SRS document but the studies focus on the document structure and its FR. As for assessed quality, there are 4 qualities to be the focus on which are completeness, consistency, correctness and preciseness. The quality properties of completeness will be used to assess the structure of the SRS document meanwhile other qualities will be used to assess the FR. Furthermore, each of the assessed

quality will inherit its own properties. A rule will be proposed for each of the quality to ensure the dependency of the quality properties.

In order to measure the quality of the SRS document, a mathematical mean equation is adopted. The equation for the measurement is generated based on the proposed rule. This means that each of the rules will have its own equation. The equation will be implemented in the prototype to automate the process of measurement. The proposal of rules and equations will be discussed further in the next subsection.

3.3.1 Propose Rule and Equation

This subsection discusses the proposed rule and equation in order to measure the quality of the SRS document. There are 4 qualities used to assess the SRS document. The completeness quality properties will be used to assess the structure of the SRS document meanwhile correctness, consistency and preciseness quality properties are used to assess the FR. As discussed in the previous subsection, each of the quality will have its own properties. The properties of the quality are chosen carefully based on their capability to assess the heterogeneous SRS domain. The rule for each quality is proposed based on the properties to assess which had been discussed in the previous subsection. The rule is proposed to control the property. The property is controlled by assigning to correspond assessed quality. Furthermore, the rule proposed to shows how the property works with each other in harmony. Once the rules are proposed, it is converted into an equation that can be implemented through the prototype.

The quality properties proposed to assess the completeness is the topic gathered from IEEE 830 standard. IEEE 830 standard is still relevant until today as many recent researchers still use it as a guideline. IEEE 830 standard suggested an outline of structure in order to

produce an SRS document. This research only adopts level one and level two of the topic as suggested by IEEE 830. Thus, the sum of topic level one and two amounts to twenty-five topics. The overview of those twenty-five topics is shown in Figure 2.2. It will be counted as a constant and minimal requirement that should be met by the SRS provided by the user. By using the IEEE 830 structure as a baseline; this will provide essential needs for each of the SRS to achieve their standard or quality.

However, 25 topics propose in the IEEE 830 table of content is not enough to evaluate the SRS document. This is due to numerous ways to represent the topic itself. To compensate for the various numbers of synonym topics, knowledge-based is built. There are two sets of operations done in order to assess the SRS document structure. The first is based on the IEEE 830 topics in the table of content. The second is based on the synonym topic. The SRS document inputted will be assessed and compared with the library of topics gathered from the IEEE 830 table of content. If it happens where the second operation needs to be activated, those SRS document will be assessed by the synonym library to search for similarity.

The idea is to run pattern-matching between the topic gathered from the IEEE 830 standard with the SRS provided by the user. Any matched topic is marked and counted for. The synonym library is also created as each of the SRS provided may have a different structure. The existence of the library will increase the reliability of the prototype as well as possible detection.

The result of the similarity is presented in table and graph form. The detected topic will be displayed as well as the possible synonym topic. This allows the user to amend provided SRS accordingly to increase the percentage of completeness.

A rule is proposed to formalize the assessment of the SRS structure. In this research, the completeness rule is proposed as:

$$(((\ T_n \cap L_t\ ) \equiv \text{Same}) \cup A_t\ ) \equiv (\ T_n \equiv \text{Complete}\ ) \qquad \text{Equation 3.1}$$

T_n represents the topic being assessed, L_t represents a list of topics gathered from IEEE 830 table of content and A_t represents the added list of topic from the tested table of content. From the rule, it clearly states that to ensure the structure is complete, the topic that is currently assessed and the list of topics from the IEEE 830 table of content must be the same or the topic is from the added list then the assessed topic is considered complete.

Measurement to evaluate the completeness properties, an equation is proposed. The equation to measure the structure of the SRS can be seen below:

$$S = \left(\frac{\sum M_t}{C_1}\right) * 100\% \qquad \text{Equation 3.2}$$

The proposed measurement of SRS structure, where *S* represents the degree of completeness of structure, M_t represent the total number of matched topics and C_1 represents a constant of 25. The constant of 25 represents 25 numbers of topics in the structure in IEEE 830. So, it can be said that a minimal number for each structure in the SRS document is 25 and the standard shall be followed by any SRS document.

Moving on to the FR element, it will be assessed by 3 quality. Similar to the completeness quality, quality properties for consistency, correctness and preciseness also will be proposed. In general, the RB template is adopted. RB template promotes the structured way of writing the FR. It also promotes consistency in writing the FR. Ambiguity also can be reduced due to restriction in usage of the term during writing the FR.

Template according to RB formatted as:

The <*stakeholder*> should be able <*additional information*>.

The template above will be used as a standard to refine the inputted SRS document. Data cleaning will be done before the refinement. Further explanations of data cleaning will be discussed in Chapter 4.

However, the refinement of raw FR to structured FR based on the RB template does not guarantee that the written FR is unambiguous and consistent. The RB did help in restricting the usage of the word in order to write FR. Yet, the word used in order to fill the <stakeholder> and <additional information> segment is not restricted. The writer still had freedom in terms of choosing a word in order to write an FR. Despite only depending on RB to refine the raw FR, a further assessment must be done. The property had been choose in order to assess the refined FR. The property is chosen carefully based on the suggestion by other researchers which had been discussed in Chapter 2.

Once the unstructured FR refine into written into structured FR based on RB, the assessment of consistency, correctness and preciseness quality may follow. A FR is considered consistent if it does not conflict with each other. Based on the RB, the FR must consist of stakeholder and the role that correspond to the stakeholder. The conflict in the role can be avoided by eliminating the possible duplication of the FR. This is possible to be accomplished by comparing each of the FR itself. A checker is proposed to check the FR similarity in terms of per character but not the meaning itself. Here, the <*user*> needs to confirm each of the FR individually.

Redundancy may occur due to the improper definition of stakeholder. The possibility of having multiple stakeholders with the same level of authority is high. This is due to the use of similar or synonym word which represents the same stakeholder. For example, the user may be called a client, customer or consumer and so on. The <*user*> needs to define its own level of stakeholder. The defined stakeholder will become the knowledge repository for specified uploaded SRS. By combining the technique of stakeholder and proper role function identification, the number of possible conflict requirement may be reduced which lead toward consistency.

In this research new consistency rules are proposed as:

Actor: Stakeholder

$$((H_n \cap R_n) \cap (H_{nl} \cap R_{n+1})) \equiv (F_n \equiv \text{Consistent}) \quad \text{Equation 3.3}$$

Actor: System

$$((Q_1 \cap R_n) \cap (Q_1 \cap R_{n+1})) \equiv (F_n \equiv \text{Consistent}) \quad \text{Equation 3.4}$$

Based on the rules (Equation 3.3) and (Equation 3.4), H_n represents the stakeholder in the function that is being assessed, R_n represents the role of the function being assessed, H_{nl} represents other function with a similar stakeholder or not, R_{n+1} represents other roles of the function, Q_1 represent the system and F_n represents the function being assessed now.

From the logical rule, there are two rules stated which defined the stakeholder and the system. For the stakeholder (Equation 3.3), even though the role in other functions of similar stakeholders is not the same during the assessment, it is considered consistent. It also applied the same rule for the system, but it only represents itself compared to the various type of stakeholders (Equation 3.4). A useful example of consistency is shown in Table 3.2.

Table 3.2: Sample assessment of FR

No	Functional Requirement	Stakeholder	Additional Information
FR1	The admin should be able to manage customer detail.	Admin	To manage customer detail
FR2	The superadmin should be able to manage customer detail.	Superadmin	To manage customer detail

Based on Table 3.4, *FR1* and *FR2* have different stakeholders. The first properties which existence of stakeholder is fulfilled. Regarding the additional information, the prototype is designed to calculate the percentage similarity of word usage in a sentence. The user of the prototype is responsible to confirm the FR based on the percentage given. Once the user confirms the FR, only then it is consistent.

Meanwhile, the proposed measurement for consistency is as follow:

$$T = \left(\frac{\sum B_t}{\sum F_t}\right) * 100\% \qquad \text{Equation 3.5}$$

The proposed measurement of the consistency of FR (Equation 3.5), T represents consistency, B_t represents the total number of consistent function and F_t represents the total number of FR available. The mean is calculated to measure the degree of consistency of the FR. To measure the mean, the total number of functions that are consistent is divided with the total number of functions in FR.

The next assessment is the correctness quality of the FR. The assessment of correctness is started once the conformation of FR is done. Validation of FR contributes toward the correctness quality. Once the consistency process is done, the assessment of

correctness can be started. The validation technique used is identifying the presence of the test case.

The capability of representing the test case in as early as the requirement phase is to test the knowledge of the requirement analyst to come out with a suitable test case. Test case is important to be generated as early as the requirement phase in order to reduce the number of disputes with the customer. Aside from that, this verification process allows the detection of possible errors that might occur.

Test case has elements which are *'Use Case Name'*, *'Actor'*, *'Precondition'*, *'Normal Flow'*, *'Alternative Flow'* and *'Postcondition'*. Out of the six elements, the study will focus on the presence of three elements which are the *'Use Case Name'*, *'Actor'* and *'Normal Flow'*. The reason for selecting those three elements is to validate the input in '*Normal Flow'* element. According to Samah (2011), the validity of test case is depending on the input in '*Normal Flow*' element. '*Use Case Name*' and '*Actor*' element are adopted in the study to restrict the way of writing '*Normal Flow*' element. '*Use Case Name*' element restricts the writer to write the corresponding test case based on the test case name. Meanwhile, the '*Actor*' element helps the writer to focus on writing the test case on the targeted actor. The other element aside from those three elements is considered as an optional element that can be ignored. In order to assess the correctness of FR, those focused elements must exist. First, the test case must have a target that is to assess which is the FR. Second, the stakeholder must be named. The prototype's targeted stakeholder is automatically detected by the prototype based on the stakeholder defined by the user. The third is the *'Normal Flow'*. Each of the test case must be properly defined step-by-step in order to validate the FR.

Test case is chosen as a property to measure the correctness quality due to its capability to validate the requirement in the early phase of software development. Based on the properties to be assessed in the test case, the correctness rule is defined (Equation 3.6).

$$(U_n \cap A_n \cap N_n) \equiv (F_n \equiv \text{Correct})$$ Equation 3.6

U_n represents the use case name, A_n represents the actor, N_n represents the normal flow and F_n represents the function being assessed. From the rule, it clearly states that to ensure the function requirement is correct, the use case name, actor and normal flow must exist. To ensure the normal flow is not being manipulated, the presented stakeholder in the actor section will be compared with the normal flow.

Based on the proposed rule, the measurement to assess the properties of correctness is proposed as:

$$V = \left(\frac{\sum \frac{U_n + A_n + N_n}{C_3}}{\sum F_t} \right) * 100\%$$ Equation 3.7

The proposed measurement of the correctness of FR (Equation 3.7), *V* represents the correctness of all FR, U_n represents the existent of use case name, A_n represents the existent of actor, N_n represents the existence of '*Normal Flow*', C_3 represents constant of 3 and F_t represent the total number of conformed non-similar function in FR. The presence of the '*Use Case Name*', '*Actor*' and '*Normal Flow*' affect the calculation for each FR. The sum of all functions will be divided with the sum of total function plus the sum of detected use case name, actor and normal flow to gather the mean value.

A vast collection of FR in a single SRS may cause the data to be imprecise. This is due to the usage of natural language to define the FR. By using the RB as a template, the

uniformity and integrity of each FR can be sustained. It is difficult but not impossible to identify the datatype from the requirement specification, which is defined by using natural language. Therefore, in this study, the suggested datatype will be formulated based on the intrinsic nature of the term, in conjunction with the knowledge repository. Knowledge repository for the datatype is created based on the possible synonyms with the corresponding datatype. The design stage will be much easier if the whole datatype of the FR is stated.

A list of the vague words is gathered for this research to identify the possibility of it being used in writing FR (Thitisathienkul & Prompoon, 2015). There are 138 words that had been identified as vague words and may promote towards ambiguity of sentences assessed (Silke, 2012). The identification of vague word and datatype are expected to increase the preciseness of written FR.

A rule had been proposed based on the situation, where it will be implemented to measure the preciseness of each function in FR. A constant of 2 had been implemented in order to propose a measurement for preciseness quality. The constant of 2 represents the two types of detail collected which are the datatype and vague word. Each detail type is denoted as constant of 1 if found otherwise 0 if not found. Calculation of preciseness for each function depends on the presence of the data type and vague word. In this research, a new preciseness rule is proposed as:

$$(D_t \cap \neg V_w) \equiv (F_n \equiv \text{Precise}) \qquad \text{Equation 3.8}$$

Based on the rule (Equation 3.8), D_t represents data type, V_w represent the vague word and F_n represents the function in the FR that is assessed. From the rule, it is stated that the assessed FR is precise if and only if the data type is presented and none of the vague words are detected.

The proposed measurement of the preciseness of FR (Equation 3.9), *P* represents the preciseness of all FR, D_t represents the existent of data type, V_w represents the existent of vague value, C_2 represents constant of 2 and F_t represents the total number of functions in FR. The constant of 2 represents the presence of possible datatype and vague word. The sum of all functions will be divided by the sum of total function plus the sum of detected vague word, in order to gather the mean value.

$$P = \left(\frac{\sum \frac{D_t + V_w}{C_2}}{\sum F_t} \right) * 100\% \qquad \text{Equation 3.9}$$

The overall quality of SRS can be measured through the result is gathered from assessing structural and FR. The proposed measurement to measure the SRS based on the assessments can be seen below as overall quality (Equation 3.10). For the overall quality, S represents the overall consistency, U represents the overall correctness, P represents the overall preciseness and T represents the overall consistency.

$$Overall\ Quality = \left(\frac{S + U + P + T}{4} \right) \qquad \text{Equation 3.10}$$

3.4 Phase 3: Developing Prototype

In order to develop the prototype, there are two steps to be followed: Implement the framework on the web-based application and Testing the functionalities of the prototype.

The prototype will be built based on the web-based development language. This is due to the ease of it to be implemented on any operating system as well as its database reliability. The web browser is needed to view the result.

The tool chosen is based on the environment of the prototype that is to be implemented. The user environment chosen is based on its reliability to operate on any operating system. Web-based development is user-friendly where it can be implemented on any operating system. The tools utilized in this study are Dreamweaver, XAMPP and web browser (e.g., Google Chrome, Mozilla Firefox). Dreamweaver is used to write the front and backend of the prototype. The frontend of the prototype will be using Hypertext Markup Language (HTML5) and Bootstrap. Meanwhile, for the backend, Hypertext Preprocessor (PHP) is used. The knowledge-based is stored in XAMPP as MySQLi language is used to retrieve and manipulate the data. The web browser is used to view the result. Google Chrome web browser is chosen as the best web browser to run the prototype.

The prototype is developed in a Windows operating system environment. The client may have different hardware and software specification to run the prototype. Shown in Table 3.3 is the minimal requirements to run the prototype.

Table 3.3: Hardware minimal requirement

Hardware Minimal Requirement	
Random Access Memory (RAM)	64 MB
Hard Disk	350 MB
Operating System	Windows XP

The latest version of the Windows 10 operating system has been used for development and implementation. This is to ensure the latest stable version of XAMPP can run on optimum performance. 8GB of RAM provided with sufficient allocation of hard disk space.

The architecture design shows the functionality of the prototype. The component diagram had been chosen due to the capability of it to express each of the function in the prototype. The component diagram is a design based on the framework that had been discussed in Phase 2 earlier. The first step discusses the implementation of the framework. The framework had been discussed in Phase 2.

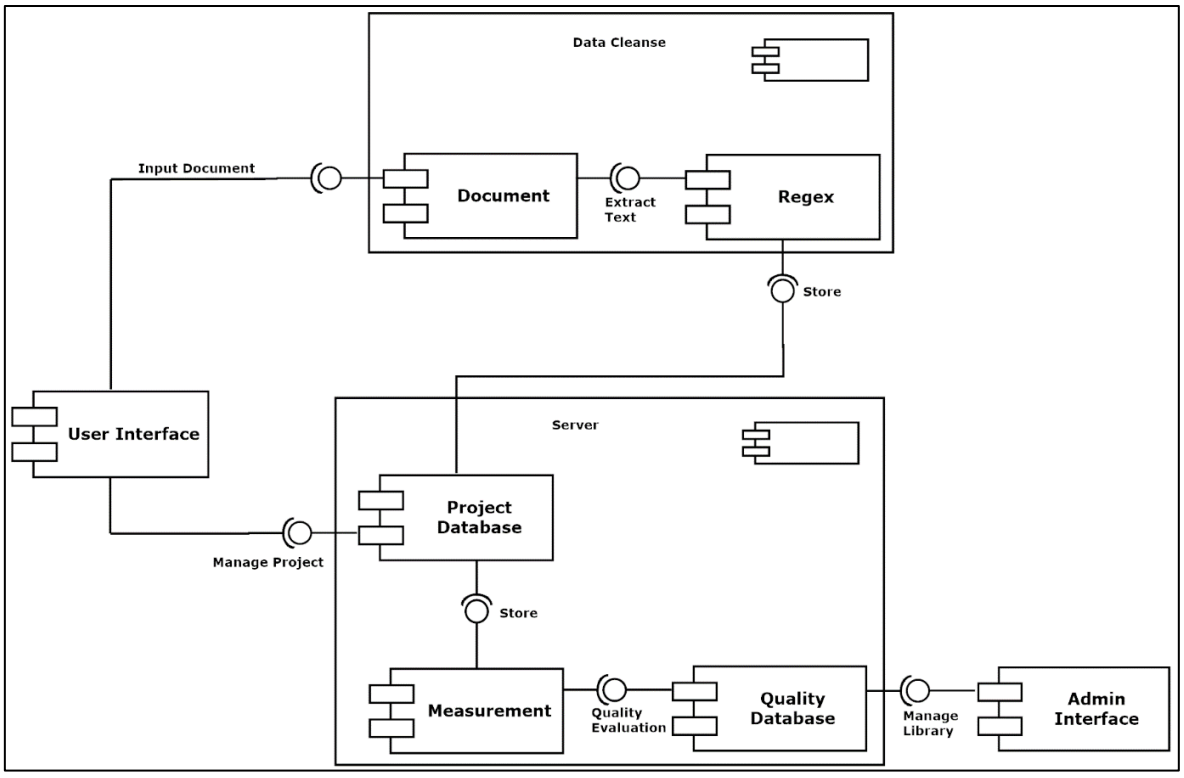


Figure 3.4: Component Diagram

The component diagram in Figure 3.4, is a visual representation of a connection between each of the components. Each component represents the subsystem that needs to work together. In general, the component diagram in Figure 3.4 best describes where it consists of the user interface, admin interface, data cleanse and server component. The data cleanse and server component consist of a nested component. The user interface component consists of numbers of a graphical user interface where exists the interaction between the user and prototype. That component allows the user to manage their own project. The admin

interface component consists of a graphical user interface where the interaction between the admin and prototype exist. The admin will only able to manage the corpus that will be used by the prototype to evaluate the inputted document by the user.

The initial input document will undergo a data cleanse component. In the data cleanse component, there are two nested components which are document and regex document. The document component will extract the text from the inputted document. Meanwhile the regex component responsible to cleanse the extracted text. The cleanse extracted text then store in the server component.

Evaluation of the inputted document started in the server component. There are three nested components in the server component which are quality database, project database and measurement component. The quality database component consists of a corpus that will be used to evaluate the document inputted by the user. Project database component consists of evaluation result, raw text, cleanse raw text and user information detail. The measurement component consists of an engine that evaluates the stored project with the corpus from the quality database. The result from the evaluation is stored in the project database which can be displayed only by the user.

3.5 Phase 4: Evaluation of Formalize Problem

There are three steps in order to evaluate the formalize problem which are; first evaluation of the framework to identify the limitation, second is to test the prototype and lastly is to evaluate the degree of quality document.

All of the steps mentioned above will be express more in Chapter 5. A case-control study is used to evaluate the implemented framework which had been discussed in Section

3.3. A number of different ways in writing FR and a sample of SRS document structure constructed as train data to study the prototype result. Each of the written FR is defined personally based on the collected SRS. The collection of the SRS document in Section 3.3 shows a variety of ways of writing the FR. Those variation patterns became an input to study the prototype output. The output will reveal the limitation of the prototype as well as its achievement.

The second step is to test the prototype with an actual case study. The actual case study is chosen randomly from the collected SRS document. The written FR and test cases are extracted from the SRS document. First, the SRS document inputted into the prototype to measure the completeness of its structure. Then the extracted FR and test case inputted manually into the prototype. Inputted FR will be converted by the prototype which resulted in it to be written in the RB template. Once the FR verified, the test case can be inputted to the correspond FR. The measurement and evaluation will be done automatically by the prototype.

The final step is the evaluation of the SRS document quality. The capability to automate the measurement increase the reliability result produce by the prototype. Aside from that, the user can automate the evaluation of their SRS document quality.

3.6 Summary

This chapter can be summarized as a discussion on the process of how the research methodology work. This research methodology had four phases. The first phase discusses the cause of the problem to be tackled on. The second phase composed of the research question, data collection, propose of properties to be assessed and framework to measure the SRS document. Rule and equation are proposed based on the proposed framework. The third

phase discusses the conversion of the framework into a prototype. Hardware and software requirement is discussed in this phase. Lastly, the fourth phase discusses the evaluation of the framework based on the prototype. Train data and an actual case study will be the input. The result of both inputs is discussed more in Chapter 5.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

This chapter focuses on the implementation of the proposed framework. The workflow of the prototype is explained in detail. Implemented also are the proposed rule and equation as stated in Chapter 3. Furthermore, sample codes relate to the prototype are explained in detail. The software tool and environment used for prototype development are also explained in detail.

4.2 Prototype Architecture

The rules and measurements to evaluate the structural and FR of SRS had been proposed in Chapter 3. Unified Modelling Language 2.0 (UML 2.0) is used to visualize the object interaction (Laplante, 2007). Two UML models techniques applied to this research which is activity diagram and entity relationship diagram.

The activity diagram is intended to show the activity involved in a process (Miles & Hamilton, 2006; Sommerville, 2016). An activity diagram to model up the activation process to measure the SRS document will be discussed in the next section.

Meanwhile, the entity-relationship diagram shows the interaction between classes in the business system (Dennis et al., 2013). An entity-relationship diagram is used as a model for the database to study the relationship between one class to another. The architecture of the prototype is divided into four segments based on the number of qualities that is to be assessed. Each of the qualities will be assessed based on the properties proposed in Chapter 3.

4.2.1 Activity Diagram

The prototype activity process is generalized in Figure 4.1 to make it easier to read.

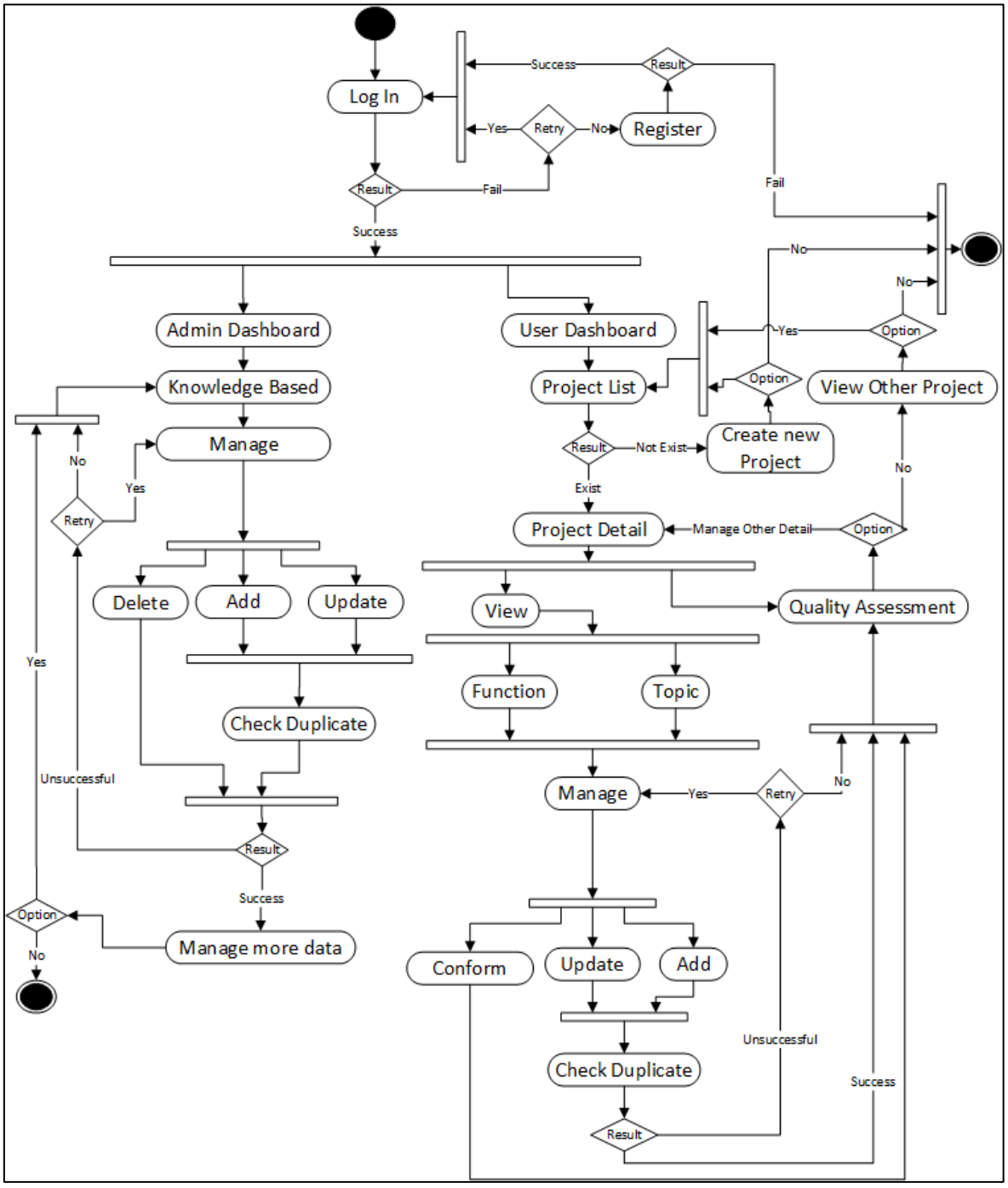


Figure 4.1: Prototype activity diagram

An activity diagram to model the process to measure the SRS document is shown in Figure 4.1. There are two levels of actor shown in Figure 4.1. Those are user, <*user*> and

admin, <*admin*>. Both of the actors will handle different processes. The activity diagram in Figure 4.1 can be summarized as the <*user*> is able to manage their own project meanwhile the <*admin*> is able to maintain the knowledge-based used for pattern-matching. The pattern-matching technique used will be explained further with a sample of codes and supported with the entity-relationship diagram in the Prototype Design section.

Based on the <*user*> perspective, the <*user*> must register before logging in into the prototype. Each of the <*user*> will have a unique email address. Upon successful login into the prototype, the <*user*> will be redirected to the user dashboard page. The list of projects is displayed on the dashboard page. If the project intended by the <*user*> for assessment is not listed, the <*user*> can add new projects. The <*user*> is supposed to supply the prototype with two documents in .docx format. First is the full SRS document and the second is the document containing the FR section extracted from the SRS document. The full SRS document is intended to be used for structural assessment meanwhile the extracted FR document is intended for FR assessment.

The assessment of the structure of the SRS document is started once the <*user*> input in the document in the correct format. The <*user*> will only be able to view the result of the assessment. The result of the assessment did include the possible suggestion of a similar topic. It is the <*user*> choice to update the document based on the suggested topic. If the <*user*> wants to update the document, the document must be updated manually and reupload again for a new assessment. Once the assessment on the structure of the SRS document completed, the next assessment will be the FR.

Assessment on the FR requires the involvement of the <user>. The <*user*> inputs the extracted FR document into the prototype. The <*user*> needs to define the stakeholder

corresponding to the assessed project. The prototype refines the provided document based on the RB template. The is also the function to add the FR manually. The <*user*> needs to conform to which FR is related to the project. Once conform, the <*user*> can manage the conformed FR. The test case can also be added once the FR is conformed.

The result of the structural and FR assessment is presented in a star chart. The result involves the calculation for each of the qualities assessed. The result is dynamically changed if the user did an update on either the SRS document or the FR. The <*user*> can only view the result for their own project. This is to ensure the security and privacy of others <*user*> data.

Another perspective that is to be discussed is the <*admin*> perspective. The <*admin*> login requisite is already pre-prepared. Once successful login, the <*admin*> will go to the admin dashboard. The <*admin*> is able to manage the knowledge-based to ensure the reliability of measurement. Those knowledge-based is composed of a library of words. Those words will be used as input in pattern-matching to assess structural or FR. Each of the quality assesses has its own library. The completeness quality composed of a library of the IEEE 830 table that contains topics and synonym topics. The consistency quality composed of stakeholders that were defined by the <*user*>. The correctness quality composed of a library of CRUD design and the defined stakeholder. The preciseness quality composed a library of possible datatype and vague words. Aside from the library of the stakeholder, all of the other libraries are manageable by the <*admin*>.

4.3 Prototype Design

The prototype design will focus on quality assessment. Before the assessment, data cleansing will be done. The activity diagram in Figure 4.2 shows the flow of data cleansing for the provided SRS document before the assessment.

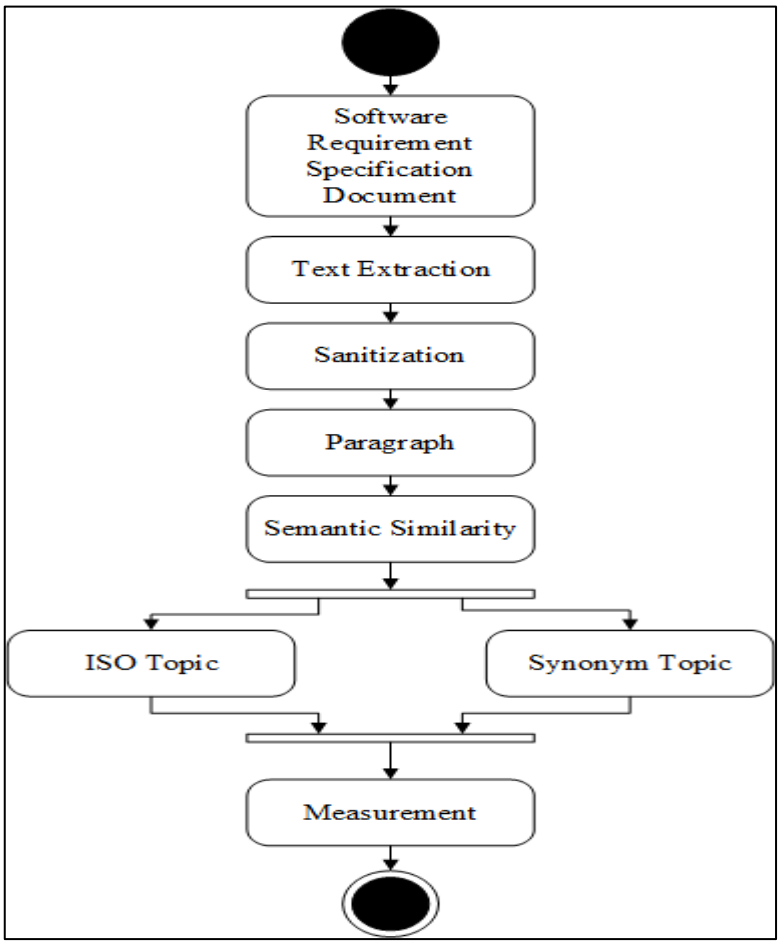


Figure 4.2: Sanitization of software requirement specification document

Based on the activity diagram in Figure 4.2, the text is extracted from the inputted SRS document. The extracted text is then stored in the database. The text is sanitized by using a regular expression. Shown in Table 4.2 is the sample code to sanitize the text and pattern matching using a regular expression.

Table 4.1: Sample code sanitization structural assessment

Rule	Code sample
R1	$CleanSpecialCharacter=preg_replace(“/[^a-zA-Z]/s”, “”, $Text);
R2	$SplitText=preg_split (‘/$\R?^/m’, $WholeText);

R1 rule applied where each word must be lowercase and all special character including numbering will be deleted. Once again, the regular expressions are used to perform pattern-matching. R2 rule applied where the text is split based on the paragraph. Pattern-matching has done between the spliced text with the IEEE 830 topics and the synonym topics. Results are stored in the database. The result is then used to measure the degree of quality of structural assessed.

```
// Assessment for IEEE topic
foreach($ArrayUnique as $indiFinalTopic){
        $querySearchIEEETopic = mysqli_query($con, “SELECT * FROM complete WHERE topic LIKE BINARY ‘$indiFinalTopic’”);
        $row24=mysqli_fetch_array($querySearchIEEETopic);
        $topic6=$row24[“comId”];
        mysqli_query($con, “INSERT INTO topic_detect(todeproId, todecomId, todecomsynId, todeboth) VALUES(‘$get_proId’, ‘$topic6’, ‘0’, ‘$topic6’)”);
}

// Assessment for Synonym topic
foreach($ArrayUnique2 as $indiFinalSynTopic){
        $querySearchSynIEEETopic = mysqli_query($con, “SELECT * FROM complete_syn WHERE comsyntopic LIKE BINARY ‘$indiFinalSynTopic’”);
        $row26=mysqli_fetch_array($querySearchSynIEEETopic);
        $topic8=$row26[“comsyncomId”];
        $topic9=$row26[“comsynId”];
        mysqli_query($con, “INSERT INTO topic_detect(todeproId, todecomId, todecomsynId, todeboth) VALUES(‘$get_proId’, ‘0’, ‘$topic9’, ‘$topic8’)”);
                                                }
```

Figure 4.3: Sample of pattern-matching code for structural assessment

Figure 4.3 shows the sample pattern-matching code to assess the structure of the SRS document. IEEE 830 ToC topics are used to compare with the provided SRS document.

Then, the detection results are stored in '*topic_detect*' table. A similar technique is used to evaluate between synonym topics with provided SRS document. Each topic in the IEEE 830 ToC and its synonym will be compared with the assessed document. Each matched topic will be denoted as one otherwise zero.

Table 4.2: Example pattern-matching denotation result

Topic	**Pattern-matching Denotation**	
	IEEE 830 Topic	**Synonym Topic**
Introduction	1	0
Beginning	0	1
Speed	0	0

Shows in Table 4.2 are an example of a topic matched with the IEEE 830 topic as well as the synonym topic. The process is quite simple and done automatically by the prototype. Let's take '*Introduction*' topic as an example. The '*Introduction*' topic will be matched first with the knowledge-based contain IEEE 830 topic. If any match found then the assessed topic will be denoted as one otherwise zero for IEEE 830 topic and it will undergo second knowledge-based which contains a synonym topic. But if no match was found then the topic will be treated as an additional topic that is not part of the assessment. The formalization of the structural in Chapter 3 is implemented in the prototype. This allows the prototype to evaluate the assess document automatically. A number of IEEE topics as well as the synonym topic detected in the '*topic_detect*' table will be calculated and became an input to the equation proposed.

The evaluation of the FR is based on the quality properties that had been proposed in Chapter 3. Before the evaluation is started, each of the FR stated in the FR document will be

converted into the RB template. Shown in Figure 4.4 is the activity diagram on how the prototype converts normal text into individual sentences based on the RB template.

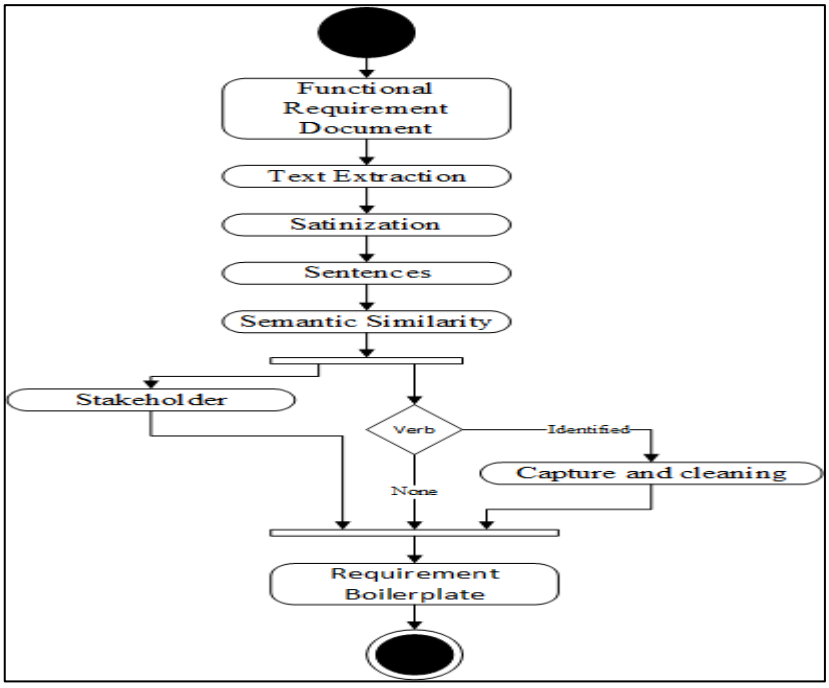


Figure 4.4: Sanitization of Functional Requirement Document

Based on Figure 4.4, the text from the file containing the FR is extracted. The text is sanitized using a regular expression. Two rules are applied in order to sanitize the text. Shown in Table 4.3 is the rule along with sample code to sanitize the FR before refining it based on the RB template.

Table 4.3: Sample code sanitization functional requirement assessment

Rule	Code sample
R1	function singularize($params)
R2	$sentencesFr=preg_split('/(?<=[.?!])\s/i', $textFr);

R1 rule contains a sample code to sanitize the extracted text by undergoing a singularization process for standardization. The singularization process is important to increase the detection process. The singularization process started once the prototype is

provided with the SRS document. The text is extracted from the document and is tokenized. The singularization process converts any plural tokenized word into a singular word. The idea is to increase the reliability of pattern-matching due to knowledge-based use only the singular word. Then R2 rule is applied. Those texts will be separated into individual sentences based on full stop punctuation. The text will then undergo pattern-matching based on R2 rule and if successful will result in the conversion of text into a single sentence.

The single sentences will be sanitized first in order to refine it based on the RB template. RB is divided into two segments which are stakeholder and capability. Based on Glinz et al. (2017), the requirement is defined based on the stakeholder. As mentioned earlier in Chapter 4**,** the <*user*> will define and input the required stakeholder into the prototype.

```
$querySearchStaNameFR=mysqli_query($con, "SELECT * FROM func_auto WHERE funcautoproId='$get_proId' AND funcautoname LIKE '%$staName%'");
```

Figure 4.5: Sample code to identify stakeholder

Shown in Figure 4.5 is the sample code to inquire about the presence of defined stakeholders in each sentence. The prototype will identify the corresponding stakeholder in each sentence based on the stakeholder define by the <*user*>.

The refinement process for the capability segment is started with each of the sentences tokenized and compared with the lists of words gathered from WordNet v3.0 SQL Builder (Christiane, 2005) in order to identify the part-of-speech (POS) of the word, in particular to the verb. The word is later stored in the database for further sanitization.

```
Preg_match_all('#(?<=^|\W)('. Implode('|', array_map('preg_quote', $array_of_verb)).
')(?=$|\W)#i', $sentence, $matches);
```

Figure 4.6: Sample code to identify verb in sentence

A regular expression is used to identify the verb in the sentence. Figure 4.6 is the sample code containing the pattern used to identify the verb in sentences after tokenization.

Any identified verb in the capital letter is filtered out as it will not produce a match when generating the capability segment.

Table 4.4: Example of sentence sanitization

Sentence		***Sign*** Up: Users need to ***sign*** up to ***use*** the web site.
Sanitization	**Before**	Sign Up: Users need to sign up to use the web site.
	After	sign up to use the web site

Shown in Table 4.4 is the example of a sentence undergoing sanitization. The result of this sanitization will become part of the capability segment. The word that is in bold and italic is the verb found in the sentence. During the sanitization process, it is clearly seen that the verb is positioned at the beginning of the sentences. A rule is added to filter out the verb that starts with a capital letter. The position of the verb in the sentence is also important. This is due to the prototype is designed to analyse the location of the verb found before it becomes part of the capability segment. Any words that happen to be in between the identified verb word will become additional information. The impact of the additional information can be seen where it is necessary for a sense to provide essential context to understand the sentence. After the original sentence is refined into the RB template, then only the evaluation quality of the consistency, correctness and preciseness started.

Evaluation of the consistency quality is based on the properties of the stakeholder and its role. Based on the activity diagram in Figure 4.1, the <*user*> is the one that will define the stakeholder. The purpose of the properties of consistency is to avoid the redundancy and inconsistent usage of the stakeholder in a defined FR. Each of the sentences identified is matched based on stakeholder defined by <*user*>. The matched sentence will be denoted as one and zero otherwise.

The second part of the consistency quality properties evaluation is to measure the role. The first step is to analyse the capability segment. The prototype will identify the similarity of one sentence to another. There is two types of method used to identify the similarity. First is by comparing it with the exact word by word. The result will be presented in percentage form. Aside from that, the nearest similar function based on the percentage is also displayed. The second method is by using a similar text technique. Levenshtein Distance works by computing the minimum number of changes needed to make the two compared sentences identical (Essatouti et la., 2018). A similar text technique is almost similar to Levenshtein Distance, but similar text technique is chosen due to its accuracy of the result and fewer modifications needed (Data, 1998).

Final conformation for consistency check will be done by the <*user*>. The <*user*> can choose in between three option; the original refinement sentence, similarity by word or similar text technique.

```
// Similarity by word
$arrayAssessFR = explode(' ', $function);
$countarrayAssessFR = count($arrayAssessFR);
$arrayAllFR = explode(' ', $allFR);
$SimilarSentence = array_intersect($arrayAssessFR, $arrayAllFR);
$countSimilarSentence = count($SimilarSentence);
$similarity=implode(' ',$SimilarSentence);
$result = array_diff($arrayAllFR, $arrayAssessFR);
$countresult = count($result);
$zem = implode(' ',$result);
$calculateDifferences = ($countSimilarSentence / $countarrayAssessFR) * 100;
// Similar text method
$Similar_text = similar_text($function, $allFR, $percentage);
```

Figure 4.7: Sample code for similarity checking

Figure 4.7 shows the sample code for the percentage calculation of similarity for the capability segment before it is stored in the database. The results are viewable by the <*user*>. <*user*> will be the one that will confirm either the FR assessed is redundant or not.

In order to evaluate the correctness quality properties, the use case testing was adopted. The prototype was designed where each of the FR needs at least one test case. In order to evaluate the test case, two use case testing elements are assessed.

```
$querySearchStakeholder=mysqli_query($con, "SELECT * FROM test_case q JOIN function w ON q.testfuncId=w.funcId AND q.testnormalflow LIKE '%$identifyFuncStaName%' WHERE q.testId='$identifyTestCaseID'");
```

Figure 4.8: Sample code to detect present of stakeholder

First is the presence of the stakeholder on the actor element. Shown in Figure 4.8 is the sample code to detect the presence of stakeholders in the assessed FR. The presence of stakeholders is automatically detected by the prototype. This is due to the test case that can only be added on a conformed FR.

```
$querySearchFlow=mysqli_query($con, "SELECT * FROM test_case WHERE testnormalflow LIKE '%$identifyFlow%'");
```

Figure 4.9: Sample code to detect present of possible design

Second is the presence of possible design and stakeholder on the normal flow element. Shown in Figure 4.9 is the sample code to detect the presence of possible design in the normal flow element. Mentioned earlier in the previous subsection, knowledge-based has been created which stores a number of terms that represent the possible design. The flow in the normal flow element is more technical than the user story. The chances that the possible design term is used in the normal flow element is high. The integrity of the flow in normal flow can be heightened by the presence of stakeholders from the actor element. The normal flow element should describe the activity between stakeholders.

Each successful detection of stakeholder and possible design in the normal flow element will be denoted as one. The results will be stored in the database. The detection result will be applied to a measurement ensuring the validity of test case for each FR. To

ensure the validity of test case generated, the stakeholder and possible design must be detected in normal flow element.

The next quality to be assessed is the preciseness quality of the FR. The use of restricted language to define the FR limits the possibility of ambiguity. The adoption of RB allows the restriction of language usage. The proposed quality properties are the presence of possible vague words and datatype.

```
$queryPreciseFunctionNew=mysqli_query($con, "SELECT * FROM function WHERE
funcproId='$get_proId' AND funcconform='Yes'");
while($row801=mysqli_fetch_array($queryPreciseFunctionNew)){
        $functionNew=$row801["function"];
        $functionId=$row801["funcId"];
        foreach($precNameLibrary as $preciseChecker){
                $matchEndSentences=preg_match("/\b($preciseChecker)\b.$/i", $functionNew);
                if($matchEndSentences){
                        mysqli_query($con, "UPDATE function SET tocalvague='$preciseChecker'
WHERE funcId='$functionId'");
                }
                $matchFrontSentences=preg_match("/^\b($preciseChecker)\b/i", $functionNew);
                if($matchFrontSentences){
                        mysqli_query($con, "UPDATE function SET tocalvague='$preciseChecker'
WHERE funcId='$functionId'");
                }
        }
}

$queryDatatypeFunction=mysqli_query($con, "SELECT * FROM function WHERE
funcproId='$get_proId' AND funcconform='Yes'");
while($row301=mysqli_fetch_array($queryDatatypeFunction)){
        $dtxt=$row301["function"];
        $dfuncId=$row301["funcId"];
        $dpattern='#(?<=^|\W)('. Implode('|', array_map('preg_quote', $array_of_datatype)).
')(?=$|\W)#i';
        if(preg_match_all($dpattern,$dtxt,$dmatches)){
        }
        $dflat=call_user_func_array('array_merge', $dmatches);
        $dflat2=implode(",",array_unique($dflat));
        $dflat3="'$dflat2'";
        mysqli_query($con, "UPDATE function SET tocaldatatype='$dflat2' WHERE
funcId='$dfuncId'");
}
```

Figure 4.10: Sample code to check the present of vague word and datatype

The prototype is designed to automatically check for the presence of vague word and datatype once the FR is inputted. Figure 4.10 shows sample code to check the presence of

vague words and datatype. Each of the sentences in inputted FR is tokenized to individual words. Then each word undergoes regular expression with the library of datatype and vague word.

Adoption of RB did help a lot in terms of reducing the ambiguity of the sentence. Reducing the ambiguity does not mean the total elimination. Knowledge-based containing the vague word is used as input for pattern-matching with the sentence. Any matched word will be denoted as one otherwise zero. Matched words are highlighted and listed for <*user*> concern. A similar activity is done on detecting the possible datatype.

The database plays an important role in this study. It consists of a library of pattern-matching input to measure the degree of structural and FR. Figure 4.11 is the entity-relationship diagram for the prototype.

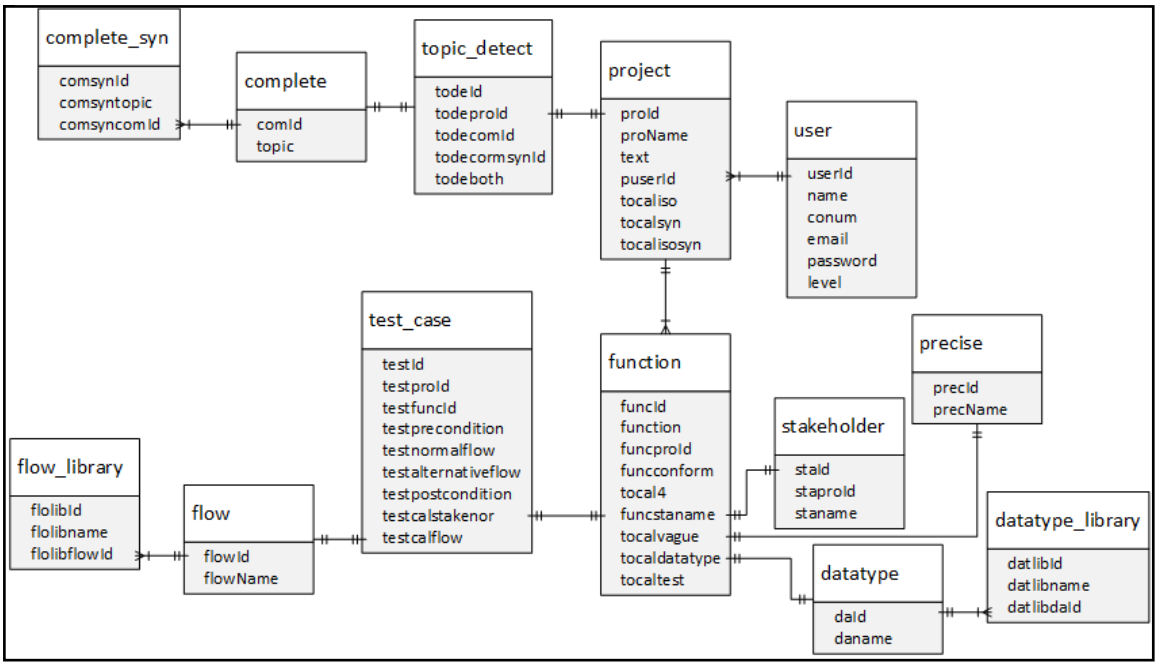


Figure 4.11: Entity relationship diagram

Based on the entity-relationship diagram, five main libraries contain terms used for pattern-matching. Those five are '*complete*' table, '*flow*' table, '*precise*' table, '*datatype*'

table and '*stakeholder*' table. Three out of five tables are synonyms table. Those three are '*flow_library*' table, '*datatype_library*' table and '*complete_syn*' table. Those three tables exist in order to increase the possible detection. '*Complete*' table consists of topics from IEEE 830 ToC with the assist of '*complete_syn*' table. '*Flow*' table is assisted with '*flow_library*' table where it is intended to detect the term that represents possible design in the '*Normal Flow*' element of test case. '*Precise*' table is independent where it only consists of numbers of possible vague words. '*Datatype*' table with the assist of '*datatype_library*' is intended to detect term to represent the possible presence of datatype in each of FR. The assist table is created to increase the reliability of the detection.

Meanwhile, for the measurement, four tables are used. Those tables are '*test_case*' table, '*function*' table, '*topic_detect*' table and '*project*' table. As discussed above, knowledge-based contains patterns that will be matched with each word in the tokenized FR. The pattern to match the properties assessed is expressed in the regular expression. Successful pattern matched will be denoted as one, and otherwise is zero. The result of pattern-matching will be stored in the corresponding results. The results of the structural will be stored in the project table. Meanwhile, for the FR, results will be stored in the function table. The equation in Chapter 3 had been implemented in the prototype. The prototype will calculate and display the results in percentage form.

4.4 Prototype Sample Output

This section discusses in detail the process conversion of input to output. As mention earlier in the previous section, the user needs to be registered first. Only successful registered user is allowed to use the prototype. Once successfully login, the user can insert 2 documents each project which is SRS document and FR document.

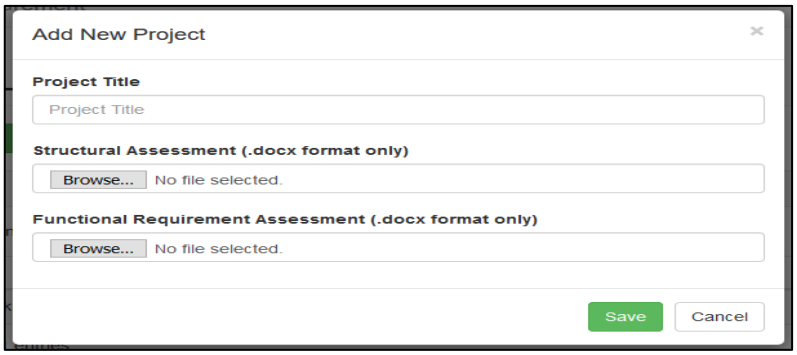


Figure 4.12: Sample of add new project

Shows in Figure 4.12 is a sample on how the user can add a new project to the prototype. The user needs to input the project title. Then the user is required to input the SRS document in the Structural Assessment box meanwhile FR document in the Functional Requirement Assessment box. The user only allows to input .docx format only as the prototype is designed in a way it only recognizes that format only.

Table of Contents

1. Introduction
   1.1. Purpose
   1.2. Document Conventions
   1.3. Intended Audience and Reading Suggestions
   1.4. Project Scope
   1.5. References
2. Overall Description
   2.1. Product Perspective
   2.2. Product Features
      2.2.1. Login 2
      2.2.2. Enrolment 2
      2.2.3. Book Flights 3
      2.2.4. Reserve Seats
      2.2.5. Flight Status 3
      2.2.6. Flight Schedules
      2.2.7. My Account 4
      2.2.8. Logout 4
   2.3. User Classes and Characteristics

Figure 4.13: Sample portion SRS document contain ToC

The sample portion of ToC from SRS document input in the prototype is shown in Figure 4.13. Initially, the sample of ToC contains 38 topics. Once the save button clicked,

the prototype will automate the process of calculating the structural assessment. Only the SRS document input in the Structural Assessment box in Figure 4.12 will involve for the structural assessment.

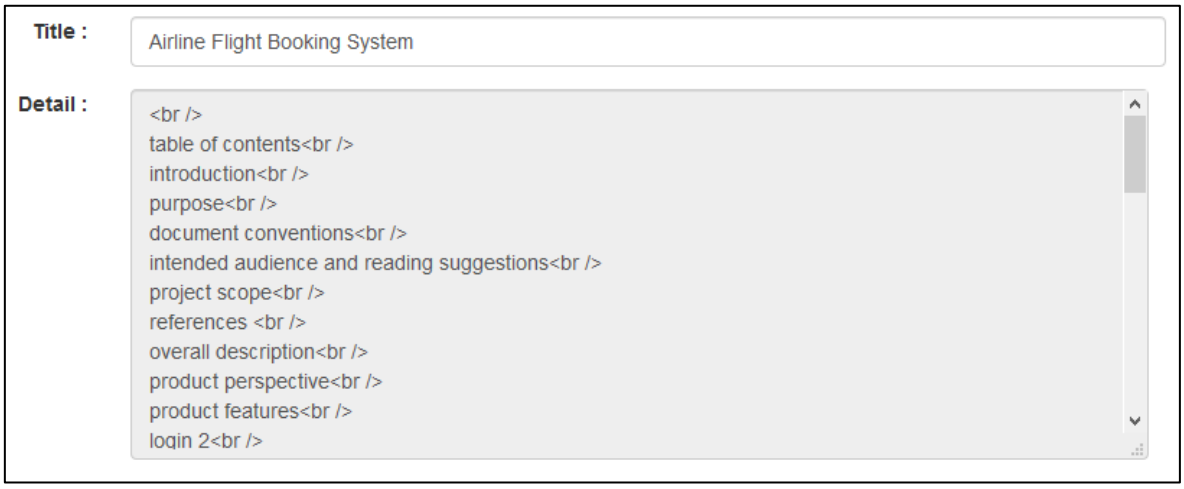


Figure 4.14: Sample result of text sanitization

Before the calculation process invoked, the raw text extracted from the SRS document inputted into the prototype and store in the database. Then the rule from Table 4.2 applied to cleanse the raw text to useful data. The text is cleansed to ensure the reliability of data produces aside from increasing the probability of pattern-matching success. The end result of data cleansing can be seen in Figure 4.14.

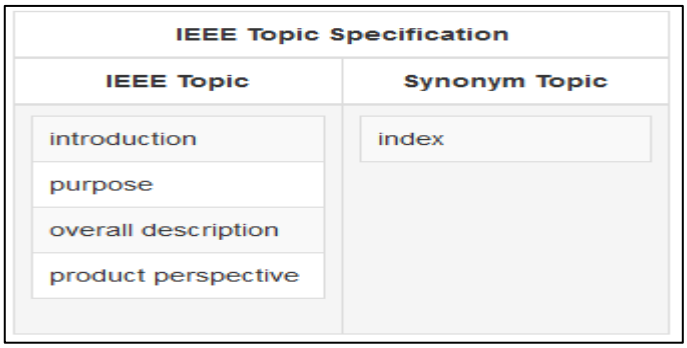


Figure 4.15: Sample result pattern-matching for structural assessment

Once the data cleanse, the prototype will automatically run the pattern-matching process. The pattern-matching process involves between the cleanse data against knowledge-

based contain IEEE 830 ToC and synonym topic. The result of pattern-matching can be seen in Figure 4.15 where only the matched topic is presented. The unmatched topic is treated as an additional topic that is ignored by the prototype. The additional topic is ignored as it presently does not give any impact on the completeness of the SRS document structure.

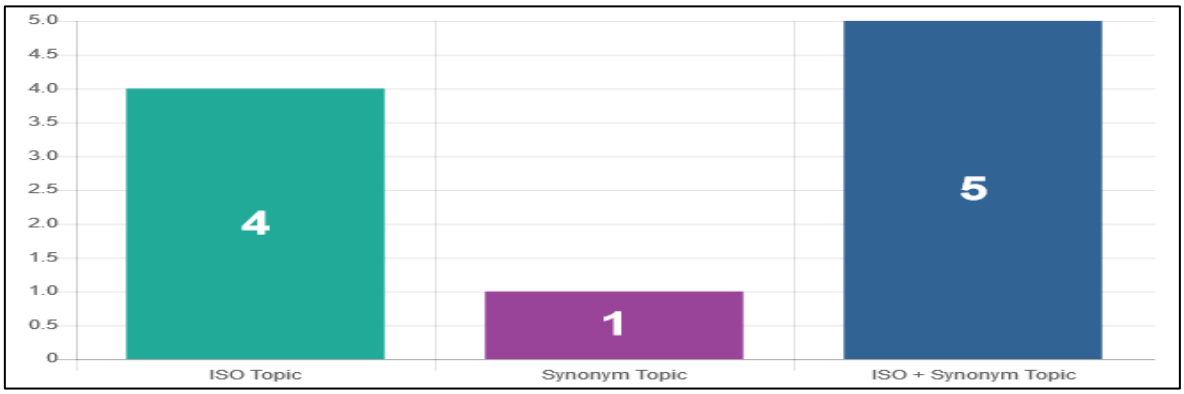


Figure 4.16: Graph presentation of detected topic

As shown in Figure 4.15, each of the detected IEEE 830 topics is denoted as 1 which resulted in 4 detections. Similar to the synonym topic, each detected topic is denoted as 1 which resulted in 1 detection. The result of detection is shown in the bar graph as in Figure 4.16. The first bar represents the number of detections for the IEEE 830 topic which is 4. The second bar represents the number of detections for the synonym topic which is 1. The third bar represents the highest result of the update if the user willing to change the detected synonym topic based on the IEEE 830 topic suggested.

Synonym Topic	
User Topic	**Similarity**
index	table of content and index

Figure 4.17: Sample suggested synonym topic to IEEE 830 topic

Shows in Figure 4.17 is the sample suggested change that should be done by the user. As the studies tried to promote the IEEE 830 topic as a general structure, the prototype

should be able to suggest the nearest similarity between the user topic toward the IEEE 830 topic. This is the reason why the prototype is designed to come out with the third bar of the graph shown in Figure 4.16.

Likewise, as the study only focuses on meeting all of the IEEE 830 topics, the measurement will only focus on the present of the IEEE 830 topic. The completeness quality equation (Equation 3.2) will be used to evaluate the SRS document structure.

$$S = \left(\frac{4}{25}\right) * 100\%$$

$$S = 16\ \% \qquad \text{Equation 4.1}$$

The calculation result shows in (Equation 4.1). As the number of IEEE 830 topics detected from the user inputted is 4 then it will be divided with 25 which is the total number of IEEE 830 topics. Then the result is shown in percentage form.

Based on the first hypothesis stated in Chapter 1, the structure of the SRS document is evaluated based on the complete presentation of the topic suggested in IEEE 830. Collection of topics regardless of its main or sub-topic gathered from IEEE 830 standard (IEEE Std. 830-1998, 1998). Altogether, there are 25 topics. IEEE 830 is used as a standard guideline to create a SRS document. It contains good characteristics that should be implemented in each generated SRS document. Included also is the standard structure which contains a collection of topics in order to produce a good and readable SRS document. Numbers of recent researchers still focus on enforcing the standard as part of the structure in the SRS document (Takoshima and Aoyama, 2015; Thitisathienkul and Prompoon, 2015; Anil and Moiz, 2017; Asif et al., 2019). This justifies the usage of 25 as a constant in calculating the completeness structure of the assessed SRS document. It is obvious that the assessed SRS document might have a different number of topics compared to the topic

suggested by IEEE 830 but the focus of the study is to enforce the use of the topic suggested by the standard.

Moving on to the FR assessment, as mention earlier, the FR document is inputted into the prototype. The FR document only contains the list of FR related to the project to be assessed. In short, the user needs to extract the FR out from the original SRS document in order to produce the FR document.

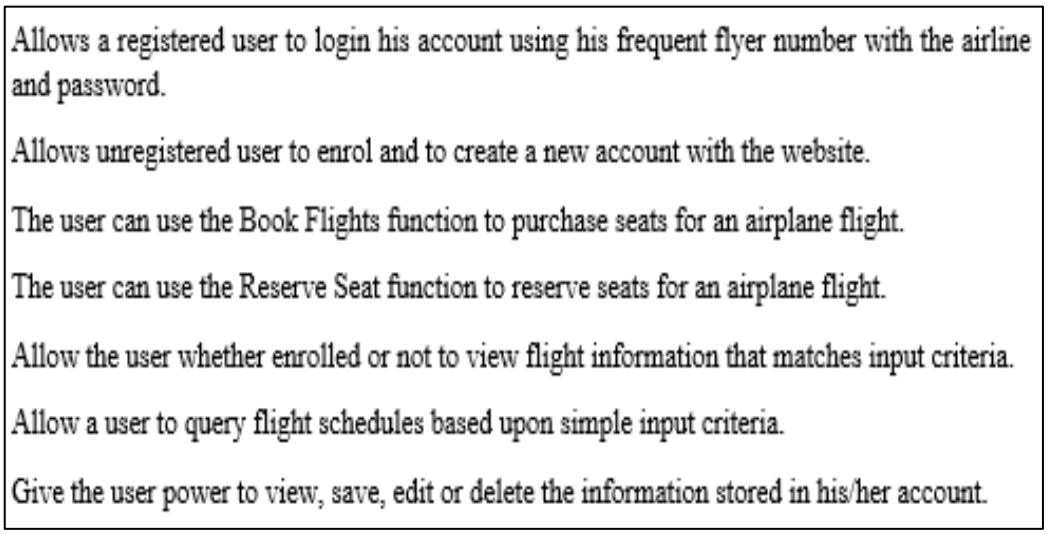


Figure 4.18: Sample of text in FR document

The initial step in converting the written FR to the RB template involves data cleansing. The sample text is shown in Figure 4.18 extracted from its document and stored in the database. Then the rules in Table 4.3 applied to cleanse the extracted text.

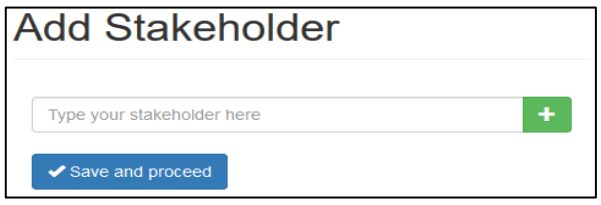


Figure 4.19: Sample on how to add stakeholder to the prototype

Before the next process started, the user will be requested to input the stakeholder into the prototype. Shows in Figure 4.19 is how the user can input the required stakeholder

into the prototype. Once the required stakeholder is defined, the prototype will automatically come out with the result of FR refinement based on the RB template as well as percentage similarity between all of the FR.

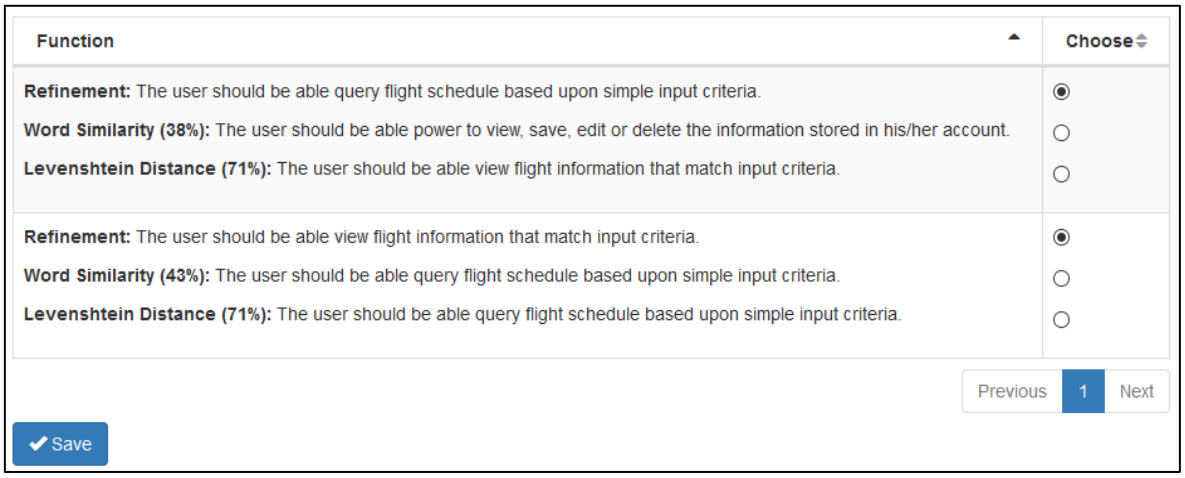


Figure 4.20: Sample result FR refinement with similarity check

The sample result FR refinement can be seen in Figure 4.20. The refinement result in the first row represents the sixth FR stated in the sample list from Figure 4.18. Meanwhile, the second row represents the fifth FR from the sample list. The sample percentage of similarity between each FR also shown in Figure 4.20. The prototype is designed to evaluate the similarity between each FR based on the similarity between word and per character used. The power to finalize the similarity between each FR is on the user's hand as the prototype is not designed to understand the meaning of each FR. Among the FR sample shown in Figure 4.18, let's say the user only agreed only six out of seven is non-similar.

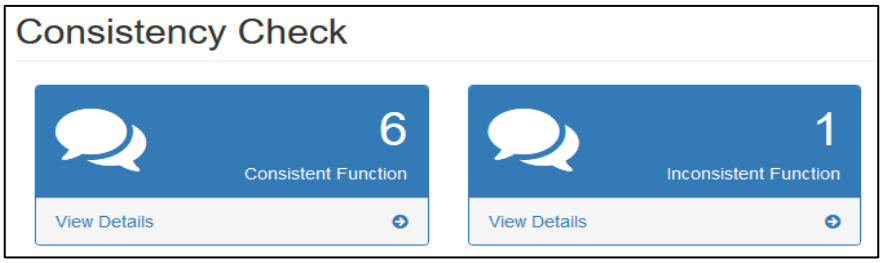


Figure 4.21: Sample result consistency check

The prototype allows the user to view the consistent and inconsistent FR as shown in Figure 4.21. Consistent Function detail contain non-similar FR. Meanwhile, Inconsistent Function detail contains similar FR.

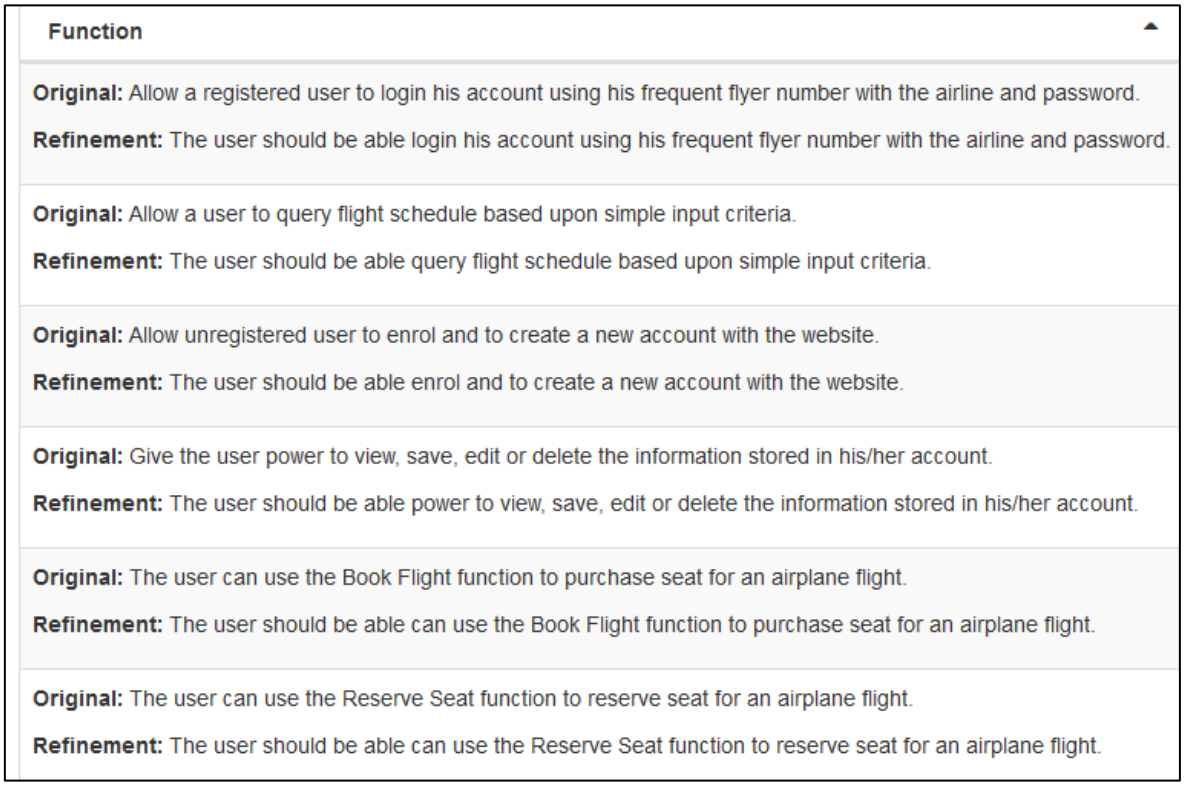


Figure 4.22: Sample contain of Consistent Function page

Once the user clicks on the Save button as in Figure 4.20, the prototype will automatically divide between similar and non-similar function. Shows in Figure 4.22 are the sample contain FR which is denoted as non-similar by the user. Each row in the table shows the original and refinement of FR. Original FR is the raw FR before refinement to the RB template. Refinement FR is the result of the raw FR converted based on the RB template.

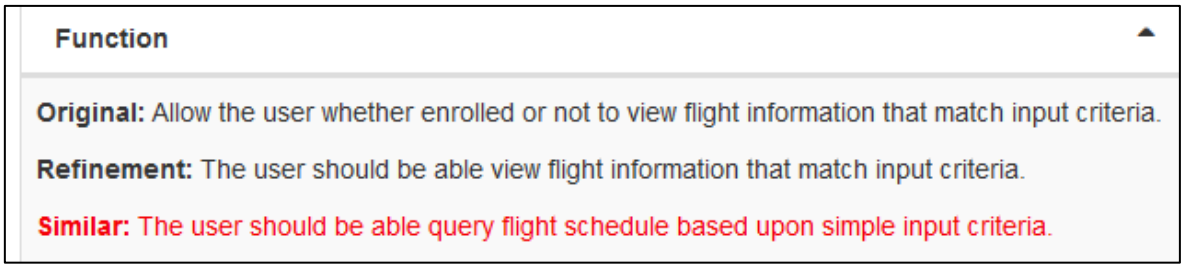


Figure 4.23: Sample contain of Inconsistent Function page

Shows in Figure 4.23 are the contents of the Inconsistent Function page. Unlike the Consistent Function page, each table row will contain Original and Refinement but in this Inconsistent Function page, the new element added which is Similar. A similar element represents the similarity that is checked by the <*user*> during the conformation step in Figure 4.20.

Once the user clicked on the save button, it means that the consistency equation will be invoked. The consistency equation (Equation 3.5) will be used in order to evaluate the consistency in all written FR.

$$T = \left(\frac{6}{7}\right) * 100\%$$

$$T = 85.7\% \qquad \text{Equation 4.2}$$

The calculation result shows in (Equation 4.2). Mention earlier, total non-similar FR agreed upon is six meanwhile total number of FR assessed is seven. The result of consistency quality in assessed FR is 85.7%.

The unstructured way of writing the FR as well as the usage of natural language lead toward ambiguity. The RB provides a structured way of writing the FR. The RB restrict the use of term or word to allow the developer to focus on writing the needs of the assessed stakeholder. The prototype helps in refining the unstructured FR into structured FR based on RB. It shows that the prototype complies with the second hypothesis as stated in Chapter 1.

Furthermore, the sentence similarity process had been done by comparing all of the refined FR to help developers check the percentage of similarity. The reliability result from the checking had been improved by using the similar_text() method instead of the Levenstein method used in research by Audytra et al. (2016). Furthermore, the second similarity check based on word usage also added in order to measure the percentage of similarity between assessed FR. However, the prototype is not designed in such it checks each FR based on its meaning. This allows the user of the prototype to have a final say in conforming to the similarity between FR to ensure its consistency.

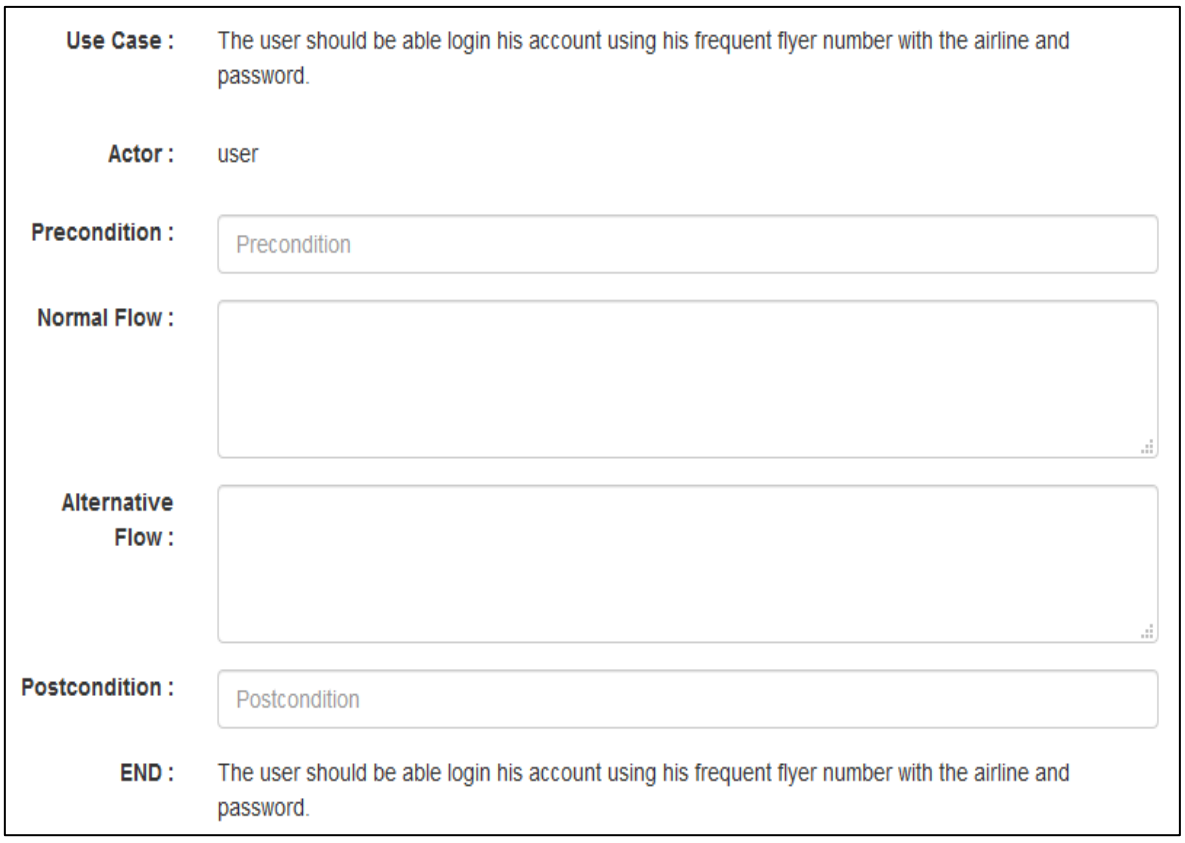


Figure 4.24: Sample of test case

The next assessment which related to correctness quality involves the user to input the test case for each written FR. The prototype is designed to automate only a certain element of test case. The initial look of the test case is shown in Figure 4.24. From the initial

look, the prototype will only help in assigning the '*Use Case Name*' and '*Actor*' element. The <*user*> needs to fill in the other element manual. Those elements are '*Precondition*', '*Normal Flow*', '*Alternative Flow*' and '*Postcondition*'. The evaluation of the test case will only involve the presentation of data in '*Normal Flow*' element.

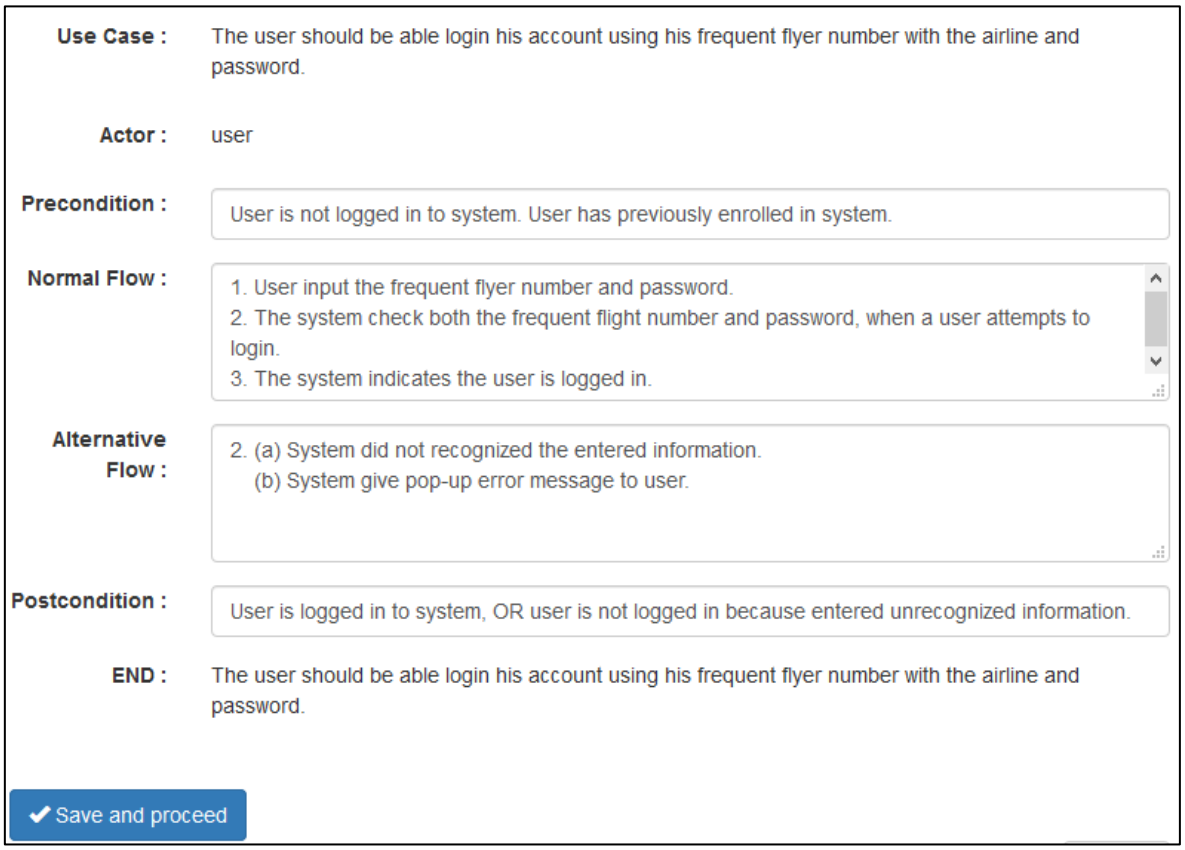


Figure 4.25: Sample input in test case

Shows in Figure 4.25 is a sample of input that can be processed by the prototype. Once the Save and proceed button clicked, the data will be saved in the database. The prototype will process the saved data to produce useful value. Based on the proposed rule (Equation 3.6), only three elements will be involved in the measurement which is '*Use Case Name*', '*Actor*' and '*Normal Flow*'. To ensure the rules are followed, the prototype is designed in a way that the validity of test case is check based on the presence of those three elements.

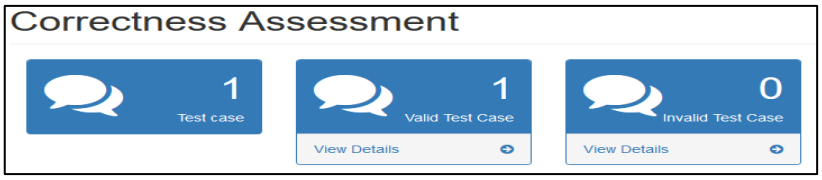


Figure 4.26: Sample of correctness assessment result

Let say the sample study only provides one test case for one FR out of seven FR provided in Figure 4.18. The result of the correctness assessment sample is shown in Figure 4.26. The first box represents the total test case inputted by the user. The second box represents the total valid test case. The third box represents the total invalid test case.

Measurement for the correctness quality is done automatically by the prototype. The equation proposed is used (Equation 3.7).

$$V = \left(\frac{\frac{1+1+1}{3} + \frac{0+0+0}{3} + \frac{0+0+0}{3} + \frac{0+0+0}{3} + \frac{0+0+0}{3} + \frac{0+0+0}{3}}{6} \right) * 100\%$$

$$V = \left(\frac{\frac{3}{3}}{6} \right) * 100\%$$

$$V = \left(\frac{1}{6} \right) * 100\%$$

$$V = 16.7\% \qquad \text{Equation 4.3}$$

Results for the correctness quality measurement is shown in (Equation 4.3). Since the test case is validated based on the presence of those three elements, each of the elements denoted as one is presented. In equation (Equation 4.3), the test case is assessed individually. As the sample only has one test case, then the prototype only checks the validity of that test case. The result from the checking will be divided by a constant of three. Then result from

the division will be divided by the total number of conformed non-similar FR. Finally, the result is shown in percentage form which in this calculation it resulted in 16.7%.

Capability to generate test case as early as the requirement phase shows an understanding of the developer toward the particular FR. Each generated test case composes of six elements. As mention in Chapter 3, the study focus on the present of '*Use Case Name*', '*Actor*' and '*Normal Flow*' element of a test case. Those three elements are compulsory to be filled in order to generate a proper test case. A recent study by Ahmad et al. (2018), the identification of the '*Actor*' element in each generated test case is important. From the generated test case, the '*Actor*' task will be elaborated in '*Normal Flow*' element. Commonly, each assessed FR corresponds to a single test case. The assessed FR corresponds to the '*Use Case Name*' element. The assessment validity of each test case is based on the presence of the compulsory element. Each of the compulsory element in the test case is denoted as one if found otherwise zero. Compared to the study done by Ahmad et al. (2018), there is no clear indication of how the '*Actor*' element help in term to validating the generated test case. In order to increase the reliability of the valid '*Normal Flow*' element, the presence of the stakeholder indicated in '*Actor*' element is a must. It's to ensure the developer stays focus only on discussing the assessed FR based on the corresponding stakeholder in '*Normal Flow*' element. To further improve the validity of the generated test case, the prototype is designed in such a way it automates the process to identify the '*Use Case Name*' and '*Actor*' element.

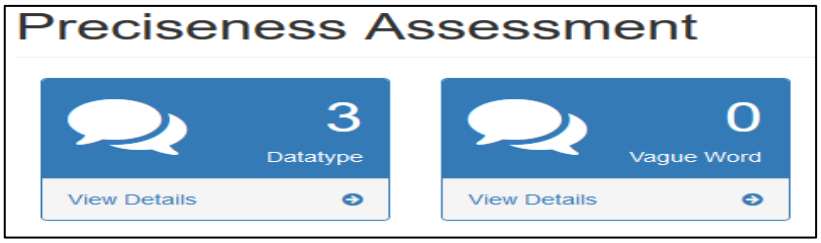


Figure 4.27: Sample of preciseness assessment result

The next assessment is regarding the preciseness quality of the FR. This assessment will be automated by the prototype itself. The prototype used a pattern-matching process to match the word in each assessed FR with two types of knowledge-based. The first knowledge-based is the datatype library which contains the possible word to represent datatype. Second knowledge-based is a vague word library which contains word represent the vague word. Shows in Figure 4.27 is the sample result from the preciseness assessment. Out of six conformed non-similar FR, only three of them show the presence of word representing possible datatype. Unlike vague words, there is no word representing vague words found.

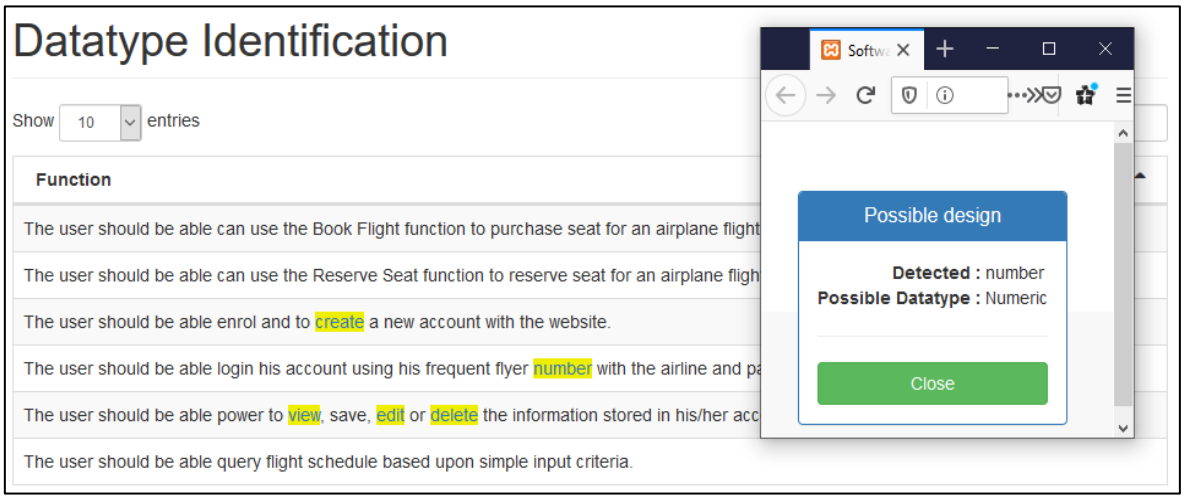


Figure 4.28: Sample of datatype identification result

Based on Figure 4.28, the highlighted is the word represent possible datatype. Once the highlighted word clicked, it will show the possible datatype that can be used in order to design the database. The view of the prototype is also similar to Figure 4.28 if there is a

presence of a vague word. The vague word in the sentences will be highlighted to alert the user on the position of those vague words.

Similar to the other qualities that had been assessed so far, the calculation will be automatically done by the prototype itself. The equation to measure the preciseness quality of FR had proposed in Chapter 3 (Equation 3.9).

$$P = \left(\frac{\frac{0+1}{2} + \frac{0+1}{2} + \frac{1+1}{2} + \frac{1+1}{2} + \frac{1+1}{2} + \frac{0+1}{2}}{6} \right) * 100\%$$

$$P = \left(\frac{\frac{1}{2} + \frac{1}{2} + \frac{2}{2} + \frac{2}{2} + \frac{2}{2} + \frac{1}{2}}{6} \right) * 100\%$$

$$P = \left(\frac{\frac{2}{2} + \frac{2}{2} + \frac{2}{2}}{6} \right) * 100\%$$

$$P = \left(\frac{3}{6} \right) * 100\%$$

$$P = 50\% \qquad \text{Equation 4.4}$$

The step by step equation to calculate the preciseness quality of FR shown in (Equation 4.4). Each of the conformed non-similar FR is evaluated. For example, let's take the first, second and sixth function in a respective row from the sample in Figure 4.28, which it shows no presence of the word represent possible datatype and vague word. Based on the rule proposed (Equation 3.8), it only met one of the properties which are the non-present of the vague word. That resulted in the example to get only one point as the word represent possible datatype are not present. In short, the first, second and sixth function in example gets zero denotation for the non-present of the word representing possible datatype and one denotation for the non-present of a word representing the vague word. This resulted in it is ignored in measurement as it did not fulfill the rule proposed. Meanwhile, since the third, fourth and fifth functions from the respective row of functions fulfill the proposed rule, each

function gets one denotation for the non-present of the word representing possible datatype and one denotation for the non-present of the word representing a vague word. The result of each assessed FR is the sum and divided with a total of non-similar conformed function. The final result then presented in percentage form which resulted in 50%.

Refinement of FR based on RB allow restriction of word usage and improve the preciseness of written FR. The restriction reduces the ambiguity of the define FR. However, there are other types of ambiguity need to be considered as such as present of the vague word. Vague word is collected as a library for pattern-matching in this research. The similar method used by other researchers (Kocerka et al., 2018; Osman and Zaharin, 2018). The preciseness of the written FR can be further improved by the usage of the meaningful term. This idea has been supported by Omoronyia and Stålhane (2017). The knowledge-based of a term representing the possible datatype is used to improve the meaning of the written FR. Collection data in Table 3.1 is based on 93 SRS document shows the frequency of term representing datatype exists. Those show the possibility to improve preciseness in writing the FR.

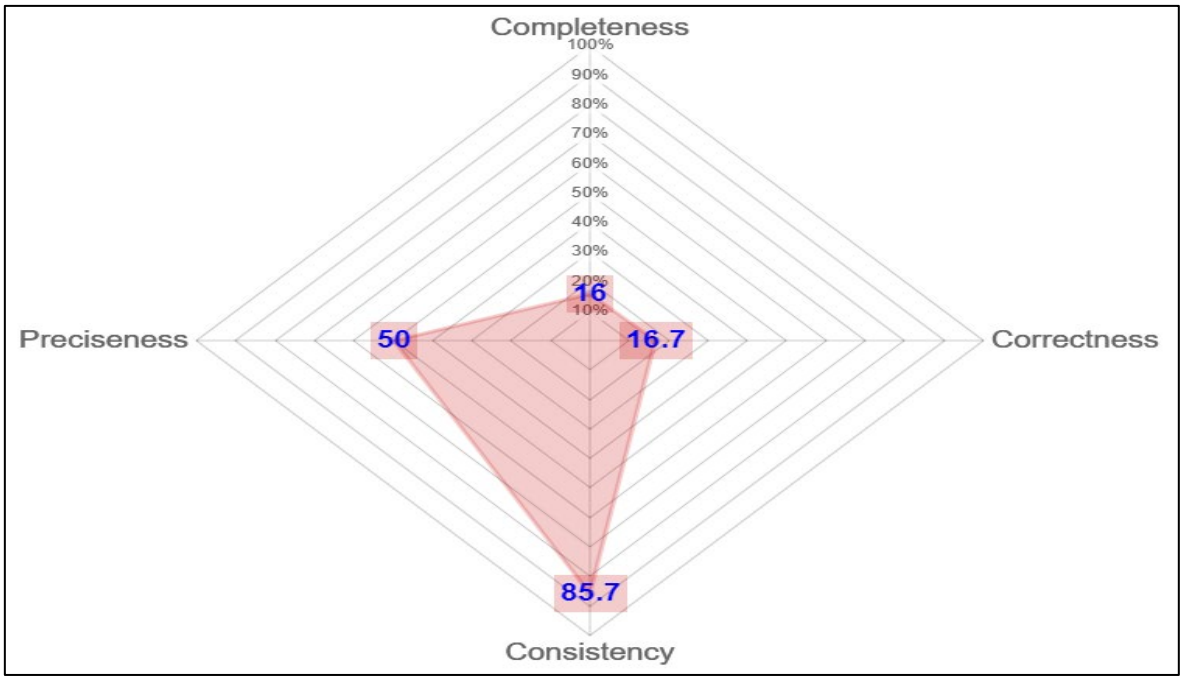


Figure 4.29: Star graph represent the overall quality of assessed SRS document

Finally, the overall quality of the SRS document can be measured. The Star graph in Figure 4.29 visualizes the percentage of each quality. The proposed rule to evaluate the overall quality of the SRS document had been implemented in the prototype (Equation 3.10).

$$Overall\ Quality = \left(\frac{16 + 16.7 + 50 + 85.7}{4}\right)$$

$$Overall\ Quality = \left(\frac{168.4}{4}\right)$$

$$Overall\ Quality = 42.1\% \qquad \text{Equation 4.5}$$

The result of each quality assessed from Figure 4.29 is sum together and divided by a constant of four since there are four qualities used to assess a single SRS document. The result from the overall quality shows the maturity of the SRS document. Based on the measurement (Equation 4.5), the overall quality of the assessed SRS document sample is 42.1%. The project manager who owns the SRS document can use the result to identify which part of the document needs to improve as the document need to reach a certain maturity stage before proceeding to the next phase in SDLC.

4.5 Summary

In summary, it is shown that the proposed framework and equation had been implemented in the prototype. The process is started by the cleansing of the raw data extracted from the inputted document. The process flow to cleanse the raw data are shown in the activity diagram. Knowledge-based consists of libraries proposed to increase the reliability of the pattern-matching process. The minimal interference of the <*user*> resulted in the minimization of human error. Rules and equations proposed in Chapter 3 had been implemented and automized. The prototype is responsible to automatically calculate the overall quality of the SRS document based on the assessment of quality properties proposed.

CHAPTER 5

DISCUSSION

5.1 Introduction

This chapter discusses the evaluation result of the case-control study and the actual case study. This includes the invalid and valid inputs which give a better understanding of the outcome. The case-control study will contain train data that will evaluate the capability of the proposed framework. The case-control study will only involve a single SRS document. Meanwhile, two SRS documents taken from the collection of gathered SRS documents became an actual case study. Results from the evaluation will impact the capability of the proposed framework.

5.2 Case-control Study

In this section, a case-control study is applied to the prototype. The case-control study is designed to accomplish the third objective which is to evaluate the proposed framework based on the software artefacts. This case-control study becomes a controlled study to understand the prototype response for each input. This case study consists of train data. The train data is composed of a single SRS document. The self-written method had been used to collect the data. The purpose of this case-control study is to study the limit and capability of the prototype in handling the input. The evaluation of the case study are shown within the next two subsections. The first subsection will discuss an evaluation of the structural assessment consisting of a mixture of IEEE 830 topics as well as additional topics. Meanwhile, the second subsection will discuss an evaluation of the FR assessment.

5.2.1 Structural Assessment

A document consists of topics to be assessed is created and named as structural. The document is produced in .docx format as the prototype is only able to read .docx format. Shown in Figure 5.1 is the sample of the structure to be assessed. The process to get the measurement for the structural assessment only needs to be run only once by the prototype. The process started by inputting the document into the prototype and the text is extracted. The extracted text is cleansed and pattern-matching is done on each of the topics.

1. Introduction
- ***1.1. Purpose***
- ***1.2. Scope***
- ***1.3. Definition, Acronym and Abreactions***
- ***1.4. References***
- 1.5. Problem
- 1.6. Overall Description
- 1.7. Overview

2. Product Perspective
- ***2.1. Product Functions***
- ***2.2. User Characteristics***
- 2.3. Intended Audience
- ***2.4. Constraints***
- ***2.5. Assumptions and Dependencies***
- ***2.6. Apportioning of Requirements***

3. Specific Requirements
- ***3.1. External Interfaces***
- ***3.2. Functions***
- ***3.3. Performance Requirements***
- ***3.4. Logical Database Requirements***
- ***3.5. Design Constraints***
- ***3.6. Software Requirements Attributes***
- 3.7. Acceptance Criteria
- 3.8. Use Cases
- 3.9. Business Scenarios

4. Supporting Information
- ***4.1. Table of Content and Index***
- ***4.2. Appendixes***
- 4.3. Annexure

Figure 5.1: Sample of structural assessment and its result

Figure 5.1 shows the sample of the structure to be assessed and its result. The process to cleanse the text into a singular sentence has been discussed in Chapter 4. Knowledge-

based containing IEEE 830 topics and synonym topics are matched with the document to search for the similarity. The prototype first matched the sentence with the knowledge-based containing IEEE 830 topics. If successfully matched, it will be denoted as one. Otherwise, it will undergo a second matching process which involves knowledge-based containing synonym topic. If matched, then it will be denoted as one otherwise zero. The sentence is considered a match if and only if the sentence only contains words similar as stated in knowledge-based. The unmatched topic will be ignored as the prototype cannot determine it is part of the IEEE 830 topics and its synonym topics. The differences study domain impact the number of topics in the SRS document. It causes the presence of additional topics aside from the topic suggested in the IEEE 830 standard. The unmatched topic is considered an additional topic that is not part of the study. The prototype is not designed in such a way it can differentiate the value of additional topic based on the SRS document study domain.

Better visualization of results can be seen in Figure 5.1 where bolded and italic topics are similar to topics in IEEE 830. Meanwhile, the topics that are not bolded and italic is treated as an additional topic that is not part of IEEE 830 topics or even the synonym topics. This is due to the prototype is designed only to check similarity for IEEE 830 topics and its synonym topics. For example, the first topic to be assessed from Figure 5.1 is '1. Introduction'. Generally, if the translation matrix is used, those assessed topic is matched as one of IEEE 830 topics. Meanwhile, to ensure the prototype recognizes the topic automatically, the topic to be assessed must be identical to one in the knowledge-based. The process is started where the prototype needs to extract and cleanse the raw topic gathered from the SRS document. The cleanse topic will look like this 'introduction'. This cleansed topic is identical to one of the topics in the knowledge-based. A similar process is done for

the whole topic to be assessed. The number of matched topics is calculated and the results are stored in knowledge-based.

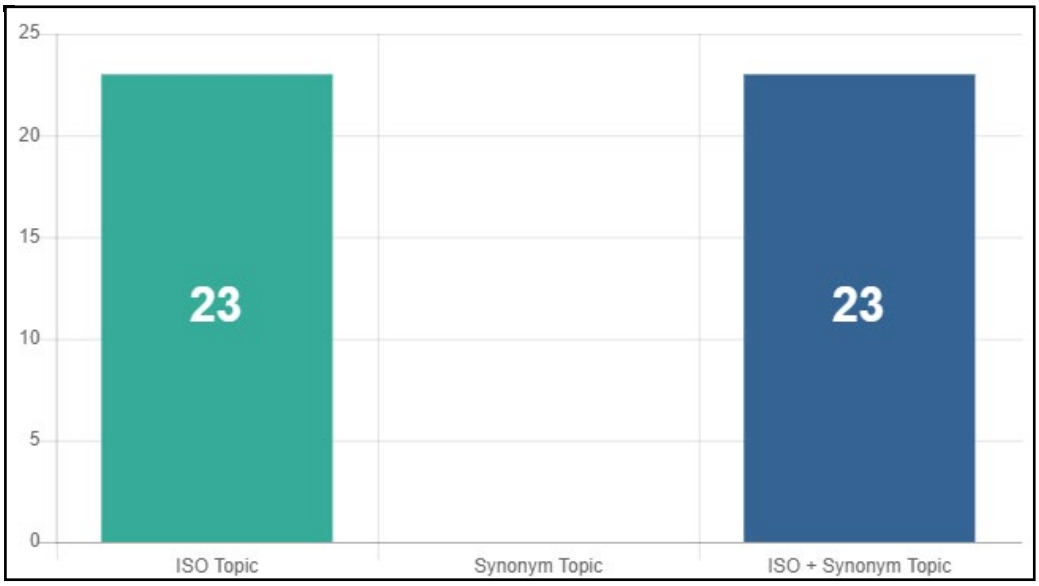


Figure 5.2: Graph preview structural assessment result

Figure 5.2 is the result of the structural assessment. Data visualized in Figure 5.1 is converted into a bar graph as presented in Figure 5.2. A bar graph is provided where the topics matched with IEEE 830 topics are in the IEEE Topic bar. Meanwhile, for the topics matched with synonym topics, it will be on the Synonym Topic bar. The last bar only to present the max number of the topic if the amendment is done as suggested. The suggestion is to convert the present of the synonym topic based on the topic suggested in the IEEE 830 standard. The first bar is resulted in, twenty-five topics found similar to IEEE 830 topic. In Figure 5.1, each of bolded and italic topics represents a single match. It totals up the sum of twenty-five as presented in the first bar of Figure 5.2. The second bar results in the number of synonym topics found. Since there is no synonym topic matched during the pattern-matching process, so the result for the second bar is zero. Lastly, the third bar results in the possible number of IEEE 830 topics found once the user adjusted the document based on the synonym topic found. The result on the third bar is dependent on the number of topics

presented in the first and second bars. The sum of the number in the first and second bars resulted in twenty-three.

5.2.2 Functional Requirement Assessment

Similar cases to the previous subsection, a document consist of the FR were produced. The document is produced in .docx format and input into the prototype. Nine numbers of FR stated as listed in Table 5.1. Based on the number of SRS documents studied in Chapter 3, most of the FR is not refine based on the semi-formal template as such as RB. Numerous ways to define FR lead to data cleansing before refinement to the RB template as discussed in Chapter 4. Table 5.1 shows that each of the FR is defined in a different way to test the prototype capability to refine each of the inputted into the RB template.

Table 5.1: List of the functional requirement

Function ID	Functional Requirement
[F1]	The user should be able add new data into the system.
[F2]	The system able manage the user data.
[F3]	The user should be able to view their own data
[F4]	The user should be able to edit their own data.
[F5]	Sign Up: The guest can sign up to be user of the system.
[F6]	The admin can monitor all user activity.
[F7]	**1.** The user can search activity by date.
[F8]	Guest sign up before use the system.
[F9]	The admin able to update all activity.

Based on Table 5.1, shows a number of FR that will be assessed by the prototype. There is nine number of FR stated in Table 5.1. *[F1]* basically follows the RB template

meanwhile the other is randomly created. This assessment only involves three number of stakeholders which are user, system and guest.

Each of the FR will be refined based on the RB template. Each of the FR yields different results for refinement. Refinement by the prototype is based on data cleansing as discussed in Chapter 4. First, the text is extracted from the document. Second, the text is tokenized and each word is singularized. Third, the text is spliced into single sentences based on the presence of full stop punctuation. Fourth, each sentence is checked for the presence of defined stakeholders; user, system and guest. Fifth, the sentence is once again tokenized to identify POS in each word focusing on the verb. Lastly, the result of the identification of the presence of stakeholders and verbs will be used to fulfil the RB template used in the prototype.

Table 5.2: Actual result refinement functional requirement

Function ID	Refined Functional Requirement
[F1]	The user should be able add new data into the system.
[F2]	The system should be able manage the user data.
[F3]	The user should be able view their own data The user should be able to edit their own data.
[F5]	The guest should be able can sign up to be user of the system.
[F6]	The user should be able can monitor all user activity.
[F7]	The user should be able can search activity by date.
[F8]	**1.** The guest should be able sign up before use the system.

The refinement result in Table 5.2 shows the capability of the prototype to convert the original FR into the RB template format. Let use *[F6]* as an example. In general, the

expected result of the conversion will include the correct use of grammar. The expected result is 'The user should be able monitor all user activity'. However, the prototype is designed to follow a certain template format. The template used is as follow:

The <*stakeholder segment*> should be able <*capability segment*>.

That template resulted in the refinement of *[F6]* that can be seen in Table 5.2. This is due to the code design to capture the <*stakeholder segment*> and <*capability segment*> did not include the understanding of the grammar rule.

Shown in Table 5.2 is the result of refinement FR done by the prototype. The original sentence in Table 5.1 has been refined into the RB template which can be seen in Table 5.2. Based on the results in Table 5.2, *[F1]*, *[F2]*, *[F5]*, *[F7]* and *[F8]* are successfully refined from its original state into the RB template. For example, *[F7]* in Table 5.1 consists of numbers at the beginning of the sentence which needs to be cleansed. As discussed in Chapter 4, that sentence needs to be cleansed. Initially, the sentence is tokenized for the singularization process. The singularization process converts each of the plural words into a singular word. This will increase the reliability of the pattern-matching as knowledge-based used are built using a singular word. Upon completion of the singularization of a word in a sentence, the sentence undergoes pattern-matching to identify the stakeholder. An example of *[F7]* will yield the result of *'user'* as a stakeholder. Next is to identify the role of the stakeholder mentioned. The prototype uses a POS tagger to tag the verb from the sentence. Any word identified in between of the verb or after the verb will be considered as part of the role. For example, the *[F7]* will yield the result of *'can search activity by date'*. The word *'can'* and *'search'* is tagged as a verb but the word *'activity by date'* is an additional information. Those results will be used to fill in the RB template. Finally, it is resulted in:

The <*user*> should be able <*can search activity by date*>.

The refinement for *[F3]* and *[F4]* is unsuccessful due to the absence of a full stop punctuation mark at the end of *[F3]* sentence. This is due to the rule applied to split the text into single sentences is based on the presence of a full stop punctuation mark. The refinement for *[F6]* is successfully due to it meeting all of the properties. The refinement for *[F9]* is unsuccessful due to the failure to detect the presence of possible stakeholders. This is due to '*admin'* is not part of stakeholder which resulted in no refinement.

The prototype is designed where each of the FR is refined based on the RB template. The template requires two properties. First is the presence of possible stakeholders. Second is the presence of a possible process verb with additional detail.

The first properties of FR refinement are also part of the consistency properties that will be measured. The <*user*> can define as many possible stakeholders, but the measurement is designed where each of the defined stakeholders must be used in defining the FR at least once. Based on the case study, only three stakeholders are defined.

The second property measured in consistency is the role of each refine FR. The prototype is designed where each of the successful refined FR will be compared with each other. As discussed in Chapter 4, the comparison is done based on two techniques which are similar_text() technique and word similarity. Table 5.3 shown is the result of a possible comparison.

The similar_text() technique calculates the percentage of transformation needed to convert the refined FR into the suggested FR. The higher the percentage means that the higher the similarity of the assessed text with targeted text.

Table 5.3: Possible similar functional requirement based on percentage

Function ID	Technique (percentage)	Functional Requirement Comparison
[F1]	Word Similarity (45%)	The user should be able can search activity by date.
	Similar Text (87%)	The guest should be able can monitor all user activity.
[F2]	Word Similarity (56%)	The user should be able can search activity by date.
	Similar Text (77%)	The user should be able can monitor all user activity.
[F3]	Word Similarity (50%)	The user should be able can search activity by date.
	Similar Text (71%)	The user should be able add new data into the system.
[F5]	Word Similarity (71%)	The guest should be able sign up before use the system.
	Similar Text (87%)	The guest should be able sign up before use the system.
[F6]	Word Similarity (70%)	The user should be able can search activity by date.
	Similar Text (77%)	The user should be able can search activity by date.
[F7]	Word Similarity (50%)	The user should be able add new data into the system.
	Similar Text (77%)	The user should be able can monitor all user activity.
[F8]	Word Similarity (82%)	The guest should be able can sign up to be user of the system.
	Similar Text (87%)	The guest should be able can sign up to be user of the system.

Shown in Table 5.3 under similar text is the result of FR highest percentage similarity based on the number of similarities per character. Example usage of similar text techniques can be represented by referring to the *[F5]* result in Table 5.3. The result shows the final comparison between *[F5]* and *[F8]*. The *[F5]* will be the assessed sentence which will be

compared with *[F8]*. The *[F5]* and *[F8]* are tokenized into per character. Shown in Table 5.4 is the tokenization of *[F5]* and *[F8]*.

Table 5.4: Result comparison and tokenization of *[F5]* and *[F8]*

Function ID	Functional Requirement
[F5]	***The guest should be able*** can ***sign up to*** be user of ***the system.***
[F8]	***The guest should be able sign up*** before use ***the system.***

Based on Table 5.4, the italic and bolded text are the characters that are found as match and similar. The formula used to calculate the similar text technique is:

$$St = \frac{Fc * 200}{Nc_1 + Nc_2}$$ Equation 5.1

Based on the formula in (Equation 5.1), the percentage of similarity can be calculated. *Fc* represents the number of characters that are similar to the assessed sentence. *Nc1* represents the number of characters in the assessed sentence. *Nc2* represents the number of characters in a comparison sentence. Based on Table 5.4, the number of similar characters is 51. The number of characters in the assessed sentence is 62. The number of characters in a comparison sentence is 55. By using the formula in (Equation 5.1), this resulted in 87%.

Concerning on word similarity technique, the assessed and targeted text is tokenized into a single word and is compared. Each of the similar words found will be counted. Shown in Table 5.3 under word similarity is the FR with the highest degree in percentage number of similarities per word.

A useful example of a word similarity technique is by referring to the *[F7]* result in Table 5.3. which shows the results of the final comparison is in between *[F7]* with *[F1]*.

Word similarity technique can be simplified by finding similarity of words by comparing two sentences. Shown in Table 5.5 is the comparison of *[F7]* with *[F1]*.

Table 5.5: Result comparison and tokenization of *[F7]* and *[F1]*

	Tokenization											
Functional Requirement	[F7]	*The*	*user*	*should*	*be*	*able*	can	search	activity	by	date	
	[F1]	*The*	*user*	*should*	*be*	*able*	add	new	data	into	the	system

The word similarity technique process is quite simple. The *[F1]* will be used to assess the *[F7]*. First, the *[F7]* and *[F1]* are tokenized by word. Then each of the tokenized words from *[F7]* and *[F1]* are compared with each other. Similar or matched word is counted. Shown in Table 5.5, the word found and matched is in italic.

$$Ws = \left(\frac{Sw}{AFr}\right) * 100\% \qquad \text{Equation 5.2}$$

Formula to calculate the percentage of similarity can be seen in (Equation 5.2). Sw represents the number of similar words found or matched. AFr is the number of words in the assessed sentence. Based on the formula in (Equation 5.2), the number of similar words found is 5. Meanwhile, the number of words in the assessed sentence is 10. The percentage is calculated and resulted in 50%.

Likewise, the measurement for the consistency quality of FR is determined by the number of conformations done manually by the <*user*>. The prototype is only designed to calculate the percentage of similarity between each FR based on character and word used, which disregard the meaning of the sentence. This is the reason that the final result is based

on <*user*> conformation. The prototype will only show the nearest possibility of similarity by showing the percentage and it corresponds FR but the <*user*> will decide either to follow based on percentage written or not.

Based on the equation proposed (Equation 3.5), let's say that a total of non-similar functions conformed by the <user> is five out of eight functions, then it will result in 62.5% of consistency quality.

In terms of correctness quality, test case generation is used to validate the FR. The test case which was developed at an early stage such as the requirement phase is quite challenging without the proper requirement specification. The advantage of developing test case on an early stage is to understand the logic flow of each defined FR. Numerous software testing techniques can be used. In this study, use case testing technique is adopted since it is almost similar to user story technique. There are two number of test cases developed. First is the valid test case which will result in a correct use of test case. Second is the invalid test case which results in the wrong use of data in test case. Both test case will be discussed as follow:

Use Case :	The user should be able can search activity by date.
Actor :	user
Precondition :	The user login into the system.
Normal Flow :	1. The user click on project. 2. System view the project detail. 3. The user click on the activity. 4. System view the list of activity detail. 5. The user search the activity by date. 6. System view the list of activity based on specified date.
Alternative Flow :	5. (a) The user click on date sorting. (b) System sort the activity list based on date. (c) The user search for the activity manually.
Postcondition :	System view the specified activity in detail.
END :	The user should be able can search activity by date.

Figure 5.3: Correct test case

The '*Use Case*', '*Actor*' and '*END*' element is specified automatically by the prototype. The '*Precondition*', '*Normal Flow*', '*Alternative Flow*' and '*Postcondition*' elements needed to be inputted manually by the <*user*>. Based on Figure 5.3, the test case shows the presence of the '*Actor'* element in the *'Normal Flow'* as well as the presence of word similarity for the possible interface design. The highlighted word is the possible interface design. The test case is counted as valid with the presence of *'Actor'* element in the *'Normal Flow'* as the presence of term representing the possible interface design.

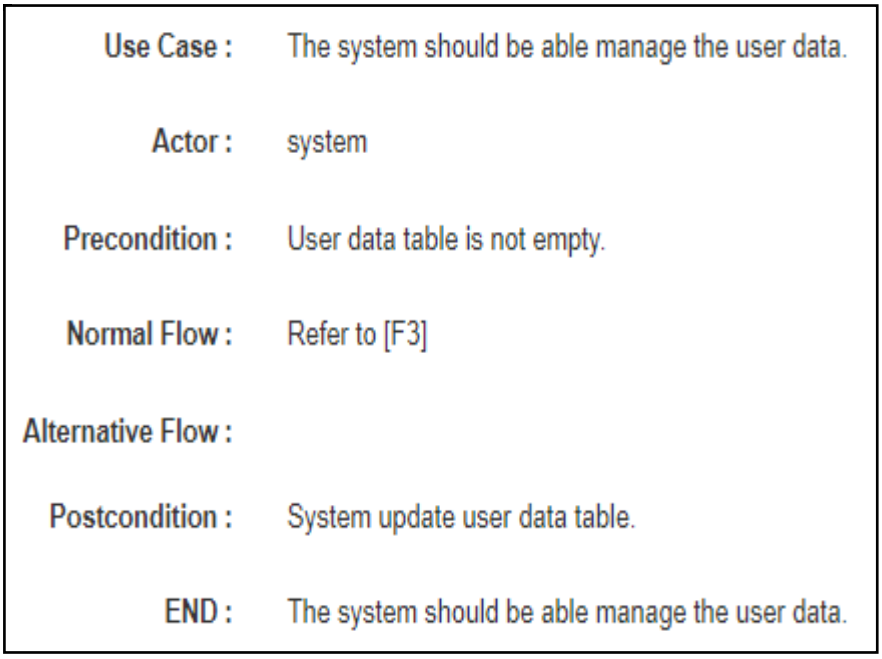


Figure 5.4: Incorrect test case

Due to the assessment only involves the presence of possible interface design and stakeholder properties in '*Normal Flow*' element, the test case in Figure 5.4 is considered invalid. The prototype is designed where it should be able to assess the presence of 'Actor' element and term representing possible interface design in the *'Normal Flow'* element. Shown in *'Normal Flow'* element in Figure 5.4, the input is 'Refer to [F3]'. The prototype is not designed where it can detect the referral function. Aside from that, those inputs are considered ambiguous where <*user*> is unable to represent the flow in *'Normal Flow'* element. The <*user*> should be able to represent the flow of the assessed function in detail to avoid ambiguity.

Based on the rule for correctness quality (Equation 3.6), the assessed test case considered valid if there are present of '*Use Case Name*', '*Actor*' and '*Normal Flow*' element. As the '*Use Case Name*' and '*Actor*' element is automatically processed by the prototype, there is a concern on how to ensure the <*user*> manual input '*Normal Flow*' element is correct. The prototype had been designed in such a way that the data input in the

'*Normal Flow*' element must contain the '*Actor*' element and present of the word representing possible interface design.

The validity of test case is calculated individually based on rules proposed in (Equation 3.6). For example, test case in Figure 5.3 shows the presence of '*Use Case Name*', '*Actor*' and '*Normal Flow*' element. The presence of each element will be rewarded as one. Then the sum of the rewarded element will be divided with a constant of three which represent three number of elements. If the final result is one then the test case is considered as valid otherwise invalid. The test case in Figure 5.4 is considered invalid due to it did not met pass the validation for the '*Normal Flow*' element. Mention earlier, the '*Normal Flow*' element needs to be clarified. It did not pass the validation because there is no presence of a word representing the possible interface design and stakeholder stated in '*Actor*' element.

The assessment of the preciseness quality properties results in finding the presence of datatype and vague words. Based on the assessment of the provided FR document, the prototype results in only finding the presence of the possible datatype. Shown in Table 5.4 is the possible datatype found based on word similarity in the FR.

Table 5.6: Present of datatype in functional requirement

Function ID	Functional Requirement
[F1]	The user should be able ***add*** new data into the system.
[F3]	The user should be able ***view*** their own data The user should be able to ***edit*** their own data.
[F7]	The user should be able can ***search*** activity by date.

Based on Table 5.6, only three numbers of FR are found which have the presence of possible datatype. Bolded and italic words are words that contain terms representing the possible datatype. Mentioned in Chapter 3 is the knowledge-based containing words related to the term that represents possible datatype. Pattern-matching using regular expression is used to locate the presence of the term representing the datatype in each sentence assessed. In the prototype, the bolded and italic word is highlighted which is clickable which will reveal the possible datatype.

Regarding the vague words, there are no vague words found. The process of detection for vague words is similar to detection for the term representing possible datatype. Regular expression supported with knowledge-based containing vague words is used to assess the sentence. This resulted in a degree of preciseness. The lower the number of present vague words, the higher the degree of preciseness overall of the FR.

A useful example can be seen below on how the prototype reacts with the presence of a vague word in the sentence.

The system should be able to calculate certain ***amount***.

The sentence in the example is tokenized. The regular expression is used to match the tokenized sentence with the knowledge-based of the vague word. Knowledge-based containing vague words are used as input for the pattern-matching. Once matched, the prototype will automatically denote the assessed sentence as zero. Otherwise if not match, denoted as one.

Based on the example, bolded and italic words are words that are considered as ambiguous. The word '***amount***' is considered as part of a vague word in this case due to ii

being not supported with additional information. Additional information to support any vague word is important as it will provide additional data that reduces the ambiguity of the sentence.

The rule to evaluate the preciseness of FR had been proposed in Chapter 3 (Equation 3.8). Based on Table 5.6, function *[F1]*, *[F3]* and *[F7]* is considered as precise because it fulfils the rule which there are present of a term representing possible datatype and non-present of the word representing a vague word. The FR is considered not precise if it did not fulfil the rule.

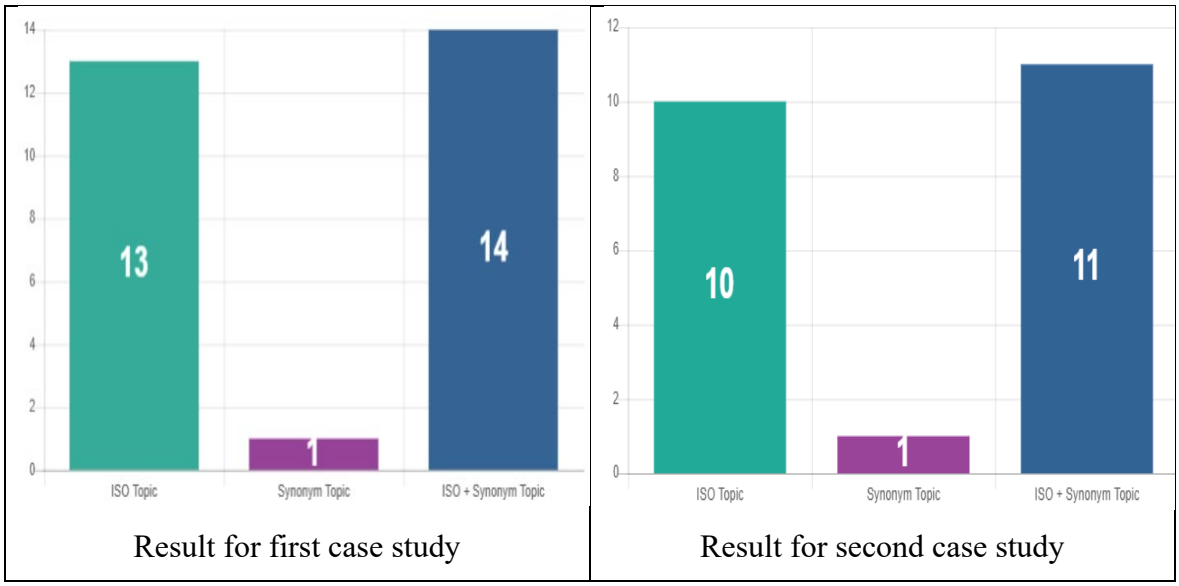
The calculation for the preciseness had been proposed in Chapter 3 (Equation 3.9). Let say the total of conformed non-similar FR is eight and the total of precise FR is 3. The division between a total of precise FR with a total of conformed non-similar FR resulted in 37.5% of preciseness.

5.3 Actual Case Study

There will be two case studies to be discussed in this section. The title of the first case study is Carpool System (Temiz et al., 2014). Meanwhile, the title of the second case study is ITS Adriatic Multi-port gateway (D'Antiga, 2013). The case study document is already in .docx format which is accepted by the prototype. The first and second case study document is selected from the ninety-three collection of SRS document which is downloaded from the internet. This case study is chosen as part of the study due to it meeting the requirement that is to be assessed. The requirement in choosing the case study is the SRS document should have the presence of structure and FR.

5.3.1 Structural Assessment

The prototype automatically assesses the structural once the SRS document is inputted. To better visualize the result, a bar graph is implemented. The bar graph is shown in Figure 5.5.



Result for first case study | Result for second case study

Figure 5.5: Graph result of structural assessment

Shown in Figure 5.5 is the result of the structural assessment for both case studies. In the first case study, the first bar indicates that thirteen out of twenty-five number of topics from IEEE 830 ToC were detected. On the second bar, one synonym topic was detected. On the last bar, a total of fourteen number of topics that <*user*> can consider to be updated. Meanwhile, in the second case study, the first bar indicates that ten out of twenty-five number of topics from IEEE 830 ToC were detected. On the second bar, one synonym topic was detected. On the last bar, a total of eleven number of topics that <*user*> can consider to be updated.

The idea to assess the structural is to ensure the <*user*> fulfill the standard given in the IEEE 830. Only the total of the first bar will be considered by the prototype to be

measured. The percentage of the completeness quality can be increased if the <*user*> would consider the synonym topic in the second bar change according to its corresponding IEEE 830 topic. Although if the <*user*> manages to make changes and convert all the synonym topics according to the topics given in the IEEE 830 ToC, the highest total that can be achieved is fourteen. This is due to the document is still lacks other topics proposed in the IEEE 830 ToC.

Pattern-matching between the synonym topic library is done on the case study document. It resulted in only one match. '*Table of Content*' topic from the case study match as a synonym to '*Table of Content and Index*' topic from IEEE 830 ToC. This allows <*user*> to reconsider the synonym topic detected to be changed as suggested. Amendment can be done on the document and reupload to be reassessed again. However, the highest percentage the <*user*> will get if the amendment is done only according to the suggested synonym topic is fourteen.

Table 5.7: Summary of structural measurement

Case Study	Detected		Possible Update	Calculation (%)	
	IEEE Topic	Synonym Topic		Expected	Actual
1	13	1	14	56	52
2	10	1	11	44	40

Shown in Table 5.7 is the summary of the structural assessment. Based on the first case study the number of IEEE Topic detected is thirteen. Meanwhile, the number of synonym topic detected is one. The highest possible number of IEEE Topic after the update is fourteen. The result for the possible update is yield by adding the value of detected IEEE Topic with the value of detected Synonym Topic. Meantime, as for the second case study,

the total number of IEEE Topic detected is ten. The total number of synonym topic detected is one. Lastly, the total number of the IEEE topic is eleven.

On the calculation side, there are two measurements given. First represents the expected calculation where the value yield if and only if the user updates the SRS document based on the advice of detected synonym topic. Meanwhile second represents actual calculation where the value is only based on the IEEE topics detected. The expected calculation will be ignored by the prototype as it is only a suggestion of possible value after an update on the SRS document. The only part to be calculated based on the prototype is the actual calculation. The idea is to emphasize the use of IEEE 830 topics as standard on the SRS document. There is numerous synonym topic that can represent their corresponding IEEE 830 topics. For example, *'Table of Content'* and *'Index'* are synonym topics that correspond to IEEE 830 topics which are *'Table of Content and Index'*. If *'Table of Content'* and *'Index'* topic existed in the document to be assessed, it is wise to change it according to its corresponding IEEE 830 topic to help in emphasizing the standard.

5.3.2 Functional Requirement Assessment

Assessments for the FR are divided into three quality. Each of the quality will be assessed based on the equation proposed in Chapter 3. The first assessment will be the properties of the consistency. Follow up by the properties of correctness. Then ended with the properties of preciseness.

The document containing FR is inputted into the prototype. Sample of the extracted FR can be seen in Appendix B for the first case study and Appendix C for the second case study. Before the FR assessment, stakeholder needs to be defined first. There is only one stakeholder defined which is *user*. Result extraction from the FR document shows there is

twelve number of FR. Those FR treated as a raw FR. It then undergoes refinement process as stated in Chapter 4 to convert raw FR into refined FR. The refined FR is written based on the RB template. Those also resulted in twelve number of refined FR. All of the refined FR is conformed as consistent.

Table 5.8: Summary of consistency properties measurement

Case Study	Detected	Calculation (%)
	Non-Similar Function	
1	12	100
2	22	100

The summary of the measurement is shown in Table 5.8. It is shown that none of the FR is redundant. The assessed FR is considered consistent if the stakeholder is presented as well as its corresponded role is not redundant with each other. In this case study, there are twelve FR conformed as consistent after filtered based on the actual documentation.

$$T = \left(\frac{12}{12}\right) * 100\%$$

$$T = 100\% \qquad \text{Equation 5.3}$$

Let take the first case study as a sample calculation (Equation 5.3), it resulted in 100%. This results in the number of FR conformed by the <*user*>. The prototype helps in terms of calculating the nearest percentage of similarity meanwhile the <*user*> has the final decision based on the result of similarity. Concerning the stakeholder, there is only one stakeholder defined which resulted in only one to be assessed.

With regard to the correctness quality of the FR, there is twelve number of test case inputted into the prototype. As discussed earlier, the assessment of the test case will only involve the *'Normal Flow'* element. Assessment of the *'Normal Flow'* element required pre-

condition which are assessed during the consistency quality assessment. Pattern-matching is used to assess the *'Normal Flow'* element to ensure the presence of stakeholders and terms represent the possible design.

Table 5.9: Summary of correctness properties measurement

Case Study	Test Case		Detected		Calculation (%)
	Identified	Valid	CRUD	Stakeholder	
1	12	9	9	11	75
2	22	6	6	22	27

The pattern-matching result for correctness quality is shown in Table 5.9. Nine out of twelve number of test cases are valid in the first case study. This shows that there are three test cases that are not defined. The validity of test case is based on the presence of term representing the possible CRUD design and stakeholder in the *'Normal Flow'* element.

$$V = \left(\frac{\frac{1+1+1}{3} + \frac{1+1+0}{3} + \frac{1+1+0}{3} + \frac{1+1+1}{3} + \frac{1+1+1}{3} + \frac{1+1+1}{3} + \frac{1+1+1}{3} + \frac{1+1+1}{3} + \frac{1+1+0}{3} + \frac{1+1+1}{3} + \frac{1+1+1}{3} + \frac{1+1+1}{3}}{12} \right) * 100\%$$

$$V = \left(\frac{1 + 0.7 + 0.7 + 1 + 1 + 1 + 1 + 1 + 0.7 + 1 + 1 + 1}{12} \right) * 100\%$$

$$V = \left(\frac{1+1+1+1+1+1+1+1+1+1}{12} \right) * 100\%$$

$$V = \left(\frac{9}{12} \right) * 100\%$$

$$V = 75\% \qquad \text{Equation 5.4}$$

V model is part of the validation of FR (Wiegers & Beatty, 2013). Assessment of FR by test case will reveal error, ambiguities and omission before the code is written by the programmer. Possible interface design which represented as CRUD shown in the '*Normal Flow'* element. This can be express by using pattern-matching to match between text written in *'Normal Flow'* element with the knowledge-based consist of terms representing the possible design. This allows the traceability between the SRS document with the design document as the V model is adopted in the study. The calculation for correctness quality is based on the validity of the define test case. The formula used for measurement can be seen in Chapter 3 (Equation 3.7). The calculation shown in Equation 5.4 is for the first case study. Notice that the test case for second, third and ninth function filters out from the equation due to it does not meet the requirement which each test case must result as one. Meanwhile, the validity of the other nine test cases is confirmed. So, the equation resulted in a division between nine and twelve. Shown in equation (Equation 5.4), is a sample measurement of correctness quality for the first case study which resulted in 75%.

In terms of the assessment of the preciseness quality properties, the measurement will be done automatically by the prototype. First, the pattern-matching is done in between assessed FR with knowledge-based containing vague words. Then once again, the pattern-matching process is done in between assessed FR with knowledge-based containing terms representing datatype. The presence of the vague words in assessed FR will cause ambiguity. Meanwhile, the presence of a term representing datatype as an additional information can increase the preciseness of assessed FR.

Table 5.10: Summary of preciseness properties measurement

Case Study	Detected		Calculation (%)
	Vague Word	Datatype	
1	0	4	33
2	0	15	68

The presence of possible datatype as part of additional information increases the readability of each FR. Pattern-matching is used to identify the presence of a term representing datatype in the assessed FR. The assessed sentence will be tokenized and each of the tokenized words will become input for pattern-matching. Knowledge-based containing terms representing datatype is used to compare with the tokenized word in the pattern-matching process. Mentioned before, the study has adopted the RB to define FR. Additional information allows a better understanding of each FR. Expressing the possible word which can be interpreted into possible design the database allow user more understanding of the project they are on.

$$P = \left(\frac{\begin{array}{l} \frac{1+1}{2} + \frac{0+1}{2} + \frac{1+1}{2} + \frac{1+1}{2} + \\ \frac{0+1}{2} + \frac{0+1}{2} + \frac{1+1}{2} + \frac{0+1}{2} + \\ \frac{0+1}{2} + \frac{0+1}{2} + \frac{0+1}{2} + \frac{0+1}{2} \end{array}}{12} \right) * 100\%$$

$$P = \left(\frac{\begin{array}{l} 1 + 0.5 + 1 + 1 + \\ 0.5 + 0.5 + 1 + 0.5 + \\ 0.5 + 0.5 + 0.5 + 0.5 \end{array}}{12} \right) * 100\%$$

$$P = \left(\frac{1+1+1+1}{12} \right) * 100\%$$

$$P = \left(\frac{4}{12}\right) * 100\%$$

$$P = 33\% \qquad \text{Equation 5.5}$$

Summarize in Table 5.10 shows the number of present vague words and datatype in twelve numbers of FR. None of FR shows the presence of vague words which can cause ambiguity. It resulted in each of the function denoted one point for the non-present of a term representing a vague word. Meanwhile, only four numbers of FR presented with datatype which resulted in respective function denoted another one point. Then each of the functions is divided by constant of two where two represent the presence of term representing datatype and the non-present of a term representing the vague word. Based on the rule to assess the quality preciseness of FR, both properties must be satisfied (Equation 3.8). All functions that satisfy the rule will result as one during the calculation as stated in the second step of calculation (Equation 5.5). Otherwise, it will be ignored as stated in the third step of the calculation (Equation 5.5). The formula used for the calculation can be seen in Chapter 3 (Equation 3.9). The sample calculation for the first case study resulted in 33% of preciseness quality.

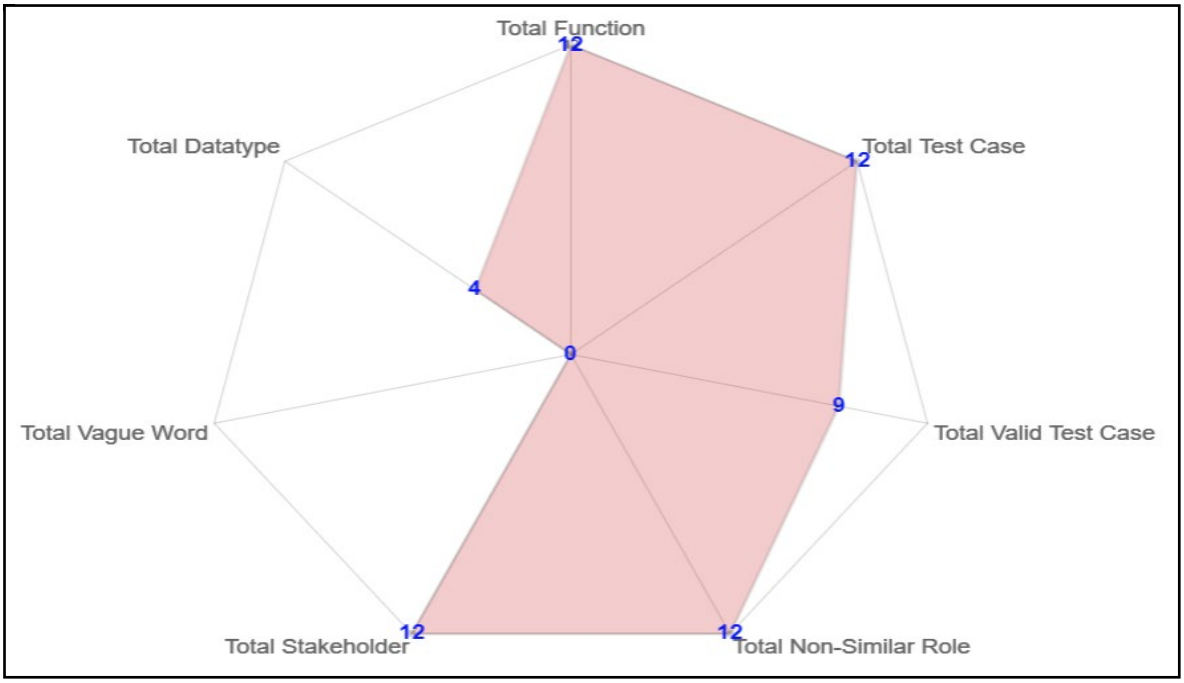


Figure 5.6: Graph summarization quality properties assessment for first case study

Summarization of the FR assessment for first is shown in Figure 5.6. Each quality property's results are shown in the graph form. There is total of six quality properties to be assessed. Based on Figure 5.6, there is total of twelve non-similar functions. Further assessment on the FR resulted in only four totals of datatype found and zero total for vague word found. There is twelve total of test case generated but only nine are a valid test case. Each result is stored in the database and used for measurement. The resulting measurement for each quality will be shown in the next section.

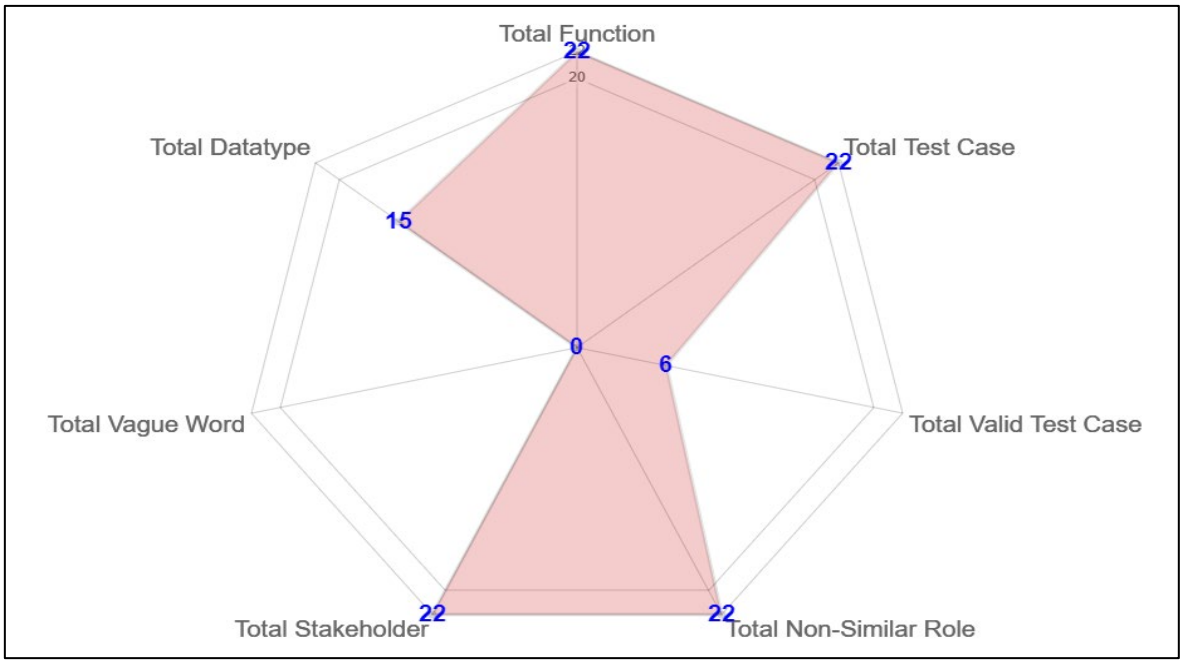


Figure 5.7: Graph summarization quality properties assessment for second case study

Meantime, the summarization of the FR assessment for the second is shown in Figure 5.7. Each quality properties results are shown in the graph form. There is total of six quality properties to be assessed. Based on Figure 5.7, there is total of twenty-two non-similar functions. Further assessment on the FR resulted in only fifteen totals of datatype found and zero total for vague word found. There is a twenty-two total of test case generated but only six are a valid test case. Each result is stored in the database and used for measurement. Similar to the first case study, the result measurement for each quality will be shown in the next section.

5.3.3 Overall Measurement

The result of the calculation can be seen in Figure 5.8 and Figure 5.9. The result in Figure 5.6 of the previous section will be used to calculate the overall measurement of the quality.

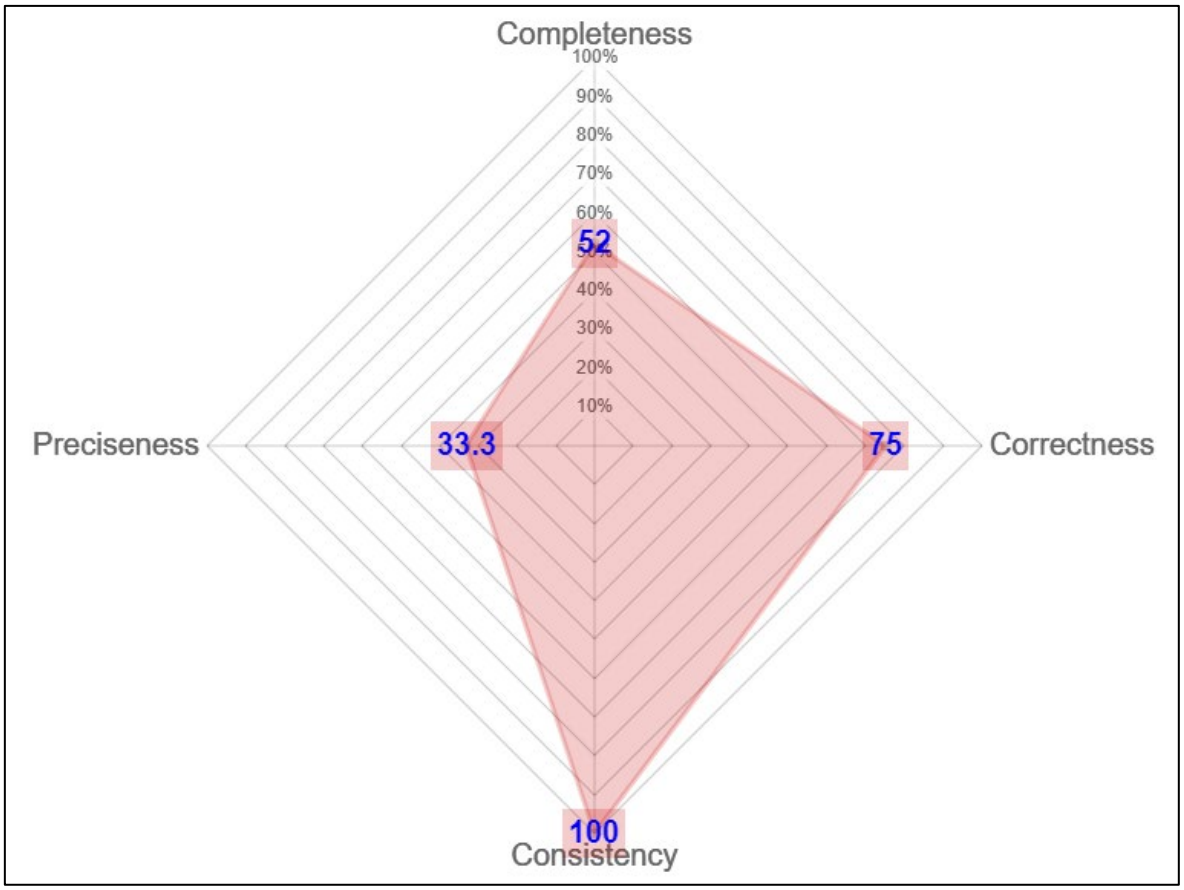


Figure 5.8: Summary degree of each quality assessment for first case study

Based on the calculation in Section 5.3.1 until Section 5.3.2, the result is summarized in Figure 5.8. First, completeness quality yield total of 52%. Second, consistency quality yields a total of 100%. Third correctness quality yield total of 75% and lastly preciseness quality total is 33.3%. The overall measurement for the quality is 65.1%. The formula to calculate the overall quality measurement can be seen in Chapter 3, Section 3.3.3.

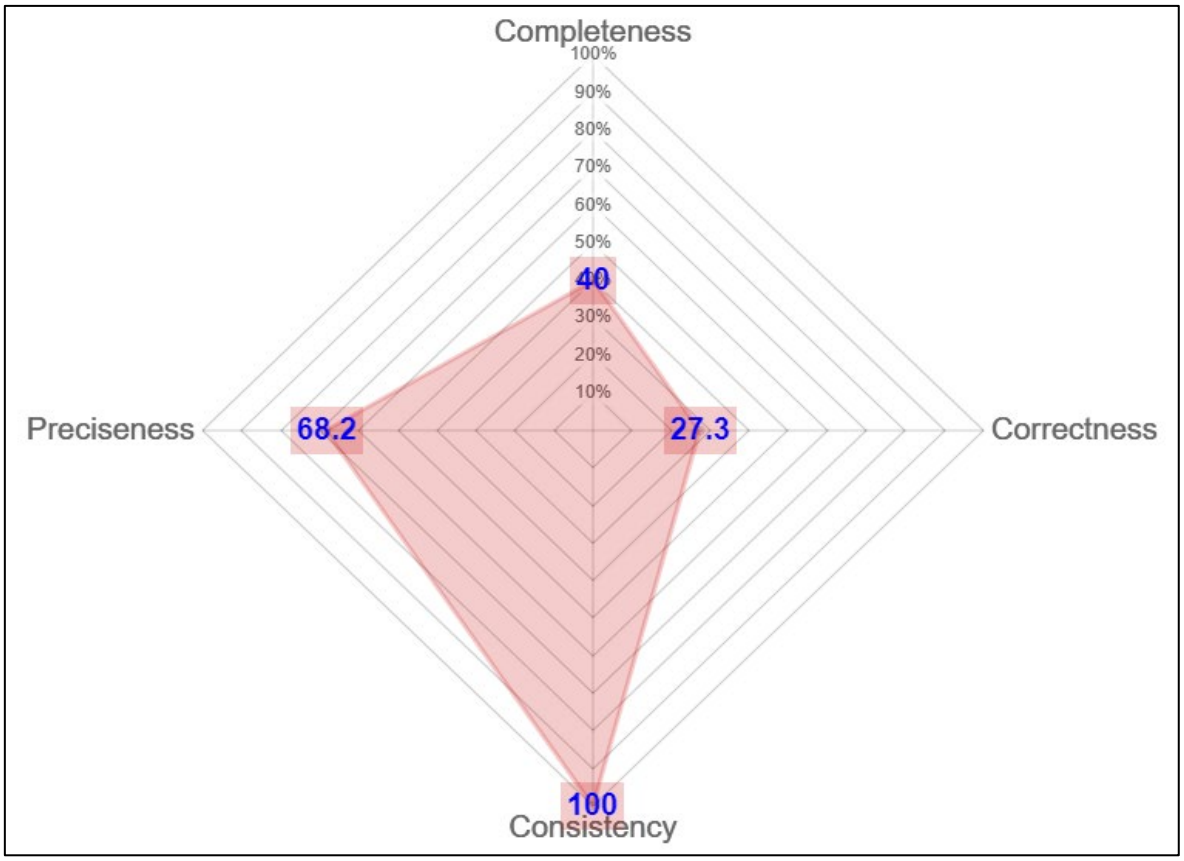


Figure 5.9: Summary degree of each quality assessment for second case study

Based on the calculation in Table 5.7 until Table 5.10, the result is summarized in Figure 5.9. First, completeness quality yields a total of 40%. Second, consistency quality yields a total of 100%. Third correctness quality yield total of 27.3% and lastly preciseness quality total is 68.2%. The overall measurement for the quality is 58.8%.

5.3.4 Discussion

The study on the structural stresses on the importance of topics gathered in IEEE 830 ToC. The case study resulted in the only portion of it follows the topics proposed in the IEEE 830 ToC. The assessment on the structural completeness resulted in 52% for the first case study and 40% for the second case study. Thirteen out of twenty-five number of IEEE 830 topics defined is found as matched in the first case study. One topic is matched as a synonym topic. The topic stated in the synonym topic can be changed to the topic suggested in the

IEEE 830 standard. This will increase the percentage of the result. Otherwise, the failure to change will cause it to be treated as an additional topic. It is understandable that each of the SRS document domains had their own focused topic which treated as an additional topic in this study. The standard topic proposed in IEEE 830 should be followed as a minimal requirement for each of the SRS document structuring. Numbers of researcher recognize the IEEE 830 standard lack of additional topic to be focus on for each of SRS document produced (Georgiades & Andreou, 2010; Saito et al., 2014; Kamalrudin & Sidek, 2015; Takoshima & Aoyama, 2015; Thitisathienkul & Prompoon, 2015; Slhoub et al., 2017). However, the researchers recognize the benefit of following the standard topic suggested in IEEE 830 which is to ensure readability between one developer to another.

Each of the FR defined must be unique. This is to ensure the consistency of each FR. The first step to ensure the uniqueness of each FR is to fully understand the define stakeholder. The next step is to ensure the role of each stakeholder is not similar. According to the case study, improper defining or understanding the stakeholder level impacts the quality of consistency. Concerning the role, the prototype is designed in helping the user to check the similarity of each FR. Shown in the first and second case studies, the stakeholder and its role are nicely defined and unique which resulted in 100% quality of consistency. A recent study was done by Ahmed et al. (2018) show improvement in evaluating the FR if it was written in a structured way. Similar to this case study, the raw FR is refined based on the RB template which is a structured way in writing an FR. Refined FR based on RB allows a better understanding of the stakeholder level. Next is to ensure the non-similarity between each FR which leads to consistency. Compared to the study done by Ali et al. (2018) and Ferreira et al. (2018), this research adopts a similarity check method. The similarity check process is automated by the prototype, allow the <*user*> to view the nearest similar between

assessed FR. Similar_text() algorithm adopted in this research is an improvement from the Levenstein method used by Audytra et al. (2016). The benefit of similar_text() algorithm is more length of sentence can be used as the written FR in such there is no length limit. Furthermore, similar_text() algorithms give more accurate results compared to the Levenstein method (Data, 1998).

The FR is validated throughout the presentation of test case. A present of test case defines the <*user*> understanding of FR in order to define the proper *'Normal Flow'* element of test case. Studies done in Chapter 2 shows that developers are encouraged to define the test case as early as the requirement phase. V model has suggested test case can be done on early as requirement phase. In the study, the use case testing had been implemented. The result will impact the correctness quality of each FR. Aside from that, the presence of the V model allows traceability between one document to another. The assessment of correctness quality properties in the case study above was impacted by the validity of each test case presented. The assessment resulted in a 75% degree of correctness quality for the first case study meanwhile 27.3% for the second case study. The validity of each test case depends on the data inputted in the '*Normal Flow*' element. The prototype is designed in such that the '*Normal Flow*' element evaluated by the presence of stakeholders from '*Actor*' element and term representing present of possible CRUD design. This is an improvement compared to the test case generated by the Ansari et al. (2017) and Ahmad et al. (2018). Compared with Ahmad et al. (2018) test case generation which only focuses on the presence of an actor in the '*Normal Flow*' element, a term representing present of possible interface design is a property that had been proposed to increase the reliability check of the '*Normal Flow*' element. Ansari et al. (2017) proposed an automated test case generation based on the presence of the conjunctive statement. However, there is no indication of how the '*Normal*

Flow' element should be validated. Based on Table 3.3, the frequent presence of test case in the collection of the SRS document shows awareness of the developer toward the importance of validating written FR as early as the requirement phase.

The adoption of RB in the study gives a positive impact on the preciseness quality of FR. RB offers restriction in usage of the language to define the FR. The restriction is supposed to limit the presence of vague words. Meanwhile, the pattern-matching of a term representing datatype is to meet the additional information element of RB. This will ensure the FR is more precise as well as increase the traceability between the CRUD design element in correctness properties with database design. The case study resulted in a 33.3% degree of preciseness quality for the first case study meanwhile 68.2% for the second case study. The case study showed none vague words were found. However, there are only four FR shown matched with the term representing the possible datatype. This means that <*user*> needs to define the correct datatype for each defined FR which impacts the preciseness quality. Similar ideas with Kocerka et al. (2018) and Osman and Zaharin (2018), a list of terms represent vague words are collected. Those collections then became an input for the pattern-matching process. However, another property had been proposed which is the presence of a term representing datatype. The purpose of this property is to ensure the term used by the <user> in writing an FR is meaningful. The idea results from Omoronyia and Stålhane (2017), where the term used in writing the FR must be specific and representable. Based on Table 3.3, the frequent presence of term represents datatype in the collection of the SRS document shows awareness of the developer toward the important usage of the specific and representable term in written FR.

5.4 Summary

This chapter has demonstrated the evaluation of the prototype. The result evaluation of the case study shown the importance of the IEEE 830 standard as well as the RB template to evaluate the SRS document. Knowledge-based contains a library of terms and IEEE 830 topics increase the reliability of the result. The case study that discusses the previous section can be improvised. First, the topic should be refined into the topics as stated in the IEEE 830 standard. Second, the FR should be more precise where the terms used in the sentence must be specific. This includes how the term in the sentence can represent something valuable for the other team member for the developer: designer, programmer and so on. Third, the developer needs to fully understand the requirement that is stated. *'Normal Flow'* element in the test case should represent the assessed FR.

CHAPTER 6

CONCLUSION

6.1 Introduction

The aim of this chapter is to summarize all activities done throughout the study. Those activities compose of achievements based on the proposed quality model, contribution alongside the study, limitation of the framework and future work.

6.2 Achievement

One major issue in software development is the proper understanding of requirements. These can be achieved by writing up the SRS document. This study focuses on measuring the quality of the SRS document. Four number of qualities had been proposed: completeness, consistency, correctness and preciseness. Each of the qualities will inherit its own properties. The assessment is done based on the properties inherited by each of the quality.

Since the study focuses on the SRS document, there are few elements in the document such as structure, FR, term, NFR and font attributes. This study focused on only two elements which are the document structure and its FR. The completeness quality is used to assess the structure of the SRS document meanwhile the other qualities were used to assess the FR.

The quality model shows the mutual influence in between of each quality. It proves the existence of an interaction between the quality that used to assess the SRS document. Each of the assessed qualities had its own properties. Completeness quality property is measured using the topics in IEEE 830. Consistency quality properties are measured based

on the presence of stakeholder and role on each defined FR. Correctness quality properties are measured based on the presence of stakeholders and terms representing the possible interface design in the *'Normal Flow'* element of the test case. Preciseness quality properties are measured based on the presence of vague words and terms representing possible datatype in each defined FR.

The further review had been done on the collection of the SRS document. The collection of the SRS document is from a various number of domains. Reviews showed that heterogeneous domains cause SRS documents produced to not comply with the standard. Even though each of the SRS documents is produced differently, it still inherits similar elements such as the presence of structure, FR and purpose of the study. As the study, the only concern about the structural and the FR of the SRS document, the other elements will be disregard as future work. The review also resulted in the negligence of standards to produce the SRS document. This negligence impacts the quality of the produced SRS document.

The study adopted the IEEE 830 (1998) standard to standardize the structure of the SRS document. For the purpose of measurement, twenty-five topics were extracted from IEEE 830 (1998) ToC and became a standard to assess the completeness quality of the SRS document structure.

RB is a semi-formal template used to define the FR. This study proposes the use of RB to restrict the use of natural language to define FR. Once the refinement from common text to RB based template is completed, the number of quality properties used to assess the refined FR: stakeholder, role, datatype, vague word, test case and possible design. Those

quality properties are assigned to their own respective quality which is consistency, preciseness and correctness.

A framework is proposed in order to better visualize the idea of formalization of the qualitative to quantitative measurement. The activity diagram is used to define the proposed flow of the prototype in detail. Those activity includes the interaction between <*user*> and <*system*> as well as data cleansing. Data cleansing is an important part of the prototype as it cleans the raw data extracted from the inputted document.

Likert Scale method is used to formalized each of the quality properties. Those are either one, for successfully found and zero, for otherwise. The proposed equation discusses in Chapter 3 used to measure the quality of the SRS document. For the assessment of the structure, applying the IEEE 830 ToC as a standard is the best way to ensure the completeness quality of the document. A similar method had been used by the other researchers which yield a good result. In contrast to the assessment of the FR, there are few quality properties that had been proposed. RB had been applied as a standard in defining the FR. The first assessment of the FR involved consistency quality properties. Consistency quality properties had been proposed to be assessed based on the defined stakeholder and role. RB template expresses the importance of the stakeholder and its additional information. An assessment of the consistency quality, the prototype is designed where two techniques have been implemented to find the closest related FR and the <*user*> had the final decision on the assessment. The purpose of the comparison is to locate the nearest percentage similarity of role with the assessed FR. The second assessment is the correctness quality properties which involves the development of test case. Meanwhile, the final assessment is

the preciseness quality properties which involve the presence of vague words and terms representing possible datatype.

A prototype is built based on the proposed framework and equation. A case-control study and actual case study had been inputted to the prototype to evaluate the result. A case-control study consists of a number of train data that had been adjusted accordingly. Meanwhile, the actual case study had been chosen randomly from one of the SRS document collections. Results from both case studies had been evaluated.

6.3 Contribution

The first contribution of this research is to develop a prototype that can be used by the developer to measure the qualities of the SRS document. This prototype can help the developer to measure the maturity of the assessed SRS document based on the result. The developer also able to identify which property needs to improve as the overall result is composed of four qualities.

The second contribution is the automation process to measure the SRS document structure. The prototype is designed to automate the process to measure the SRS document structure. This automation benefits the developer in terms of time consuming to verified their topic against the topic suggested in the IEEE 830 standard.

The third contribution is the automation process refinement of raw FR to the structured way in writing FR. The prototype is designed in such that it helps to convert the raw FR into a structured way of writing FR based on the RB template. The refinement FR benefits the developer to understand the stakeholder focus needs.

The fourth contribution is the assessment of the refined FR. The property for each quality had been proposed. Based on the property rule are proposed in order to control its connection with other quality. Conversion of the rule into equation and implementation of it in the prototype resulted in a much more reliable end result for the FR assessment.

The last contribution in this research is toward the software developer communities. The prototype is designed in such a way that the quality of the SRS document can be measured. Besides that, the developer is able to view which property needs to be a focus on in order to produce a high quality document. The result of the measurement also shows the degree of maturity of the produced document as well to evaluate the understanding of the developer before proceeding to the next phase in software development.

6.4 Limitation

This section discusses the limitation that leads to future work. There are numerous limitations in this section that can be discussed and considered as future work:

i. **Number of qualities being assessed.** The number of qualities being assessed is not sufficient to justify the quality of SRS. The study only focusses on four number of qualities: completeness, correctness, consistency and preciseness. There are few other qualities that should be taken into consideration when assessing the SRS document: reliability, readability, manageability and so on.

ii. **Number of quality properties.** This study proposed few properties for each of the quality being assess. Only general properties are proposed as the study only concerns on the heterogeneous domain. A variety of domains has its own target quality properties to be assessed. The quality properties to be assessed

in this study are not enough to represent those qualities as each domain has its own properties to be stressed on. This can be seen in the transportation domain in which safety is the main concern.

iii. **Number of SRS element being assessed.** There are numerous elements in the SRS document: topic in the ToC, document formatting and so on. The study only focused on the structural of the document as well as the FR topic. The other elements should be studied in order to optimize the SRS document quality.

iv. **Reliability of knowledge-based.** The knowledge-based needs to be updated periodically to ensure the reliability of the outcome result. The <*admin*> is responsible to ensure the library is updated accordingly. The prototype provides a platform where the <*admin*> can manage knowledge-based easily.

v. **Prototype understanding 'role' meaning.** The prototype is designed where the assessment of FR role is based on the similarity of word and character used. To overcome this issue, the human interference which was <*user*> is needed to confirm the resulted degree of similarity that was produced by the prototype.

6.5 Future Work

This study is focusing on measuring the degree of quality of SRS. However, there are some features that can be added for future enhancement:

i. **Considering the other quality.** The other quality such usability, reliability, traceability, testability and so on. A larger number of qualities assessed will impact the degree of the measurement.

ii. **Add user story feature.** Allow the user's involvement in the prototype. Adopt Likert Scale analysis to increase user acceptance toward FR.

6.6 Conclusion

The objectives of the study had been achieved. This study is targeted to assess the quality of SRS in a heterogeneous domain. Only two elements of the SRS document are chosen which are structural and FR. Four qualities are chosen to assess the SRS which is completeness, consistency, correctness and preciseness. The properties for each quality had been chosen based on its capability to generalize on each domain. A framework had been proposed to better visualize the idea. Those properties are then formalized by using Likert Scale analysis. The range of the Likert Scale analysis is between one and zero. One represents successfully found and zero is otherwise. The rule and equation for each quality are also proposed in Chapter 3. The rule and equation are proposed in order to convert the qualitative analysis to quantitative measurement.

A prototype is built based on the proposed rule and equation to help in evaluating the provided SRS. The proposed flow of the prototype can be seen in the activity diagram located in Chapter 4. The evaluation result shows improvement in terms of evaluating the structural and FR in the SRS document. Studies on the collection of SRS document shows lack of similarity topic in between of the IEEE 830 ToC and assessed SRS document. Furthermore, there is no standardization of how the FR is defined.

The adoption of the IEEE 830 standard as a minimum standard must be followed as it increases the readability and reliability. Studies show that other researchers used a translation matrix in order to compare their SRS document structure with IEEE 830 ToC. Significant improvement can be seen where the prototype automates the process of comparison without any human interference. This prototype reduces the time needed to evaluate the SRS document structure.

Further studies done on the collection of the SRS document shows a lack of standardization used in order to define the FR. RB is adopted in the prototype in order to refine the original text written in the inputted document. RB is a semi-formal template which restricts the usage of natural language. Assessment of consistency, correctness and preciseness quality improves the readability and traceability. The term that is used in defining the role of each FR and *'Normal Flow'* element of test case must be specified as this SRS document also will be a baseline for other team developers to develop the system. The developer which was the <*user*> of the prototype also has the advantage to evaluate their FR by using use case testing for validation and similarity check to check for redundancy.

This thesis has provided a deeper insight into the assessment quality of the SRS document. This work contributes to the element of the SRS document which was the structural and FR. Despite the number of qualities to evaluate the SRS document is not enough, this study represents a groundwork for future researchers. More qualities or properties need to be figured out in order to improve the evaluation result. Another important implication that needs to be reconsidered is the usage of the IEEE 830 standard as minimal standard and must be followed in order to structure the SRS document. The implication is

the level of readability will increase when the SRS document is passed on from one developer to another. Aside from that, the adoption of semi-formal template RB provides the uniformity for defining FR.

REFERENCES

Abuqaddom, I., Alazzam, H., Hudaib, A., & Al-Zaqhoul, F. (2019). A measurable website usability model: Case Study University of Jordan. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 2019 10th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 11-13 June 2019, IEEE.

Ahmad, A., Anuar, U., & Emran, N. A. (2018). A Tool-based Boilerplate Technique to Improve SRS Quality: An Evaluation. *Journal of Telecommunication, Electronic and Computer Engineering, 10*(2 – 7), 111 – 114.

Ali, M. J. (2006). *Metrics for requirements engineering*. [pdf] Available at: http://www8.cs.umu.se/education/examina/Rapporter/JaveedAli.pdf [Assessed on 9 October 2018]

Ali, S. W., Ahmed, Q. A., & Shafi, I. (2018). Process to enhance the quality of software requirement specification document. In: Institute of Electrical and Electronics Engineers, *Proceedings of the International Conference on Engineering and Emerging Technologies*, Lahore, Pakistan, 22-23 February 2018, IEEE.

Anil, G. R., & Moiz, S. A. (2017). A holistic rubric for assessment of software requirements specification. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 5th National Conference on E-Learning & E-Learning Technologies*, Hyderabad, India, 3-4 August 2017, IEEE.

Ansari, A., Fatima, A. S., Shagufta, M. B., & Tehreem, S. (2017). Constructing Test Cases Using Natural Language Processing. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 2017 Third International Conference on Advances in*

Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, India, 27-28 February 2017, IEEE.

Anuar, U., Ahmad, S., & Emran, N, A. (2015). A simplified systematic literature review: Improving software requirements specification quality with Boilerplates. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 9th Malaysian Software Engineering Conference*, Kuala Lumpur, Malaysia, 16-17 December 2015, IEEE.

Asif, M., Ali, I., Malik, M. S. A., Chaudary M. H., Tayyaba, S., & Mahmood, M. T. (2019). Annotation of Software Requirements Specification (SRS), Extractions of Nonfunctional Requirements, and Measurement of Their Tradeoff. *IEEE Access, 7*, 36164 – 36176.

Audytra, H., Hendradjaya, B., & Sunindyo, W. D. (2016). A Proposal for quality assessment model for software requirements specification in Indonesian language for e-Government. In: Institute of Electrical and Electronics Engineers, *Proceedings of the International Conference on Data and Software Engineering*, Denpasar, Indonesia, 26-27 October 2016, IEEE.

Aurum, A., & Wohlin, C. (2005). *Engineering and managing software requirements*. Verlag Berlin Heidelberg: Springer.

Bano, M. (2015). Addressing the challenges of requirements ambiguity: A review of empirical literature. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, Ottawa, Canada, 24-24 August 2015, IEEE.

Besrour, S., Rahim, L. B. AB., & Dominic, P. D. D. (2016). A Quantitative Study to Identify Critical Requirement Engineering Challenges in the Context of Small and Medium Software Enterprise. In: Institute of Electrical and Electronics Engineers, *Proceedings*

of the 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 15-17 August 2016, IEEE.

Bhatia, M. P. S., Kumar, A., & Beniwal, R. (2016). Ontology based framework for detecting ambiguities in software requirement specification. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 3rd International Conference on Computing for Sustainable Global Development*, New Delhi, India, 16-18 March 2016, IEEE.

Christiane. F. (2005). *About WordNet*. [online] Available at: https://wordnet.princeton.edu/ [Assessed on 9 October 2018]

D'Antiga, C. (2013). *ITS Adriatic Multi-port gateway*. [pdf] Available at: https://ec.europa.eu/inea/en/ten-t/ten-t-projects/projects-by-country/multi-country/2010-eu-21106-s [Accessed on 22 December 2015]

Data, R. (1998). *PHP similar_text() Function*. [online] Available at: https://www.w3schools.com/php/func_string_similar_text.asp [Assessed on 9 October 2018]

Dennis, A., Wixom, B. H., & Roth, R. M. (2013). *System analysis and design*, 5th ed., Hoboken, NJ: Wiley.

Essatouti, B., Khamar, H., Fkihi, S. E., Faizi, R., & Thami, R. O. (2018). Arabic sentiment analysis using a Levenshtein Distance based representation approach. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 5th International Congress on Information Science and Technology*, Marrakech, Morocco, 21-27 October 2018, IEEE.

Farbey, B. (1990). Software quality metrics: consideration about requirements and requirement specifications. *Information and Software Technology, 32*(1), 60-64.

Fernández, D. M., & Wagner, S. (2015). Naming the pain in requirements engineering: A design for a global family of surveys and first results from Germany. *Information and*

Software Engineering, 57, 616-643.

Ferreira, B., Silva, W., Barbosa, S. D., & Conte, T. (2018). Technique for representing requirements using personas: a controlled experiment. *IET Software, 12*(3), 280–290.

Galin, D. (2018). *Software Quality: Concepts and Practice: Concepts and Practice*, Hoboken, USA: Wiley

Georgiades, M. G., & Andreou, A. S. (2010). Automatic generation of a software requirements specification (SRS) document. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications*, Cairo, Egypt, 29 November-1 December 2010, IEEE.

Ghazi, P., & Glinz, M. (2018). FlexiView Experimental Tool: Fair and Detailed Usability Tests for Requirements Modeling Tools. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 2018 IEEE 26th International Requirements Engineering Conference (RE)*, Banff, Canada, 20-24 August 2018, IEEE.

Glinz, M., Paech. B., & Rogers. G. (2017). *International Requirements Engineering Board (IREB) - Foundation Level*. [pdf] Available at: https://ireb.ita-stqb.org/docs/ireb_cpre_syllabus_fl_en.pdf [Assessed on 9 October 2018]

Graham, D., Veenendaal, E. V., Evans, I., & Black, R. (2006). *Foundations of software testing ISTQB certification*. Cengage Learning Emea, United Kingdom: Thomson.

Harmon, S. Y., & Youngblood, S. (2008). Lessons learned about developing acceptable acceptability criteria. In: Society for Modeling and Simulation International, *Proceedings of the 2008 Summer Computer Simulation Conference*, Edinburgh, Scotland, June 2008, ACM.

Hovorushchenko, T., & Pomorova, O. (2016). Evaluation of mutual influences of software quality characteristics based ISO 25010:2011. In: Institute of Electrical and Electronics

Engineers, *Proceedings of the 11th International Scientific and Technical Conference Computer Sciences and Information Technologies*, Lviv, Ukraine, 6-10 September 2016, IEEE.

Hull, E., Jackson, K. & Dick, J. (2011). *Requirements engineering*, 3rd ed., New York: Springer.

Ibrahim, N., Kadir, W. M. N. W., & Deris, S. (2014). Documenting requirements specifications using natural language requirements Boilerplates. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 8th Malaysian Software Engineering Conference*, Langkawi, Malaysia, 23-24 September 2014, IEEE.

Institute of Electrical and Electronics Engineers Standards Association. (2010). ISO/IEC/IEEE Standard No. 24765 - *Systems and software engineering – Vocabulary*. Switzerland: IEEE.

Institute of Electrical and Electronics Engineers Standards Association. (1998). ISO/IEC/IEEE Standard No. 1233 - *IEEE guide for developing system requirements specifications*. USA: IEEE.

Institute of Electrical and Electronics Engineers Standards Association. (1998). ISO/IEC/IEEE Standard No. 830 - *IEEE recommended practice for software requirements specifications*. USA: IEEE.

Kamalrudin, M., & Sidek, S. (2015). A review on software requirements validation and consistency management. *International Journal of Software Engineering and Its Application, 9*(10), 39-58.

Kocerka, J., Krześlak, M., & Gałuszka. A. (2018). Analysing quality of textual requirements using Natural Language Processing: A Literature Review. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 2018 23rd International Conference on*

Methods & Models in Automation & Robotics (MMAR), Miedzyzdroje, Poland, 27-30 August 2018, IEEE.

Kuchta, J., & Padhiyar, P. (2018). Extracting concepts from the software requirements specification using natural language processing. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 2018 11th International Conference on Human System Interaction (HSI)*, Gdansk, Poland, 4-6 July 2018, IEEE.

Laplante, P. A. (2007). *What every engineer should know about software engineering*, Boca Raton, Florida: CRC Press.

Maguire, M., & Bevan, N. (2002). User requirements analysis: A review of supporting methods. In: Ricardo, A. B. Y., Ugo, M., and Santoro, N., *Proceedings of the IFIP 17th World Computer Congress*, Boston, Massachusetts, August 2002, Semantic Scholar.

McCall, J. A., Richards, P. K. & Walters, G. F. (1977). *Factors in Software Quality - Volume 1 - Concept and Definitions of Software Quality*. [pdf] Available at: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA049014 [Accessed on 22 December 2015]

Medeiros, J., Goulão, M., Vasconcelos, A., & Silva, A. (2016). Towards a model about quality of software requirements specification in agile projects. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 10th International Conference on the Quality of Information and Communications Technology*, Lisbon, Portugal, 6-9 September 2016, IEEE.

Miles, R., & Hamilton, K. (2006). *Learning UML 2.0*, Sebastop, CA: OReilly.

Mund J., Femmer H., Fernandez D. M. & Eckhardt J. (2015). Does quality of requirement specifications matter? Combined results of two empirical studies. In: Institute of Electrical and Electronics Engineers, *Proceedings of the International Symposium on*

Empirical Software Engineering and Measurement, Beijing, China, 22-23 October 2015, IEEE.

Nazir, F., Butt, W. H., Anwar, M. W., & Khattak, M. A. K. (2017). The Applications of Natural Language Processing (NLP) for Software Requirement Engineering - A Systematic Literature Review. In: Kim, K. and Joukov N., *Proceedings of the ICISA: International Conference on Information Science and Applications*, Macau, China, 20-23 March 2017, Springer.

Omoronyia, I., & Stålhane, T. (2016). *Guided Natural Language and Requirement Boilerplate*. [PowerPoint Slides] Available at: www.idi.ntnu.no/emner/tdt4242/foiler/2-3-Boilerplates.ppt [Assessed on 9 October 2018]

Osman, M. H., & Zaharin, M. F. (2018). Ambiguous software requirement specification detection: an automated approach. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 5th International Workshop on Requirements Engineering and Testing*, Gothenburg, Sweden, 2-2 June 2018, IEEE.

Pohl, K., & Rupp, C. (2015). R*equirement engineering fundamentals*, 2nd ed., Santa Barbara, United States: Rocky Nook.

Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioner's approach*, 8th ed., Penn Plaza, New York: Mc Graw Hill Education.

Riaz, M. Q., & Butt, W. H. (2019). Automatic Detection of Ambiguous Software Requirements: An Insight. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 2019 5th International Conference on Information Management (ICIM)*, Cambridge, United Kingdom, 24-27 March 2019, IEEE.

Sabriye, A. O. J. & Zainon, W. M. N. W. (2017). A framework for detecting ambiguity in

software requirement specification. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 8th International Conference on Information Technology*, Amman, Jordan, 17-18 May 2017, IEEE.

Saito, S., Takeuchi, M., Yamada, S., & Aoyama, M. (2014). RISDM: A requirements inspection systems design methodology: Perspective-based design of the pragmatic quality model and question set to SRS. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 22nd International Requirements Engineering Conference*, Karlskrona, Sweden, 25-29 August 2014, IEEE.

Samah, M. A. (2011). *Chapter 4: Software Testing. Training presented for Certified Tester*, Foundation Level (CTFL) in Univeristi Malaysia Sarawak (UNIMAS), Kuching. Sarawak.

Sarmiento, E., Leite, J. C. S. d. P., & Almentero, E. (2015). Using correctness, consistency, and completeness patterns for automated scenarios verification. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 5th International Workshop on Requirements Patterns*, Ottawa, Canada, 25-25 August 2015, IEEE.

Shah, A., Alasow, M. A., Sajjad, F., & Baig, J. J. A. (2017). An Evaluation of Software Requirements Tools. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 8th IEEE International Conference on Intelligent Computing and Information Systems*, Cairo, Egypt, 5-7 December 2017, IEEE.

Silke (2012). *Vague Words List*. [online] Available at: http://evilauthor.com/blog/downloads/6 [Assessed on 9 October 2018]

Singh, B., & Kannojia, S. P. (2013). A review on software quality models. In: Institute of Electrical and Electronics Engineers, *Proceedings of the International Conference on Communication Systems and Network Technologies,* Gwalior, India, 6-8 April 2013,

IEEE.

Slhoub, K., Carvalho, M., & Bond, W. (2017). Recommended practices for the specification of multi-agent systems requirements. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference*, New York, United States of America, 19-21 October 2017, IEEE.

Sommerville, I. (2016). *Software engineering*, 10th ed., Kirby Street, London: Pearson Education Limited.

Standish Group, (1994). *The Chaos Report*. [pdf] Available at: https://www.standishgroup.com/chaosReport/index?archive=0#myModal_1 [Accessed on 9 October 2018]

Standish Group, (2010). *Chaos Manifesto The Laws of CHAOS and the CHAOS 100 Best PM Practices*. [pdf] Available at: https://www.standishgroup.com/chaosReport/index#myModal_13 [Accessed on 9 October 2018]

Su, J., Han, L., Li, Y., Liu, S., Yao, Z., & Bao, T. (2016). Research on building and analysis for attribute model in quality evaluation of domain software. In: Zu, Q. and Hu, B. *Proceedings of the International Conference on Human Centered Computing*, Colombo, Sri Lanka, 7-9 January 2016, Springer.

Takoshima, A., & Aoyama, M. (2015). Assessing the quality of software requirements specifications for automotive software systems. In: Institute of Electrical and Electronics Engineers, *Proceedings of the Asia-Pacific Software Engineering Conference*, New Delhi, India, 1-4 December 2015, IEEE.

Temiz, U., Altay, F., Mehteroğlu, C., Dülgeroğlu, M. S., & Aktaş, M. H. (2014). *Carpool*

System. [pdf] Available at: https://senior.ceng.metu.edu.tr/2014/such/documents/SRS.pdf [Assessed on 8 November 2010]

Thitisathienkul, P., & Prompoon, N. (2015). Quality assessment method for software requirements specifications based on document characteristics and its structure. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 2nd International Conference on Trustworthy Systems and their Applications*, Hualien, Taiwan, 8-9 July 2015, IEEE.

Wiegers, K., & Beatty, J. (2013). *Software requirement*, 3rd ed., Redmond, Washington: Microsoft Press.

Zafar, I., Shaheen, A., Nazir, A. K., Maqbool, B., Butt, W. H., & Zeb, J. (2018). Why Pakistani Software Companies don't use Best Practices for Requirement Engineering Processes. In: Institute of Electrical and Electronics Engineers, *Proceedings of the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, Canada, 1-3 November 2018, IEEE.

Zhang, M., Yue, T., Ali, S., Selic, B., Okariz, O., Norgre, R., & Intxausti, K. (2018). Specifying Uncertainty in Use Case Models. *The Journal of Systems & Software, 144*, 573 – 603.

APPENDICES

Appendix A: List of SRS document based on domain

Domain: Account/Finance
Business Accounting and Payroll System
Central Accounts Management System
Road Usage Charge Implementation
Domain: Administration
Carpool System
Coordinated Highways Action Response Team
Reversible Lane Control System
Domain: Agriculture
Marketing Infrastructure for Agriculture, Horticulture, Floriculture, Livestock and Fisheries Culture
STEWARDS (Sustaining the Earth's Watersheds)
WMITS (Waste Management)
Domain: Analysis Measurement
CMCF (Canadian Macromolecular Crystallography Facility) Beamline Operating System
E-LEDA: E-Learning Data Analysis
Open Source Sustainability Assessment Framework, Format Converter Module
Optics Lab Measurement System
Social Networks Visualizer
SPINGRID
WAVED (Web App for Visualizing Environmental Data)
Campbell Prediction System

MEMAC - Memory Modules All Chip Analyser Statistical Analysis Tool
Domain: Communication EIRENE (European Integrated Railway Radio Enhanced Network) Applying Broadcasting / Multicasting / Secured Communication to agentMom in Multi-Agent Systems Folk - An Online Community Library Blog
Domain: Education Educational Game System E-Learning Tool Online Examination System
Domain: Game / Entertainment Wings Lego Robot Soccer Project Gerbil
Domain: Geolocation Amazing Lunch Indicator i-Android The Social Travel App
Domain: Management College Library Management System Design and Development of CSC based Multi-Utility System Including Access Control and Attendance Monitoring CSIR Infrastructure Works and Services Project DropIt

EMMON - EMbedded MONitoring
E-Platform
Monitoring and Management of ESI Scheme (ESIC-Employee's State Insurance Corporation)
EVLA Software System
Libra: An Economy-Driven Cluster Scheduler
Project Management System
Schedule and TimeLine Java Bean Components Tide project
Time Monitoring Tool
Volere Online registration system
Comprehensive Watershed Management Water Use Tracking Project
Bus Tracking System
Buyer Purchase Management System (BPMS)
Graduation Outliner
Content Management Service
Student and Unit Management System (SUMS) - Registration Module
Domain: Marketing
"Flea Market" System
Airline Flight Booking System
BooksUnlimited: Fast Track Shopping Software System
Global Personal Marketplace
I.D.S. International Drug Store Online
SkyNet Online Shopping Mall
Taylor's Professional Services
Airline Reservation System
Domain: Medical

Digital Hospital and Medical Information System
Domain: Mobile Apps Scavenger Hunt SplitPay FACETs Mobile Mobile Client Query Application (MCQA) SyncML project
Domain: Multimodal Multi-touch Newspaper Scriber Thraxion Electronic Scrolling Displays
Domain: Network AIM Template and AIM Template Manager Home Subscriber Server Port Community System (PCS) Protogate Freeway® Software Offline Charging System
Domain: Program Tools JHotDraw NBDiff Documentations Ostories Matter PDF Split and Merge PeaZip
Domain: Safety / Security

KeePass Password Safe Safehome Project Surveillance Dashboard Web Services Security Federal Manager 7.5 BaseBlockSystem
Domain: Society Connecticut Electronic Poll Book System Distributed Polling System E-Elections Software "The Pericles Project" E-Voting
Domain: Software Development DEVCLOUD Web Based Integrated Development Environment
Domain: Traffic / Transportation RROWDyS Project Development STC UTC System Centralized Traffic Control PARKme System

Appendix B: FR extracted from Carpool System (Temiz U. et al., 2014)

Sign Up: Users need to sign up to use the web site. The users should have a username and password. After filling their name, surname, e-mail, age, job, phone and gender information, they register the system.

Sign In: If a user is signed up, s/he can sign in the system by filling username and password boxes.

Sign Out: A user may need to sign out the system. S/he can do it by clicking the sign out button which is placed in every page.

Add Transportation Route: Users may add transportations by specifying a route, time/time period and number of empty seats. The user can select the route by two different way. The first way is entering start and end locations. Thus, the route is drawn on the map. The other way is selecting start and end locations on the map. Also, he/she can select at most 8 waypoints.

Delete Transportation Route: A user may delete his/her transportation route. After deleting route, other passengers in that transportation will be informed by the system.

Request Transportation Route: A user may use a transportation by sending transportation request to the driver of the transportation.

Search Transportation Route: A user can search for transportations that the user can see suitable routes to his/her route by specifying time and route.

Send Message: The users can have communicate each other by sending message.

Reply To A Message: After receiving a message, the user can read the message.

Block User: When a user receive disturbing message, s/he can block the user who send that message.

Rate User: After having a transportation, the users in the same transportation can rate each other on the web site. Thus, other users can see the user rates and they can decide which transportation is better.

Change Language: The user can change the web site language by clicking small flags that is available in all pages. There are two suitable languages which are Turkish and English.

Appendix C: FR extracted from ITS Adriatic Multi-port gateway (D'Antiga, 2013)

The platform administrator inserts a communication protocol that can be used by Client Systems for connecting to the platform.
The platform administrator modifies a communication protocol already catalogued in the system.
The platform administrator deletes an already inserted communication protocol.
The platform Administrator inserts information about a data format that can be used for formatting the documents exchanged by Client Systems through the platform.
Platform administrator modifies a data format already catalogued.
The platform administrator deletes an already inserted data format.
The platform administrator inserts the information about the ability of each type of document to be transmitted to the receiver (Client System) and/or stored in the EDI-platform for being consulted by common users.
The platform administrator modifies a document type already catalogued in the system.
The platform administrator deletes an already inserted document type.
Platform Administrator inserts a transmission modality.
Platform administrator modifies a transmission modalities already catalogued.
The platform administrator deletes an already inserted data format.

Platform administrator registers a Client System into the platform and creates the user credentials for accessing the system configured services for document exchanging.

Platform administrator modifies a Client System already catalogued.

The platform administrator deletes an already inserted Client System.

The client System administrator subscribes the Client System to the platform making a choice of the type of documents that are transmitted and received and specifying for each one the transmission modality.

Client System Administrator modifies a Client System already catalogued.

The Client System administrator deletes an already inserted Client System.

The platform administrator accesses a platform console for monitoring the document flow that passes through the platform.

The Client System dispatches the document to the platform according to the protocol and the format configured for the document type in the Client System profiling.

The System make a service available to the Client System in order to have a list of the downloadable documents.