

Islamic University of Gaza
Deanery of Higher Studies
Faculty of Information Technology
Information Technology Program



An Ontology-Based Approach to Support the Process of Judging Hadith Isnad

Prepared By:
Yehya M. Dalloul

Supervised By:
Dr. Rebhi S. Baraka

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master in Information Technology

March 2013

إقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

An Ontology-Based Approach to Support the Process of Judging Hadith Isnad

أقر بأن ما اشتملت عليه هذه الرسالة إنما هي نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل، أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أية مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification

Student's name:

اسم الطالب: يحيى محمد مصطفى دلول

Signature:

التوقيع:

Date:

2013/2/17

التاريخ:

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



الجامعة الإسلامية - غزة
The Islamic University - Gaza

هاتف داخلي: 1150

عمادة الدراسات العليا

ج س ع/35/
الرقم Ref
2013/02/10
التاريخ Date.....

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة عمادة الدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ يحيى محمد مصطفى دلول لنيل درجة الماجستير في كلية تكنولوجيا المعلومات برنامج تكنولوجيا المعلومات وموضوعها:

An Ontology-Based Approach to Support the Process of Judging Hadith Isnad

وبعد المناقشة التي تمت اليوم الأحد 29 ربيع أول 1434هـ، الموافق 2013/02/10م الساعة الحادية عشرة ظهرًا بمبنى القدس، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

	مشرفاً ورئيساً	د. رحي بركة
	مناقشاً داخلياً	د. إياد الأغا
	مناقشاً خارجياً	د. مصطفى جرار

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية تكنولوجيا المعلومات / برنامج تكنولوجيا المعلومات.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر عظمه في خدمة دينه ووطنه.

والله والتوفيق،،،

عميد الدراسات العليا

أ.د. فؤاد علي العاجز

Acknowledgment

First, I thank Allah for giving me the strength and ability to complete this study.

I also thank my family: mother, father, brothers, and sisters; I also thank my wife for support and encourage working on this thesis.

I also thank Mr. Nawaf AlBadia, Dr. Aqil Azmi, Osama Qunoo, Wael Al Shurafa, and Abd El Rahman Abu Sahnab for their valuable technical help.

I also thank Mr. Ibrahim Al Kurd for his role in manually judging the sample Hadiths that used for testing and evaluating.

Thanks must also go to my colleagues at the college of information technology for the many inspiring and fruitful discussions and suggestions.

Finally, I should extend special thanks to my supervisor Dr. Rebhi Baraka, without his help, guidance, and continuous follow up; this research would never have been.

*Yehya M. Dalloul
March 2013*

Abstract

The two fundamental sources of Islamic legislation are Qur'an and Hadith. Hadiths, or prophetic traditions, are narrations originating from the sayings and actions of Prophet Muhammad (peace be upon him). The narrators transmitted the sayings of the Prophet by Isnad and for the importance of Isnad Muslims keen interest of Isnad science because It helps differentiate between the accepted and rejected Hadith, or, in other words, Sahih and weak Hadith. The Islamic scholars were the first to study Isnad accurately to determine who the trusted men are and vice versa. We want to develop this science through this research. We build an ontology-based Isnad Judgment System (IJS) that automatically generates a suggested judgment of Hadith Isnad. It based on the rules that Hadith scholars follow to produce a suggested judgment. A prototype of the approach implemented to provide a proof of concept for the requirements and to verify its accuracy. We evaluated the system according to Al-Albani scholar and according to Hadith specialist. The Results discussed in both approaches where the accuracy of the system in the first approach is 75% in the second approach is 81%. These results prove that the ontology supports the process of Isnad judgment. We evaluated the ontology using Task-Based framework it indicate that the accuracy of using the IJS ontology is 100%.

Keywords: Hadith, Hadith Narration Chains, Hadith Isnad, Hadith Ontology, Hadith Judgment, Isnad Judgment.

عنوان البحث

إستخدام الأنتولوجي في دعم عملية الحكم على سند الحديث

الملخص

يعتبر القرآن الكريم والحديث النبوي الشريف من مصادر التشريع الإسلامي الأساسية. والأحاديث هي الروايات التي منشؤها من أقوال وأفعال النبي محمد (صلي الله عليه وسلم). وقد نقل المحدثون هذه الأحاديث بالأسانيد، ولأهمية الأسانيد اهتم المسلمون اهتماما شديدا بعلم الإسناد لأن عليه الإعتقاد في تميز الاحاديث وبيان المقبول منها والمردود، والصحيح والسقيم، وكان لعلماء الحديث المسلمين سبق في هذا العلم فلا يوجد ممن سبق من اعتنى بدراسة رجال إسناد روايات الأنبياء بهذا الإتقان والضبط للرجال ومعرفة الثقة والضعيف المدلس وغيره. ومساهمة منا في تطوير هذا العلم قمنا ببناء نظام يقوم بالحكم على سند الحديث ويعتمد على الأنتولوجي (Ontology)، حيث يقوم باعطاء حكم مقترح على سند الحديث بشكل الي ومبني على القواعد التي يتبعها علماء الحديث في الحكم على سند الحديث. وقد تم تطوير نموذج أولي لتقديم دليل على نجاح الطرح وقد تم التحقق من دقته من خلال تقييمه مقارنة بأحكام الألباني وأحكام متخصص في تخريج الاحاديث. كانت النتائج في كلا النهجين من حيث دقة النظام في النهج الأول هو 75% وفي النهج الثاني هو 81%. هذه النتائج أثبتت أن الأنتولوجيا تدعم عملية الحكم على سند الحديث. قمنا أيضا بتقييم الأنتولوجي باستخدام إطار Task-Based وكانت دقة الأنتولوجي هي 100% مما يدل على صحة بناء الأنتولوجي.

كلمات البحث: الحديث، سلسلة رواة الحديث، الإسناد، علم مطلح الحديث، انتولوجيا

الحديث، الحكم على الإسناد.

Table of Contents

Acknowledgment	I
Abstract	II
المخلص	III
Table of Contents	IV
List of Figures	VII
List of Tables	IX
List of Appendices	X
List of Terminology	XI
Introduction	1
1.1 Hadith Science and Ontology Development.....	1
1.2 Statement of the Problem	2
1.3 Objectives	3
1.3.1 Main Objective.....	3
1.3.2 Specific Objectives	3
1.4 Importance of the Research	3
1.5 Scope and Limitations of the Research.....	4
1.6 Methodology	4
1.7 Resources, Methods, and Tools.....	5
1.8 Thesis Structure	6
1.9 Summary	7
State of the Art and Review of Related Works	8
2.1 Hadith Electronic Encyclopedias.....	8
2.2 Hadith Science Researches	8
2.3 Arabic and Islamic Ontology Development	10
2.4 Ontology and Decision Support	11
2.5 Summary	12
Theoretical and Technical Foundations	14
3.1 Hadith Science and Isnad Judgment.....	14

3.1.1	The Importance of Isnad	14
3.1.2	The Importance of the Science of Judgment on Isnad	15
3.1.3	Judgment of Isnad VS Judgment of Hadith	15
3.1.4	Steps of Studying Isnad	16
3.1.5	Judging on Isnad.....	17
3.2	Ontology and the Semantic Web	18
3.2.1	Introduction	18
3.2.2	Ontology Development	20
3.2.3	Ontology Evaluation	22
3.2.4	Ontology Tools.....	23
3.3	Summary	25
	IJS Ontology Development.....	26
4.1	IJS Ontology Development.....	26
4.2	Evaluating the IJS Ontology	33
4.2.1	Quality through Hadith Number Six Example	33
4.2.2	Quality through Ontology Querying	36
4.3	Summary	38
	Algorithm for Isnad Judgment.....	39
5.1	Separating the Isnad from Hadith (Matn)	39
5.1.1	e-Narrator Overview	39
5.1.2	Creating the e-Narrator Web Service	40
5.1.3	Testing e-Narrator Web Service.....	42
5.2	Identifying the Narrator Using the IJS Ontology	42
5.2.1	Narrator Identification Rules	44
5.2.2	Hadith Number Six Example (continued).....	44
5.3	Check Continuity of Isnad	48
5.3.1	Hadith Number Six Example (continued).....	49
5.4	Applying the Judgment Rules	49
5.4.1	Hadith Isnad Judgment Algorithm (HIJ)	49
5.4.2	Hadith Number Six Example (continued).....	53
5.5	Summary	54

IJS Prototype	55
6.1 System Analysis.....	55
6.1.1 Overall Description:	55
6.1.2 System Functions.....	55
6.1.3 Specific Requirements.....	56
6.1.4 Hadith Isnad Judgment Process	60
6.2 System Design	61
6.2.1 System Architecture	61
6.2.2 IJS Components.....	62
6.2.3 IJS Diagrams	64
6.3 System Implementation.....	65
6.3.1 User Web Interface	66
6.3.2 Hadith Database	67
6.3.3 e-Narrator Web Service	67
6.3.4 IJS Ontology	68
6.3.5 Judgment Rules	68
6.3.6 Judgment System.....	69
6.4 System Testing	70
6.5 Summary	70
Results and Evaluation	72
7.1 IJS Prototype Evaluation.....	72
7.1.1 Accuracy According to Al-Albani Scholar	72
7.1.2 Accuracy According to Hadith Specialist.....	74
7.2 IJS Ontology Evaluation	75
7.3 Summary	76
Conclusion and Future Work.....	77
Bibliography.....	78
Arabic Bibliography	79

List of Figures

Figure 3.1: Berners-Lee Semantic Web Layers [16]	19
Figure 3.2 Ontology Development Processes [26].....	20
Figure 4.1: HadithRDF Ontology [29].....	27
Figure 4.2: IJS Ontology Main Classes in OWL	29
Figure 4.3: IJS Ontology Individual	32
Figure 4.4: Ontology of Hadith Number 6.....	34
Figure 4.5 Ontology of Hadith Number 6 in Protégée.....	34
Figure 4.5: Query for Teachers and Student of Narrator.....	36
Figure 4.6: Query the Narrator by Konia.....	37
Figure 4.7: Query the Narrators of Hadith Number 6.	37
Figure 5.1: Abstract Level of the e-Narrator System Components [29]	40
Figure 5.2: WSDL of e-Narrator Web Service	41
Figure 5.3: HIJ Algorithm Flow Chart.....	50
Figure 6.1: Use Cases of Researcher	57
Figure 6.2: Use Cases of Administrator	58
Figure 6.3: Hadith Isnad Judgment Process	60
Figure 6.4: System Architecture	61
Figure 6.5: Searching and Exploring Hadiths Interface	62
Figure 6.6: Output Interface	62
Figure 6.7: Hadith Books Database ER-Diagram	63
Figure 6.8: Deployment Diagram of IJS	64
Figure 6.9: Get the Judgment of Hadith.	65
Figure 6.10 Searches and Exploring Interface.....	66
Figure 6.11: Results Interface	67
Figure 6.12: Hadith Input to e-Narrator Web Service	68
Figure 6.13: Narrator Chain as Output of e-Narrator Web Service	68
Figure 6.14: Part of the Implemented Algorithm.....	69
Figure 7.1: Results of IJS Compared to Al Albani.....	74
Figure 7.2: Results of IJS Judgment Compared to Hadith Specialist Judgment	75
Figure A.1: Class Diagram of Book	81
Figure A.2: Class Diagram of Narrator	81
Figure A.3: Display the Hadith by Exploring.....	82
Figure A.4: Display the Hadith by Search.	82
Figure A.5: Get the Judgment of Hadith.	83
Figure A.6: Add New Narrator Information.....	83
Figure A.7: Update Narrator Information.	83
Figure A.8: Add New Book Information.....	84
Figure A.9: Update Book Information	84
Figure A.10: Add New Hadith Information.	84
Figure A.11: Update Hadith Information	85
Listing B.1: Hadith Chains Judge.....	86
Listing B.2: Hadith Handler	87
Listing B.3: Hadith Web Service Handler	88

Listing B.4: Narrator Full Info Handler	88
Listing B.5: Paths Judgment.....	91
Figure C.1: OWL Source Code of IJS Ontology	101
Figure E.1: Code of JUnit Test for User Web Interface.....	106
Figure E.2: JUnit Test Results Window of User Web Interface.....	106
Figure E.3: Code of JUnit Test for e-Narrator Web Service Function	107
Figure E.4: JUnit Test Results Window of e-Narrator Web Service.....	107
Figure E.5: Code of JUnit Test for Isnad Judgment System Function	108
Figure E.6: JUnit Test Results Window of Isnad Judgment System.....	108

List of Tables

Table 3.1: Isnad Judgment Rules	18
Table 4.1: IJS Ontology Classes	29
Table 4.2: Hadith Ontology Object Properties	30
Table 4.3: Hadith Ontology Data Properties	31
Table 4.4: Narrator of Hadith Number Six	34
Table 5.1: Input and Output of the Web Service	42
Table 5.2: Name Manners of Narrators	43
Table 5.3: Result of Query about First Narrator of Hadith.....	45
Table 5.4: Result of Query about Second Narrator of Hadith.....	46
Table 5.5: Result of Query about Third Narrator of Hadith	46
Table 5.6: Result of Query about Fourth Narrator of Hadith.....	47
Table 5.7: Result of Query about Fifth Narrator of Hadith	48
Table 5.8: Results of Ontology Reasoning	49
Table 5.9: Results of Applying HIJ Algorithm.....	53
Table 6.1: System Use Cases.....	56
Table 7.1: Accuracy According to Al albani.....	73
Table 7.2: Accuracy According to Isnad Judgment by Hadith Scholar	74
Table 7.3: Results Overview	76

List of Appendices

Appendices	81
Appendix A: IJS Analysis and Design Components	81
Appendix B: IJS Implementation	86
Appendix C: OWL Source Code	92
Appendix D: Hadith Samples	102
Appendix E: System Testing	106

List of Terminology

Bida'a	Method invented in matters of religion, by which people meant to extremes in worshipping God.
Chain of Isnad	Chain of narrators who were transferred Hadith from the first source, which is the Messenger of Allah.
Continuity of Isnad	Means that each narrator of it narrators take the Hadith from his teacher from the first Isnad to its limit.
Hadith	Is oral traditions related to the words and deeds of Prophet Muhammad (peace be upon him).
Hasan Hadith	Linguistically means good. Ibn Hajar defines Hasan “Hasan in and of itself”.
Ikhtelat	Narrator oblivion old either age or go blind or the burning his books if that adopted.
Isnad	Chain of narrators who were transferred Hadith from the first source, which is the Messenger of Allah.
Isnad Judgment	Refers to the judgment of Hadith through its Isnad and determining whether it Sahih or weak.
Konia	Name begins with father, mother, brother, sister, uncle, aunt, free, aunt. We say: (Abu Asim, Umm Ali).
Luqab	Name indicates the capacity as Secretary, sincere, ... Etc.
Madar	Is a narrator who tells him most of the narrators in the Isnad tree.
Marfu' (elevated)	What added to the Prophet, peace be upon him from the said and done and the report of the recipe, add either by companion, or without him, it is generally from the sender.
Matn	Speech was saying of the Prophet, peace be upon him and his or approval or other.
Mawquf	Is what tells about the Sahabah of their words and deeds, and so on, depending on them does not exceed to the Messenger of Allah, peace be upon him, and this kind of its Sahih and Hasan and weak.
Mudallis	Who hears what he did not hear from, him formula suspicion that he had heard of it, like saying: for so and so, or so said.

Munqati' (broken)	Is not related Isnad to any face disconnection.
Mursal	What fell from the last Isnad of Tabi, and his image that says Tabi big or small: The Messenger of Allah peace be upon him, or did or so.
Mutassil (continuous) Narrator	Hadith Mutassil is what Contact Isnad, and peace of interruption. Is who the transfer of Hadith from the Messenger of Allah and peace be upon him.
Nasab	Nasab in the language means kinship, said Nasab in the family of so and so: any of them.
Rotba	Ibn Hajar put the narrators into classes according to their experience in terms of the wound and the amendment.
Sahih Hadith	Linguistically are well-built bodies. Metaphorically singular narration conveyed by a trustworthy, completely competent person to another one with neither a serious concealed flaw nor irregularity.
Tabaqa	Ibn Hajar classes the narrators into classes according to their age and deaths
Wahem	Illusion is the deformation occurs for the senses, and reveals how the brain regulates sensory and explains the excitement. Despite the fantasies, distort the truth, usually shared by most people and affecting illusions usually more on the human sense of vision.
Weak Hadith	Means either tangible or incorporeal, The latter celebrated here. Weak Hadith refers to the narrations below the Hasan Hadith due to lacking one of its conditions.

Chapter 1

Introduction

This chapter introduces the thesis by describing the Hadith science and ontology development, the thesis problem, the research objectives, the importance of the research, the scope, and limitation of the thesis work, the methodology, resources, and tools.

1.1 Hadith Science and Ontology Development

Hadith is oral traditions related to the words and deeds of Prophet Muhammad (peace be upon him). The traditional Muslim schools of jurisprudence regarding Hadith constitute an important tool for understanding Qur'an and in all matters related to jurisprudence [1].

The Hadith consists of two parts: the actual narration, known as Matn (المتن); and the chain of narrators through whom the narration has transmitted, traditionally known as Isnad (إسناد). The Isnad consists of a chronological list of the narrators, each mentioning the one from whom he heard the Hadith all the way to the prime narrator of the Matn followed by the Matn itself [1].

Hadith scholars agreed that the Hadith Isnad is required in the narration of Prophetic Sunnah and it is a condition to work it out, otherwise, the Hadith is not accepted, and Imam Abdullah Bin Mubarak said, "*Isnad is part of the religion and without Isnad who ever would have said whatever he wants*" [33].

Research in the Isnad is very important in the science of Hadith. In order to know whether the Hadith is authentic or not, the Hadith scholars follow clear steps in the judgment on the Hadith Isnad that considered as traditional methods.

Nowadays, software tools help judge the Hadith Isnad like electronic Hadith encyclopedias and some websites. Additionally, information retrieval and search engines that related to semantic web can used to serve in deciding the degree of the Hadith Isnad. Scholars such as Al-Albani has agreed and encouraged using computers and programs in serving religion and Hadith.

Ontologies are a semantic web concept that can use in many applications like information retrieval systems and decision-support systems [4]. Based on these applications, ontologies can also be helpful in the process of judgment of Hadith Isnad.

Ontology defined as *"a formal explicit description of concepts in a domain of discourse (classes). Properties of each concept describe various features and attributes of the concept (slots), and restrictions on slots (facets) ontologies together with a set of individual instances of classes constitutes a knowledge base"* [2].

In this research, we design a domain-dependent ontology for Isnad judgment in the Hadith domain. We use the ontology development processes [26] for designing the ontology, which based on structuring Arabic language into a set of equivalent classes, properties, and relationships.

The ontology development processes lead to build an ontology-based Isnad judgment system to support the process of judgment on the Hadith Isnad. The current Hadith systems and traditional methods are studied, the proposed Isnad judgment system (IJS) prototype are implemented, then evaluated by Hadith scholars, it get an excellent results and prove that the accuracy is achieved.

The main reasons for developing ontology are to share a common understanding of the structure of information among people or software agents, where it's a standard schema-based and the changes done on schema not on the ontology and can be queried over the web. In addition, to enable reuse of domain knowledge, and to make explicit those assumptions about a domain that are normally implied [3].

1.2 Statement of the Problem

The judgment of Hadith, whether authentic or not, requires a lot of time, effort, experience and knowledge from Hadith researchers.

There are various applications help researchers in the process of judgment on the Hadith Isnad but they lack to the automation of judgment on the Hadith Isnad. Researchers only have access to the narration chain and the translation of the entire narrator with name, surname, death date, Rotba, Tabaqa, teachers, and students. In addition, some of these applications draw a tree of the Hadith Isnad but leave it to the

researchers after that to study the produced data on the basis and rules set by the Hadith scholars to decide the judgment of the Hadith Isnad.

The problem of this research is how to build an ontology-based Isnad judgment system depending on the rules followed by Hadith scholars in judgment of Hadith. This would support judging the Hadith Isnad that achieves the required accuracy.

1.3 Objectives

In this section, we present both main and specific objectives of the research work.

1.3.1 Main Objective

The main objective of this research is to build an ontology-based Isnad judgment system to support the process of judgment on the Hadith Isnad that achieves the accuracy required by the Hadith scholars.

1.3.2 Specific Objectives

The specific objectives of this research are:

- Studying the current Hadith systems to determine the shortcomings and the requirements respect to accuracy.
- Determining the rules that followed by Hadith scholars to use them in the judgment process.
- Building Hadith Isnad ontology includes Hadith terms, relations, properties and rules.
- Implementing the system prototype uses the ontology for judgment on Hadith Isnad.
- Evaluating the system for accuracy based on a chosen evaluation strategy.

1.4 Importance of the Research

- Hadith considered as the second source for rules in Islam. Without it, we would have lost Sunnah. Therefore, it is very important to study Isnad, and care about it to know whether Hadith is authentic or not.

- The system will facilitate the science of Hadith for scholars and researchers specialists and non-specialists.
- The developed ontology used as a basis for other applications, through which Hadith Matn can be judged.

1.5 Scope and Limitations of the Research

- The scope of research will only be on the six Hadith books, which are Sahih Bukhari, Sahih Muslim, Sunan Tirmithi, Sunan Abu Dawood, Sunan Alnasaei, and Sunan Ibn Majah. The authors of these books worked hard to know the narrators and identify them. They worked the judgment of the Hadith out which is easy to validate our system.
- The narrators information are been taken from Ibn Hajar book [51] because that most of Hadith scholar depend on this book, this book contain all narrators information such as ids, names, Tabaqa, Rotba...etc.
- The research focus only on the judgment process of the Hadith Isnad, and will skip focusing on the Matn because supporting both together is invalid except for one of the following two cases [34]:
 - The first case: follow an Imam when judging the Hadith.
 - The second case: Is an Imam specialized in Hadith judgment.
- The research only addresses and achieves the rule, which the Hadith scholar follows to produce a judgment on Hadith Isnad.
- The system not be fully implemented, but rather a prototype of it are implemented. The prototype provides a proof of concept for the proposed approach.

1.6 Methodology

To accomplish the objectives of the research, the following methodology will be followed:

- Studying and analyzing the current Hadith systems and applications in comparison to our goal, which is accuracy in judging Hadith Isnad.

- Studying Hadith sciences to extract the components of the Hadith ontology (objects, properties, relations and rules).
- Building the ontology using ontology development processes [26] and tools such as, OWL-API, DL-Query, and Protégé, which explained in Section 3.2.4 (Ontology Tools). Then we validate it through the Task-Based Methodology, The development of ontology includes the following tasks:
 - Determine the domain and scope of the ontology.
 - Consider reusing existing ontologies.
 - Enumerate the important terms in the ontology.
 - Define the classes and the class hierarchy.
 - Define the properties of classes (slots).
 - Define the facets of the slots.
 - Create instances.
- Develop a prototype of the proposed approach using Java, Web Services and OWL-API , the development including :
 - Specify the requirements of system.
 - Design the system.
 - Implement the prototype.
 - Test the prototype.
- Evaluate the implementation and confirm that it achieves accuracy. The methods of verification that are used:
 - Al-Albani scholar judgment.
 - Hadith specialist judgment.
 - Task-Based methodology.

1.7 Resources, Methods, and Tools

The following resources and methods will be required:

- Supervisor: for guidance and advice.
- IUG Library: in literature review for accessing books and journals.

- Journal and conference papers to be use in literature review, and through the research.
- Internet connectivity to assist in literature review and work design.
- Ethical issues, and consideration methods: based on the currently available regulations or laws regarding information systems, ethics and privacy, or methods based on standards, principles, and guidelines.
- Network platform including web server, and web client.
- Other research groups working in the same area for advice.
- The required software to conduct this research will be software packages and developing tools, and a platform that supports ontologies. The tools are (Java Netbeans, JSP, and Protégé).

1.8 Thesis Structure

This thesis consists of mainly eight chapters: introduction, state of the art, and review of related works, theoretical and technical foundations, IJS ontology engineering, algorithm for Isnad judgment, Isnad judgment system prototype, results and evaluation, and conclusions and future works. The main points discussed in the chapters listed below:

- Chapter 1 (Introduction): gives a short introduction on the Hadith science, ontology development, and the thesis problem and objectives.
- Chapter 2 (State of the art and review of related works): presents related works to the thesis.
- Chapter 3 (Theoretical and technical foundations): describes the theoretical and technical foundations needed for thesis work, Hadith science, Isnad judgment, ontology concepts, and Web Services.
- Chapter 4 (IJS ontology development): describes the development steps of ontology and the evaluation method.
- Chapter 5 (Algorithm for Isnad judgment): describes using the ontology is in the algorithm of Isnad Judgment, the role of e-Narrator Web Service in this algorithm and the rules of Isnad Judgment.

- Chapter 6 (IJS Prototype): is devoted to present the development phases of the Isnad Judgment prototype.
- Chapter 7 (Results and evaluation): presents the evaluation of (IJS) Prototype using experimental and the evaluation of IJS ontology using Task-Based framework.
- Chapter 8 (Conclusions and future work): discusses the conclusions and presents possible future works.

1.9 Summary

In this chapter, we have introduced the thesis by describing the Hadith science and ontology development it divided into eight sections. In the first section, we introduced the Hadith science and ontology science and the importance of each one of them.

In the second section, we stated the research problem is the possibility of using ontology in supporting the Hadith Isnad judgment.

In the third section, we explained the main objective of this research is to build a system based on ontology to support the process Hadith Isnad judgment.

In the fourth section, we explained the importance of this research; the most important is facilitating access to Hadith Isnad judgment by specialists and non-specialists.

In the fifth section, we stated the scope and limitation of this research, the limitation is that the IJS system will only judge the Hadith Isnad and not the Hadith and we are building a prototype for the system not a complete system.

In the sixth section, we presented the methodology that will follow in this research comprehensive the method of building ontology, phases of building the prototype, and evaluation methods in terms of accuracy.

In the seventh section, we presented the tools and resources that we will use during this research. In the eighth section, we explained how this research divided in terms of chapters and titles.

Chapter 2

State of the Art and Review of Related Works

In this chapter, we study and investigate different related works. They introduced and analyzed with respect to the thesis problem to show how far these works address the requirements of our thesis problem. Parts of the related works can be a basis for solving the thesis problem, but no one can present a complete solution. The related works and researches focus on Hadith and ontology. The following sections presented to discuss Hadith electronic encyclopedias, Hadith science researches, Arabic and Islamic ontology development, and ontology and decision support.

2.1 Hadith Electronic Encyclopedias

There exist various electronic encyclopedias that are helpful in Hadith judgment like Shamela library [9], encyclopedia of the narrators of Hadith (Noor Islam) [35], encyclopedia of Harf the nine books [10], Aldourar Alsunnia site [11].

From our review, we noticed that they are all similar in their outputs (related to judgment of Hadith process). They all attributed to the researcher. They usually provide the narrators' chain and the biography of the entire narrator with name, surname, death date, Rotba, Tabaqa, teachers, and students.

However, Encyclopedia of Harf additionally draws a tree of the Hadith Isnad. Aldourar alsunnia website prioritizes the author's judgment such as Imam Bukhari, Imam Muslim, and Al-Albani.

In spite of all these features in Hadith electronic encyclopedias, they lack the automation of judgment on Hadith Isnad. This would result in time-consumption, and much of the effort lies on the researcher of the Hadith. We want to support this process in our research by using ontology.

2.2 Hadith Science Researches

iTree [1] automatically generates the narrators' chain of a given Hadith and graphically displays it. This process involves creating natural language lexer, performing shallow parsing, building syntactic analyzer, and finally a graph presenter that displays the narrators' chain graphically.

e-Narrator [29] parses a plain Hadith text and automatically generates the full narration tree. It involves parsing and annotating the Hadith text and recognizing the narrators' names. It uses shallow parsing along with a domain specific grammar to parse the Hadith content. Moreover, it uses a transformation mechanism based on ontology to represent the narration chain in a standard format and then graphically renders its complete tree. The experiments on sample Hadiths show that it has a very good success rate. But they have a limitation in the scope of application that it designs on Sahih Muslim Book only, the effect of this limitation that it can't draw the Isnad tree for all Hadith's in the Hadith six books. It only generates a narration tree and does not give any judgment on the Isnad.

Because e-Narrator application gives very good success rate, we depend on it in our research, we convert this application to a Web Service to gets the narrators chain from the Hadith. In addition, we benefited from the properties and the relations on HadithRDF ontology to reach our research goal in the development the IJS ontology. See Section 5.1(Separating the Isnad from Hadith (Matn)).

AuthenTique [5] is a complete-text-mining system that used for knowledge extraction from a database of Prophetic Traditions (Hadiths). It provides a list of classified Hadiths according to their degrees of similarity with respect to the user's query. The implemented methods of text mining in AuthenTique based on vector space model, Term Frequency Inverse Document Frequency (TFIDF), and cosine measure.

In [6], they use techniques from data mining to extract Islamic knowledge from its resources, and to represent that knowledge meaningfully for the user. Moreover, their study concentrates on Hadith as knowledge resource, and proposes approach to classify Hadith to its categories using supervised learning classification. The finding of their study is that there are several ways to extract knowledge from Hadith depending on the goal of the knowledge. The used techniques are different according to the purpose of the knowledge. Unsupervised learning classification and supervised learning classification are the two famous techniques that used to classify Hadith. The first techniques required to clarify all rules that are necessary for classification. On the other hand, supervised learning can learn from a training set to induct the rules.

In [7], they present the results of expanding user's query using Malay thesaurus in the process of searching Malay documents from Malay Hadith retrieval system. The

results obtained show that the retrieval effectiveness improves by 4% when thesaurus employed in the process of retrieving Malay translated Hadith documents compared to when single term queries employed.

The above works show a growing interest in studying the subject of Hadith and taking advantage of modern technologies in serving Hadith such as classifying it using data mining, extracting knowledge from it depending on the goal of the knowledge. This research adds a new approach to the judgment on Hadith Isnad using ontologies, based on the rules that sets by Hadith scholars and used in the judgment of Hadith Isnad.

2.3 Arabic and Islamic Ontology Development

Few published papers address the issue of Islamic and Arabic language such as [8]. Here, ontology development, based on the Solat (prayer) domain, is an attempt to understand the characteristics of an ontology development including a concept of Islamic knowledge. In Solat-based ontology development, they use the up-coming methodology, based on information from multiple sources provided by domain experts. It involves Qur'an, authentic Hadith, and books that focus on the Shafie's school of thought. The ontology developed from information gathered by domain experts, and assigned to the ontology expert in the form of a set of concepts, relationships, and definitions. This methodology can subdivided into the following stages: determine the domain and scope of the ontology, consider reusing existing ontologies, enumerate the important terms in the ontology, define the classes and the class hierarchy, define the properties of classes (slots), and define the facets of the slots.

To enrich the Arabic content in the Semantic Web, [2] proposes a model for representing Arabic knowledge in the computer technology domain through using ontologies. The model starts by elicitation users' informational needs. Ontologies play a major role in supporting information search and retrieval processes of Arabic blogs on the Web. The ontology designed based on WSMO framework for modeling semantic Web Service s. A model driven architecture is used, and forward engineering approach adopted. They started by modeling the ontology first and then using this ontology as a domain model to form the basis of the generation of the semantic search engine. The process of ontology engineering encompasses many phases. It starts with eliciting the domain knowledge to represent by the ontology.

The above works show growing interest Arab and Islamic researchers adopt to build Arabic ontologies to increase the Arabic content on the Semantic Web. This is very helpful in the stage of development our Hadith ontology.

2.4 Ontology and Decision Support

In [12], they describe a model for building and integrating ontologies that aim at building a domain of common ontology and use it in the approach of context mediation for assisting decision-making. This happens when delivering data to decision support systems in an acceptable form. The use of ontologies in the approach of context mediation allows reducing a number of context conversions and, as a result, improving performance of the process of mediation. The process of building ontology figured out as follows. First, identification of ontology purpose needed to achieve semantic interoperability among data sources and DSS. Second, pre-processing, conceptualizations, pre-integration and integrating are important to build the ontology. Third, evaluation and documentation applied.

In [13], project-handling module based on decision-making ontology model (PMDOM) proposed to accelerate communication of decision-making, and help in the selection of projects. Based on PMDOM, an automotive body assembly domain ontology model is set up and the decision-making support system (DSS) has been implemented by using java. Protégé and OWL used to set it up and describe it. The main classes that are included are decision-making, participants, project, and decision-making template. The relationships between these classes are of dependent relationship. Other relationships between classes are association relationship. This domain ontology made up of assembling body information, joint information, parts information, and tolerance. It linked PMDOM to expediently pickup corresponding domain parameters and information. The assembly sequence, assembly quality, assembly cost, and joint type joined as targets of performance evaluation. These targets evaluated and decided by experts. At last, some optimized rules entered into the DM template to achieve the optimized methods scheme. Experts depending on domain knowledge choose these rules.

In [14], they propose framework for ontology-based group decision support system (ONTOGDSS), connected to decision process, which exhibits the complex structure of decision-problem and decision-group. It is capable of reducing the complexity of

problem structure and group relations. The system allows decision makers to participate in-group decision-making through the web environment via the ontology relation. It facilitates the management of decision process as a whole, from criteria generation, alternative evaluation, and opinion interaction to decision aggregation. The embedded ontology structure in ONTOGDSS provides the important formal description features to facilitate decision analysis and verification. They firstly present an ontology-based construction approach according to the complex structure of group and its task in reality. Then, based on ontological problem representation, they provide designs of group decision process and the framework of support system. Finally, they present an implementation on ontology-based decision problem representation and extraction.

In [15], they discuss the design of a framework for supporting multi criteria decision making using ontology. The role of this framework intended to use as general decision-making aid. An extraction engine supports it in order to find salient information from the unstructured documents. The benefit is automation in transforming unstructured to structured data for decision-making analysis. The framework divided into three main phases. The first phase is ontology modeling; the second phase is information extractor engine; and the third phase is DSS models.

The above works show that ontologies are useful in decision-making such as reducing the number of context conversions, improving performance, and facilitating decision analysis and verification. This could be helpful in reaching a decision on the judgment of correctness or weakness of the Hadith.

2.5 Summary

In this chapter, we have studied and investigated some related works on Hadith science and ontology. This chapter divided into four sections. In the first section we studied the encyclopedias of Hadith and identified the advantages and capabilities of it, we found that all of these encyclopedias lack to automation of judgment on the Hadith Isnad and the maximum offer it view the full information of narrators.

In the second section we studied, the research's that has been on the Hadith and the most important of these researches are two research papers, which is iTree and e-Narrator. These two research papers used in drawing the Isnad tree. We depend on it in our research. We use e-Narrator in the IJS system as a Web Service to extract the Isnad

form the Matn. In addition, we use the HadithRDF ontology as a basis for developing the IJS ontology. We use it to identify narrators' names and information that needed in the process of Isnad judgment.

In the third section, we have studied supporting Arabic content in the semantic Web and found weakness in this area. In fourth section, we studied the research that used the ontology in decision support to make sure, that we can use the ontology in supporting the process of Isnad judgment and this study brought us to the possibility of it.

Chapter 3

Theoretical and Technical Foundations

In this chapter, the fundamental concepts and technical knowledge, which represent the basis for understanding of the thesis work, presented. The Hadith science and Isnad Judgment discussed, followed by ontology and semantic web.

3.1 Hadith Science and Isnad Judgment

Hadith terminology science is concerned with specifying the acceptability of narrations, attributed to the Prophet Mohammed (PBUH), the conditions of the narrator in terms of reliability, and the significance of the text, through which rules are determined [33].

Rules were set to distinguish between what accepted and what refused, regarding the narrations of the Prophet. The scholars of Islam put a science called Mustalah Al-Hadith, Hadith terminology, by which the prophetic traditions examined in order to cite, review, and determine the different narrations from one layer to another and from one generation to another. The Islamic scholars were the first to study Isnad accurately to determine who are the trusted and weak narrators.

3.1.1 The Importance of Isnad

The Isnad allows researchers to distinguish between Sahih (sound) and Weak Hadith, Mutassil (continuous), and Munqati' (broken), Marfu' (elevated) and Mawquf (related to narrations of companion) [40].

The narrators transmitted the sayings of the Prophet. None of them belied one another since the early times of the Prophet. They were all confident, and faith fell their hearts. They did not need to look up the conditions of the narrator, until Fitna (Islamic civil war) took place in Islam. The Sahabah (prophet companions) and the followers did their best to maintain the Hadith, and Muslims concentrated on following the Isnad and examine the conditions of the narrators.

Imam Mohammed Ibn Sirin says, “*they (the Sahabah) did not ask for Isnad, but when the Fitna occurred, they identified the Sunni men to follow them Bida' men to avoid them*”

[42]. Muslims paid much attention to Isnad and they considered it as part of religion. Ibn al-Mubarak said "*Without Isnad whoever would have said whatever he wants*" [43].

In the 2nd century A.H., writing activated, and Muslim scholars determined the accepted narrators and the rejected ones. In the 3rd century A.H., Hadith subdivided into many specific branches, i.e., the science of Sahih Hadith, the science of Mursal Hadith and the science of biography. Both the second and the third centuries A.H. were the most flourished centuries in the Islamic history.

Insnad is only limited to the Islamic nation. The cultural heritage of all the other nations has never transmitted with Isnad. Isnad accompanied the Prophet's Sunnah. Ibn Hazm says, "*Allah gave only the Islamic nation a trait that a trustworthy person relates what he says to another trustworthy person, until it is all related to the Prophet (PBUH)*" [44]. Abu Ali Al-Jayyani said "*Allah granted this ummah (nation) alone three traits: Isnad, genealogy and assigning*" [44].

3.1.2 The Importance of the Science of Judgment on Isnad

- This science is one of the most elegant Islamic sciences because it depends on Sunnah, which is the second legislative source in Islam.
- It helps differentiate between the accepted and rejected Hadith, or, in other words, sahih and weak Hadith.
- Many other Islamic sciences rely on it. Any fakih needs to get to know Hadith before he gives rules.
- It defends and maintains the Sunnah, and it shows the weak Hadith and the mawdu' (fabricated) Hadith.
- It is the outcome of all sciences of Hadith. It shortlists the Hadith for being: Sahih, Hasan, or Weak [40].

3.1.3 Judgment of Isnad VS Judgment of Hadith

Judging on the Isnad should meet three conditions: authenticity, accuracy, and continuity of Isnad. Judging on Hadith, however, meets these three conditions, in addition to being detached away from flaw and oddness. This requires searching and

comparing Isnads to one another and Matn to one another. This is an advanced level of judgment on Hadith [40].

3.1.4 Steps of Studying Isnad

First Step: Identifying the narrator

The Hadith Scholars concerned by men regards to all respects. They tried to get rid of ambiguity and set the names of the narrators, along with their fathers, their surnames, and their Nasab. Hadith Scholars cited all these stuff accurately; they branded each narrator precisely; they documented their birth dates and death dates clearly. Al-Sakhawi said, "*Documenting birth and death dates is a great art that is largely praised by Muslims*"[46].

The Islam scholars talked a lot to tell how important this science is. Sufyan al-Tawri said, "*We used history, when some of the reporters lied*" [45]. Further, Hafs Ibn Ghayyath said "*if you accused an old man of something, count his age*" [46]. Moreover, Hassan Ibn Zaid said "*we did count on history only to expose the liars*" [50].

This is an essential step because the names and the surnames of men might be the same, which would put the researcher into illusions. Al Molallimi said, "*The names are so many, and confusion might happen*". Muhadith may say something about somebody, but the listener may think that he is talking about somebody else. For example, Al Moghera Ibn Abdullrahman Ibn Al Harith Ibn Hisham Al Makhzomi is one of the narrators. However, Al Moghera Ibn Abdullrahman Ibn Abdullah Ibn Khalid Ibn Hizam Al Hizami is another narrator [50].

Second Step: Defining the narrator, whether he is weak or trusted as follows:

- Is the narrator trustworthy weak or needs further investigation?
- Is he quoting his Sheikhs equally?
- Are his students quoting him equally?
- Are all of his narrations sound in his lifetime or they changed? Did the narrator transmit after he changed or not? Both the change and Ikhtelat must identify in detail.
- Are all of his narrations equal? If not, why?

- Has the narrator ever called Mudallis? Did somebody prove that the narrator is Mudallis? What is the kind of Tadlees? Does his Tadlees happen when it comes to all his Sheikhs or one of them? How do the Imams deal with his Tadlees?
- Does he narrate Hadith that is Mursal? If so, did he manage to listen to his teacher?

Third Step: Naming the narrators and the Madar, what did they agreed on and what they did not. The following must take into consideration:

- Making sure that the chain of Isnad is sound from the first narrator up to the Madar. If not, then one should not depend on it, and it should point at to warn.
- Making sure than no confusion happens in terms of the narrator up to the Madar.
- Studying the conditions of the narrator and clarifying what rank he is occupying.
- Arranging the narrations that quoted from the Madar according to what agreed on and what was not.

Fourth Step: Make comparisons between the narrations and showing which is better from the others, and telling why.

This is the most important step of this study, through which we differentiate between the good and bad critic. In addition, it helps us identify how accurate and knowledgeable the researcher is. Further, it helps us realize how good and perfect Imams are. This is all because Imams followed some rules to compare between the narrations [50].

3.1.5 Judging on Isnad

Judging on Isnad at this particular phase is the fruit of the studying Isnad in the previous steps. The researcher must decide precisely as follows [40]:

1. Identifying the narrator, the narrator must identify without confusion.
2. Authenticity of the narrator, identified either by fame or by recommendations.
3. Check degree of authenticity, weak or trustworthy.
4. Check the Continuity of Isnad.

This would help judge the Isnad. If the narrator is trustworthy, then the Isnad is Sahih. If he is weak, then the Isnad is weak as well. This presented as in Table 3.1:

Table 3.1: Isnad Judgment Rules

Narrators' Rotba	Flaw	Condition	Judgment
2-3	None		Sahih
	Tadlees	Degree of Tadlees is (1-2)	Sahih
		If the degree of Tadlees is 3, we should say "So-and-so reported"	Sahih otherwise it is Weak
		If the degree of Tadlees is 4	Weak
	Ikhtilat	Before Ikhtilat	Sahih
		Not distinguished	Weak
		After Ikhtilat	Weak
	Irsal	Not mentioned	Sahih
		Mursal	Weak
	Bida 'a	No Bida'a	Sahih
		Bida'a	Weak
	Wahem	Hadith is centered	Weak
		No continuity	Weak
	4	None	
Tadlees		Degree (1-2)	Hasan
		If it Is degree 3, we should say "So-and-so reported "	Hasan otherwise it is Weak
		Degree 4	Weak
Ikhtilat		Before Ikhtilat	Hasan
		Not distinguished	Weak
		After Ikhtilat	Weak
Irsal		Not mentioned	Hasan
		Mursal	Weak
Bida'a		No Bida'a	Hasan
		Bida'a	Weak
Wahem		They told about the Hadith	Weak
		No continuity	Weak
5-12			

3.2 Ontology and the Semantic Web

In this section we present the definition of ontology, how we can develop it and the tools and software's that are using to implement it.

3.2.1 Introduction

With development of Internet, users found it is more difficult to get their necessary information on the web. How to search information on the web quickly and effectively has become an important problem. Designing a better search technology or adding the semantics to web resources is the best way to improve the quality of information

searching. Semantic Web is the way to represent the semantics of documents and enable these semantics to use by web applications and intelligent agents [16].

Berners-Lee proposed the systematic structure of Semantic Web, which includes of seven layers: UNICODE and URI, XML, RDF, Ontology, Logic, Proof, Trust as shown in Figure 3.1.

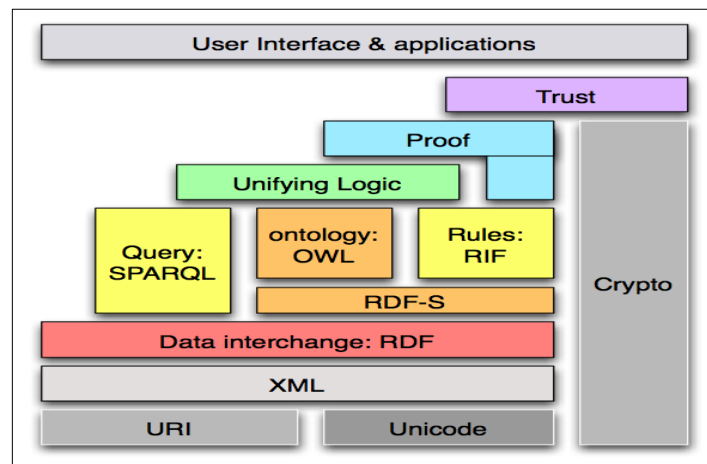


Figure 3.1: Berners-Lee Semantic Web Layers [16]

In these layers, the hardcore layers are XML, RDF, and ontology they used to describe semantic of web information. XML is a kind of language to describe resources. It fits to express all kinds of information and been regarded as a standard for data exchange. RDF recommended as a protocol to describe and deal with metadata by W3C, It provide some information understood by computer for applications' alternation on the web. XML and RDF can all provide semantic for computer resources [17].

Ontology is a formal specification of a shared conceptualization [18]. Ontology is a formal explicit representation of concepts in a domain, properties of each concept describes characteristics and attributes of the concept known as slots and constrains on these slots. Ontology is a shared conceptualization with a clear hierarchy and a strong support for logical consequences [18].

Ontology contains a set of specific and clearly described classes or concepts, property of the concepts, slot, restriction, facet and a series of instance related to one class, which combines to form the knowledge storage [19]. Class is the core of ontology, which describes the concepts in some domain. Slot describes the property of the class and the instance.

3.2.2 Ontology Development

A team of people, such as domain experts and ontological engineers, normally carries out the development of ontology. Noy and McGuinness [3] address reasons for developing ontologies and enumerate the stages involved in developing ontology.

In the process of creating the ontology, some basic rules should emphasize in the design of the ontology. According to [3] other things that need to be considered are:

- There is no right way to model a domain - there is always a suitable alternative. The best solution depends on the application to be developed and thought out and extension that you anticipate and develop.
- Ontology development is a repetitive process.
- The concepts in the ontology should be close to the object (physical or logical) and relationships in the domain that you study. Normally nouns (objects) or verbs (relationships) in sentences will clarify and explain our domain.

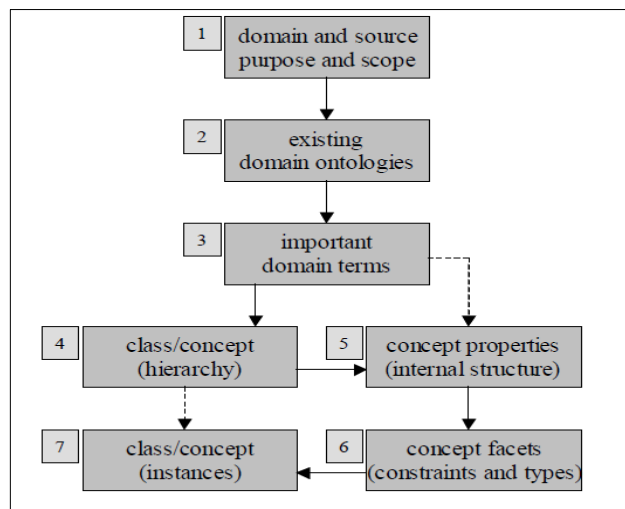


Figure 3.2 Ontology Development Processes [26].

The general stages in the design and development of ontology are as in Figure 3.2:

Step 1: Determine the Domain and Scope of the Ontology

This step defines the purpose and boundaries of the ontology. That is, answer several basic questions: what domain will the ontology cover? What is the purpose of the ontology? For what sorts of questions should the information in the ontology be able to provide answers? [26].

The answers to these questions may change during the ontology-design process, but at any given time, they help limit the scope of the model.

Step 2: Consider Reusing Existing Ontologies

This step is to ascertain if ontology has developed previously in the same subject area. If such ontology exists, it is easier to modify the existing ontology to suit ones needs than to create a new ontology [26]. Reusing existing ontology may be required if our system needs to interact with other applications that have been committed to a specific ontology or controlled vocabulary.

Step 3: Enumerate the Important Terms in the Ontology

This step can viewed as a brainstorming activity [28] and it is useful if we could list all the words that we want to use, either in the form of a statement or explanation to the user. The following questions guide this brainstorming: What the terms would like to talk about? What properties do those terms have? What needed to say about those terms?

Step 4: Define the Classes and the Class Hierarchy

There are several possible approaches in developing a class hierarchy [27]: a top-down development process, which starts with the most general concepts and subsequent specialization of the concepts. Bottom-up starts with the most specific concepts or classes, the leaves of the hierarchy with subsequent grouping of these classes into more general concepts. Middle-out is a combination of the top-down and bottom-up approaches starts with the salient concepts first and then generalize and specialize them appropriately.

Step 5: Define the Properties of Classes (Slots)

This step used to describe the attributes or properties of the classes. These properties defined as the ‘slots’ of the models [28]. Once the classes have defined, the next step is to describe the internal structures (properties) of the concepts. Again, these should be readily available from the list produced because of Step 3 [26].

Step 6: Define the Facets of the Slots.

This step involves attaching facets to the properties, i.e., describing the value type, allowed values, the number of allowed values (cardinality) and other features that are necessary. In this way, constraints placed on the types of data that allowed.

Step 7: Create Instances

The last step allows the data to entered and displayed. An instance (or individual) is the information that entered into the knowledge base. To create an instance the following method needs to carried out:

1. Choose a class.
2. Create an instance and name instance after the source.
3. Fill the slot values.

3.2.3 Ontology Evaluation

The evaluation of the quality of ontology is an important part of ontology development. Because this we present in this section the evaluation methodology we carried out to evaluate the IJS ontology.

An ontology can be evaluated against many criteria: its coverage of a particular domain and the richness, complexity and granularity of that coverage; the specific use cases, scenarios, requirements, applications, data sources it was developed to address, formal properties such as the consistency and completeness of the ontology and the representation language in which it is modeled [30].

There are various approaches to the evaluation of ontologies, depending on what kinds of ontologies are evaluate and for what purpose, most evaluation approaches fall into one of the following categories [31]:

- Based on compare the ontology to a golden standard, that may itself be ontology.
- Based on use the ontology in an application and evaluating the results (Task-Based) [32].
- Involve comparisons with a source of data (e.g. a collection of documents) about the domain to cover by the ontology.
- Evaluate by humans who try to assess how well the ontology meets a set of predefined criteria, standards, requirements, etc.

Task-based evaluations offer a useful framework for measuring practical aspects of ontology deployment such as the human ability to formulate queries using the query language provided by the ontology. The accuracy of responses provided by the system's inferential component, the degree of explanation capability offered by the system, the coverage of the ontology in terms of the degree of reuse across domains, the scalability of the knowledge base, and the ease of use of the query component.

Task-Based evaluations can leverage use-cases or scenarios to characterize the target knowledge requirements [30]. In a Task-Based evaluation, the results should show the following shortcomings:

- Insertion errors indicating superfluous concepts,
- Deletion errors indicating missing concepts, and
- Substitution errors indicate off-target or ambiguous concepts.

With this, we can provide performance measures that can:

- Evaluate one or more ontologies in terms of their performance on a given task (ideally to measure only the ontology-specific aspect of the performance),
- Quantify the respective gains and losses of the insertion, deletion and substitution errors,
- Populate/improve the ontology as derived from the individual error type specific results, and
- Re-evaluate the respective performance increases resulting from the improvements.

By applying this evaluation scheme, we can test and measure the respective improvements that brought about by learning approaches that target the same levels and issues in the ontology learning and population field [32].

3.2.4 Ontology Tools

OWL (Web Ontology Language) is standard of ontology language recommended by W3C. It is located on the top of the ontology language stack. Therefore, the best method to manage decision-makings in collaborative design is set up a decision-making ontology and use OWL to describe [13].

OWL is an object-oriented representations that consist of classes (concepts), properties (relationships), and individuals [13]. Individuals represent objects in the domain, maybe members of one or more classes. Properties describe the relationships between individuals, link two individuals. Classes, also known as sets members of classes, share some properties or characteristics.

OWL ontologies maybe categorized into three species or sub-languages [20]: OWL-Lite, OWL-DL, and OWL-Full. A defining feature of each sub-language is its expressiveness. OWL-Lite is the least expressive sub-language it intended to use in situations where only a simple class hierarchy and simple constraints needed. OWL-Full is the most expressive sub-language it intended to use in situations where very high expressiveness is more important than being able to guarantee the decidability or computational completeness of the language. It is therefore not possible to perform automated reasoning on OWL-Full ontologies. The expressiveness of OWL-DL falls between that of OWL-Lite and OWL-Full. OWL-DL considered as an extension of OWL-Lite and OWL-Full an extension of OWL-DL.

There are two important types of properties in OWL [21]: data type properties and object properties. Data type properties help describe individuals, they are not typically used to describe classes and are certainly not dependent on classes. The set of allowable values for data type properties are typed literals. Typed literals are literal values (not abstract objects) with a specific data type. Object properties allow you to create associations or relationships between two individuals. That means the subject and the object the triple are both individuals.

Protégé is a free, open-source platform that used to construct domain models and knowledge-based applications with ontologies. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé can customize to provide domain-friendly support for creating knowledge models and entering data. Further, Protégé can extend by way of a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge-based tools and applications [22].

The OWL API is a Java API and reference implementation for creating, manipulating, and serializing OWL Ontologies [23].

The Protégé-OWL API is an open-source Java library for the Web Ontology Language and RDF(S). The API provides classes and methods to load and save OWL files, to query and manipulate OWL data models, and to perform reasoning. Furthermore, the API optimized for the implementation of graphical user interfaces [24].

The DL Query provides a powerful and easy-to-use feature for searching a classified ontology. The query language (class expression) based on the Manchester OWL syntax, a user-friendly syntax for OWL DL that fundamentally based on collecting all information about a particular class, property, or individual into a single construct, called a frame [25].

3.3 Summary

In this chapter, we have discussed the theoretical bases that used in this research. In the first section, we defined the Hadith science and the importance of Isnad. Therefore, the idea of research spins about Isnad judgment. We explained the difference between Isnad judgment and Hadith judgment. In addition, we explained the steps of studying and judging the Isnad.

In the second section, we identified the ontology where it used in supporting the process of Isnad judgment. In addition, we explained how to build ontology and the steps that must follow. We explain the Task-Based framework, which used for evaluating the ontology. We explain the tools that will be used in the construction and programming of the IJS system.

Chapter 4

IJS Ontology Development

In this chapter, we present the steps to develop the IJS ontology based on the Hadith domain and we present the evaluation of the IJS ontology.

4.1 IJS Ontology Development

The IJS ontology is developed with the assistance of a domain expert. He helped to identify concepts, relationships, and definitions of the Hadith domain. The development methodology is performed according to the steps listed in Section 3.2 (Ontology Development). The steps as applied here are:

Step 1: Determine the Domain and Scope of the Ontology

The first step in ontology development is defining ontology domain and scope, in which the ontology will be developed in order to answer some basic questions:

1. *What is the domain that the ontology will cover?*

The domain of the ontology will cover judging the Hadith Isnad.

2. *What is the use of the ontology?*

The ontology is to provide a schema-base of narrators and Hadiths, which is used by the IJS system to make judgment on the Hadith Isnad.

3. *What types of questions would be answered by the information contained in the ontology?*

The ontology would provide comprehensive answers to questions relating to Hadith domain and like:

- What is the First Name of Narrator?
- What is the Nasab of Narrator?
- What are the Tabaqa and Rotba of Narrator?
- Who are Narrator Teachers and Students?
- What are the Hadiths that Narrator had suffered Ikhtelat when he narrated them?
- Who are the Narrators whom the Narrator had Irsal about them?

4. Who will use the ontology?

The ontology will be available by the IJS system to identify narrators' information.

Step 2: Reuse Existing Ontologies

We use HadithRDF ontology [1] [29] as a basis for developing the IJS ontology as explained earlier in Section 2.2 (Hadith Science Researches). We used it to identify narrators' names and information's that needed in the process of judgment the Isnad. HadithRDF ontology specification was written by using RDF. HadithRDF ontology can be written as n-triple in the form of subject – predicate – object. HadithRDF is designed to span a wide range of Hadith books in the Hadith corpus, e.g., Sahih of Bukhari. The books have chapters and each chapter has multiple Hadiths with each narrated by several narrators. Taking all the above into consideration, Figure 4.1 illustrates HadithRDF ontology design. Shown are classes and the set of properties along with the proper associations between them.

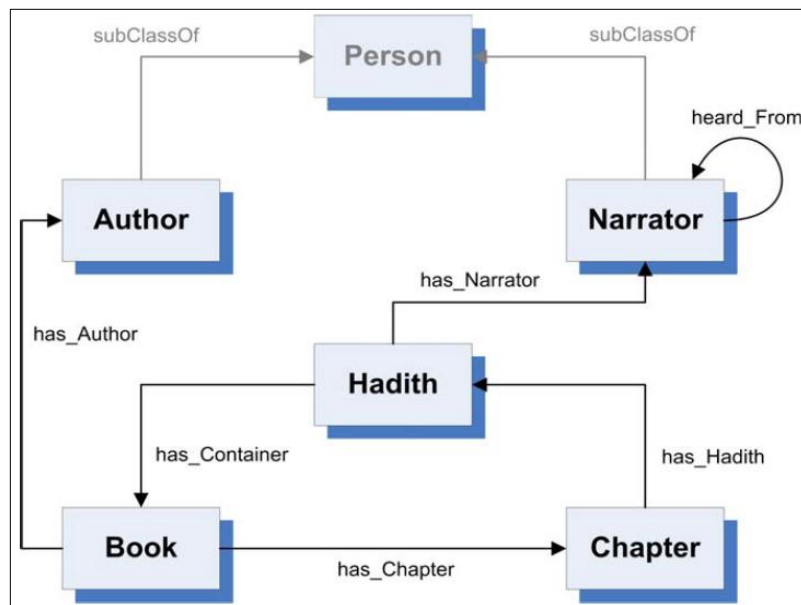


Figure 4.1: HadithRDF Ontology [29]

Step 3: Enumerate the Important Terms in IJS Ontology

This step represents a brainstorming activity. See Section 3.2.2 (Ontology Development). We use the terms in HadithRDF Ontology [1] [29] as a basis of the terms that we want to use. We add more properties for these terms by studying the science of judging Hadith Isnad which is explained in Section 3.1 (Hadith Science and Isnad Judgment), and through analyzing the structure of Hadith in the six Hadith books

which are listed in Section 1.5 (Scope and Limitations of the Research). The following questions guide our brain storming activity to determine the terms:

1. *What are the main terms that we want to talk about it?*

The main terms we talk about are *Books, Hadith, Person, Author, and Narrators*.

2. *What are the properties that owned by these terms? What needed to say about those terms?*

- The Book term has the following properties: *hasAuthor, hasChapter, ContainerOf, hasHadith, Content, NoOfPages, Title, Topic and YearOfPublication*
- The Hadith term has the following properties: *HadithOf, hasContainer, hasNarrator, BedaaHadiths, WahemHadiths, IkhtelatHadiths, Title and Topic*
- The Person term has the following properties: *Id, Name, FirstName, FatherName, FatherNickName, GrandFatherName, GrandFatherNickName, FourthName, FourthNickName, FifthName, FifthNickName, Nasab, BirthYear, DeathYear, Gender, LiveIn, OriginallyFrom, and DenotedBy*.
- The Author term has the AuthorOf property.
- The Narrator term has the following properties: *HeardFrom, TransferTo, notHeardFrom, NarratorOf, IrsalAbout, FatherOf, hasFather, MotherOf, hasMother, BrotherOf, hasBrother, GrandFatherOf, hasGrandFather, UncleOf, hasUncle, CousinOf, hasCousin, NephewOf, hasNephew, DaughterHusbandOf, hasDaughterHusband, GrandSonOf, hasGrandSon, MawlaOf, hasMawla, AttributedToGrandfather, Konia ,Luqab ,Rotba, Tabaqa, Tadlees, hasBedaa ,hasIkhtelat, hasIrsaland hasWahem*.

Step 4: Define Classes and Class Hierarchy of IJS Ontology

This step starts by defining classes. From the list, which created in Step 3, terms selected whether they describe objects having independent existence or terms that describes these objects. The terms in Table 4.1 are classes in the ontology and will become anchors in the class hierarchy. Classes also organized into a hierarchical taxonomy.

Table 4.1: IJS Ontology Classes

No.	Class	Description
1	Hadith	Hadith represents the Hadith Matn and Isnad
2	Person	Person represents information about a human being
3	Narrator	Represents a person who narrates (tells) Hadith
4	Author	Represents author of the book
5	Book	Represents book of Hadith
6	Chapter	Represents chapter in Hadith book

There are three possible ways to develop the class hierarchy [27]: top-down approach, bottom-up approach, or combination of both. In our approach, we use the top-level concept such as *Book*, *Chapter*, *Hadith*, and *Person*. Then we generate all other classes that could expand from *Person*.

The IJS ontology produces an OWL source code. A full copy of the OWL source code is the found in Appendix C: OWL Source Code). A sample of the code listed in Figure 4.2 to represent the classes in OWL format.

```

<!-- http://protege.org/ontologies/Hadith.owl#Author -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Author">
  <rdfs:subClassOf rdf:resource="&ontologies;Hadith.owl#Person"/>
</owl:Class>
<!-- http://protege.org/ontologies/Hadith.owl#Book -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Book"/>
<!-- http://protege.org/ontologies/Hadith.owl#Chapter -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Chapter"/>
<!-- http://protege.org/ontologies/Hadith.owl#Hadith -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Hadith"/>
<!-- http://protege.org/ontologies/Hadith.owl#Narrator -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Narrator">
  <rdfs:subClassOf rdf:resource="&ontologies;Hadith.owl#Person"/>
</owl:Class>
<!-- http://protege.org/ontologies/Hadith.owl#Person -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Person"/>

```

Figure 4.2: IJS Ontology Main Classes in OWL

Step 5: Define the Properties of Classes (Slots)

Once we defined the classes, we clarify and reflect the internal structure of concepts. This considered the property of the developed classes. These properties extracted from classes that illustrated in Table 4.1. For Example: *Person* has two types of *Person*, which are *Narrator* and *Author*. Every instance of *Narrator* has properties such as *HeardFrom*, *hasFather*, *TransferTo*, *NarratorOf*. Table 4.2 illustrates the object properties structure of IJS ontology.

Table 4.2: Hadith Ontology Object Properties

Object Properties	Domain	Range	Description
<i>AuthorOf</i>	Author	Book	Book that the author/person is an author
<i>has Author</i>	Author	Book	Refers to the person who authored the book
<i>ChapterOf</i>	Book	Chapter	Book that hosts the chapter
<i>hasChapter</i>	Book	Chapter	Chapter that this book hosts
<i>ContainerOf</i>	Book	Hadith	Hadith that this book contains
<i>hasContainer</i>	Book	Hadith	Book that contains this Hadith
<i>HadithOf</i>	Chapter	Hadith	Chapter that hosts the Hadith
<i>hasHadith</i>	Chapter	Hadith	Hadith that this chapter contains
<i>HeardFrom</i>	Narrator	Narrator	The narrator who hears the Hadith
<i>TransferTo</i>	Narrator	Narrator	The narrator who told the Hadith
<i>notHeardFrom</i>	Narrator	Narrator	The narrator who did not hear the Hadith
<i>NarratorOf</i>	Hadith	Narrator	Hadith the narrator is narrating
<i>hasNarrator</i>	Hadith	Narrator	Narrator who narrates Hadith
<i>IrsalAbout</i>	Narrator	Narrator	Narrator who Irsal Hadith About Narrator
<i>BedaaHadiths</i>	Hadith	Narrator	Hadith Contains Bedaa of Narrator
<i>WahemHadiths</i>	Hadith	Narrator	Hadith Contains Wahem Narrator
<i>IkhtelatHadiths</i>	Hadith	Narrator	Hadith form Ikhtelat of Narrator
<i>FatherOf</i>	Narrator	Narrator	Narrator is Father of Narrator
<i>hasFather</i>	Narrator	Narrator	Narrator has Father Narrator
<i>MotherOf</i>	Narrator	Narrator	Narrator is Mother of Narrator
<i>hasMother</i>	Narrator	Narrator	Narrator has Mother Narrator
<i>BrotherOf</i>	Narrator	Narrator	Narrator is Brother of Narrator
<i>hasBrother</i>	Narrator	Narrator	Narrator has Brother Narrator
<i>GrandFatherOf</i>	Narrator	Narrator	Narrator is grandfather of Narrator
<i>hasGrandFather</i>	Narrator	Narrator	Narrator has grandfather Narrator
<i>UncleOf</i>	Narrator	Narrator	Narrator is Uncle of Narrator
<i>hasUncle</i>	Narrator	Narrator	Narrator has Uncle Narrator
<i>CousinOf</i>	Narrator	Narrator	Narrator is Cousin of Narrator
<i>hasCousin</i>	Narrator	Narrator	Narrator has Cousin Narrator
<i>NephewOf</i>	Narrator	Narrator	Narrator is Nephew of Narrator
<i>hasNephew</i>	Narrator	Narrator	Narrator has Nephew Narrator
<i>DaughterHusbandOf</i>	Narrator	Narrator	Narrator is Daughter Husband of Narrator
<i>hasDaughterHusband</i>	Narrator	Narrator	Narrator has Daughter Husband Narrator
<i>GrandSonOf</i>	Narrator	Narrator	Narrator is Grandson of Narrator
<i>hasGrandSon</i>	Narrator	Narrator	Narrator has Grandson Narrator
<i>MawlaOf</i>	Narrator	Narrator	Narrator is Mawla of Narrator
<i>hasMawla</i>	Narrator	Narrator	Narrator has Mawla Narrator

Table 4.3 illustrates the data properties structure of IJS ontology.

Table 4.3: Hadith Ontology Data Properties

Data properties	Domain	Range	Description
<i>Id</i>	Person	integer	The identifier of the person
<i>Name</i>	Person	Literal	The Name of the person
<i>FirstName</i>	Person	Literal	The First Name of the person
<i>FatherName</i>	Person	Literal	The Father Name of the person
<i>FatherNickName</i>	Person	Literal	The Father Nick Name of the person
<i>GrandFatherName</i>	Person	Literal	The Grandfather Name of the person
<i>GrandFatherNickName</i>	Person	Literal	The Grandfather Nick Name of the person
<i>FourthName</i>	Person	Literal	The Fourth Name of the person
<i>FourthNickName</i>	Person	Literal	The Fourth Nick Name of the person
<i>FifthName</i>	Person	Literal	The Fifth Name of the person
<i>FifthNickName</i>	Person	Literal	The Fifth Nick Name of the person
<i>Konia</i>	Narrator	Literal	The Konia of the Narrator
<i>Luqab</i>	Narrator	Literal	The Luqab of the Narrator
<i>Nasab</i>	Person	Literal	The Nasab of the Narrator
<i>BirthYear</i>	Person	integer	The Birth Year of the Person
<i>DeathYear</i>	Person	integer	The Death Year of the Person
<i>Gender</i>	Person	Literal	The gender of the person
<i>LiveIn</i>	Person	Literal	The Country of the person
<i>OriginallyFrom</i>	Person	Literal	The Country of the person
<i>DenotedBy</i>	Person	Literal	The Nick name of the person
<i>Rotba</i>	Narrator	integer	The Rotba of the Narrator
<i>Tabaqa</i>	Narrator	integer	The Tabaqa of the Narrator
<i>Tadlees</i>	Narrator	integer	The Tadlees of the Narrator
<i>AttributedToGrandfather</i>	Person	Literal	Represents if Narrator Attributed to Grandfather
<i>hasBedaa</i>	Narrator	Integer	Represents if Narrator has Bedaa
<i>hasIkthelat</i>	Narrator	Integer	Represents if Narrator has Ikthelat
<i>hasIrsal</i>	Narrator	Integer	Represents if Narrator has Irsal
<i>hasWahem</i>	Narrator	Integer	Represents if Narrator has Wahem
<i>Content</i>	Hadith	Literal	The Hadith content (Matn)
<i>NoOfPages</i>	Book	Integer	Represents the number of pages in a book
<i>Title</i>	Hadith	Literal	The resource title e.g. book, chapter and Hadith titles
<i>Topic</i>	Hadith	Literal	Represents the Hadith topic
<i>YearOfPublication</i>	Book	Date	The year in the book was published

Step 6: Define the Facets of the Slots

Slots have different facets that describe the value type, allowed values, the number of the values (cardinality), and other features of the values the slot can take. In our case most of the slot values are string and integer either using ASCII or UTF-8 (Arabic). For example, the value type of *Name* Property is string and value type of *Tabaqa* is integer.

Step 7: Create IJS Instances

The last step is creating instances (individuals) of classes in the hierarchy. The creation of individuals allows for all the properties of the classes to record. The created individuals in IJS ontology represent the chosen sample of Hadiths, which used in the evaluation of system. These individuals include books, Hadiths, and narrators as shown in Figure 4.3.

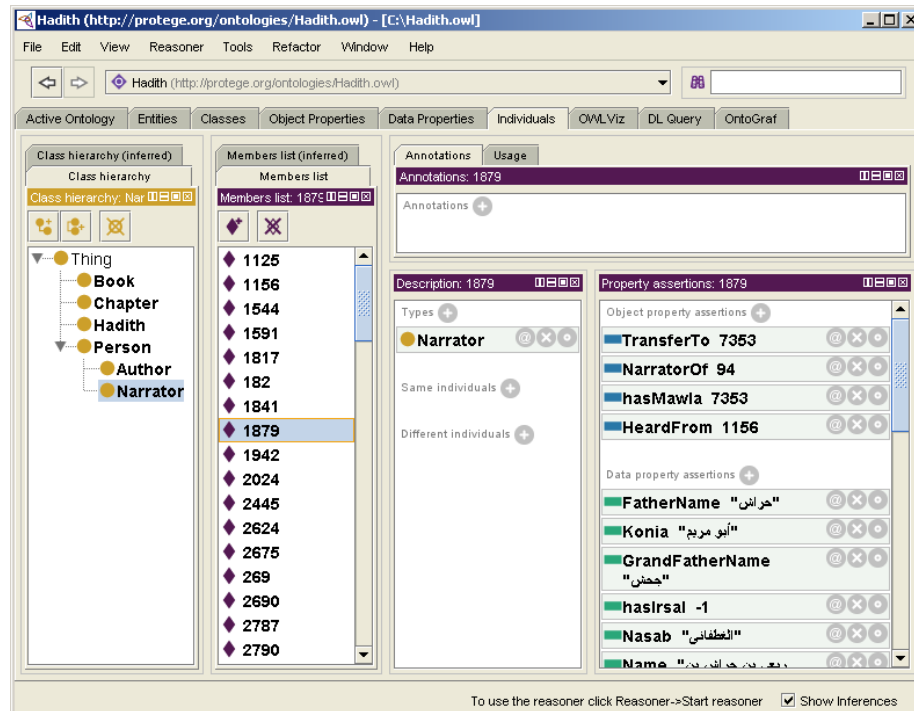


Figure 4.3: IJS Ontology Individual

Figure 4.3 shows the individuals tab in Protégé ontology editor. This tab contains class hierarchy, members list, object property, and data property.

The information of Individuals' taken from Ibn Hajar book [51] as we explained earlier in Section 1.5 (Scope and Limitations of the Research). Ibn Hajar identifies each narrator by a unique identifier and we use this identifier to identify the narrators' in members list.

Ibn Hajar identifies the teachers and students of each narrator. In addition, he identifies the family relation between narrators. We use these relations in object property such as *HeardFrom*, *TransferTo*, and *hasFather* properties.

Ibn Hajar identifies the *Rotba* of narrators by numeric list from one to twelve. In addition, he identifies the *Tabaqa* of narrators by numeric list from one to twelve, and he identifies the *Tadlees* of narrators by numeric list from one to five. We use these values when inserting narrators' data property.

Ibn Hajar also noted that a narrator might have *Ikhtilat*, *Irsal*, *Bida'a*, and *Wahem*. To represent these in ontology as follows: we put value -1 if the narrator do not has this data property *hasBedaa*, *hasIkthelat*, *hasIrsal* and *hasWahem* and value 1 if the narrator has this property. Then we refer to the Hadith that achieves this property using *BedaaHadiths*, *WahemHadiths*, and *IkthelatHadiths* object property.

4.2 Evaluating the IJS Ontology

In this section, we evaluate the quality of the created ontology in representing all terms, properties, and relations by a Hadith example and ontology querying. The Task-Based evaluation methodology, which explained in Section 3.2.3 (Ontology Evaluation), will be used later in Chapter 7 (Results and Evaluation) because the Task-Based evaluation framework used with application and ontology not with ontology alone.

4.2.1 Quality through Hadith Number Six Example

To evaluate the quality of the IJS ontology we choose Hadith example to show if the ontology represent terms, properties and relations of Hadith sample. The chosen Hadith is Hadith Number six in Sunnan Ibn Majah book:

6- حَدَّثَنَا مُحَمَّدُ بْنُ بَشَّارٍ حَدَّثَنَا مُحَمَّدُ بْنُ جَعْفَرٍ حَدَّثَنَا شُعْبَةُ عَنْ مُعَاوِيَةَ بْنِ قُرَّةَ عَنْ أَبِيهِ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ لَا تَزَالُ طَائِفَةٌ مِنْ أُمَّتِي مُنْصُورِينَ لَا يُضْرُّهُمْ مَنْ خَذَلَهُمْ حَتَّى تَقُومَ السَّاعَةُ.

It was narrated that Muhammad Ibn Bashir Tell us Muhammad Ibn Jaafar Tell us Shuba Tell us Mu'awiyah Ibn Qurrah narrated that his father said: The Messenger of Allah (ﷺ) said: "A group of my Ummah will continue to prevail and they will never be harmed by those who forsake them, until the Hour begins."

When we analyze the Isnad of the Hadith, we find that it contains the narrators that listed in the Table 4.4.

Table 4.4: Narrator of Hadith Number Six

No.	Narrator Name	ID
1	Muhammad Ibn Bashar “محمد بن بشار”	5754
2	Muhammad Ibn Jaafar “محمد بن جعفر”	5787
3	Shuba Ibn Al Hajaj “شعبة بن الحجاج”	2790
4	Mu'awiyah Ibn Qurrah “معاوية بن قره”	6769
5	Qurrah Ibn Iyas “قره بن إياس”	5537

Figure 4.4 shows the extracted terms, properties, and relations of Hadith that represented in the ontology.

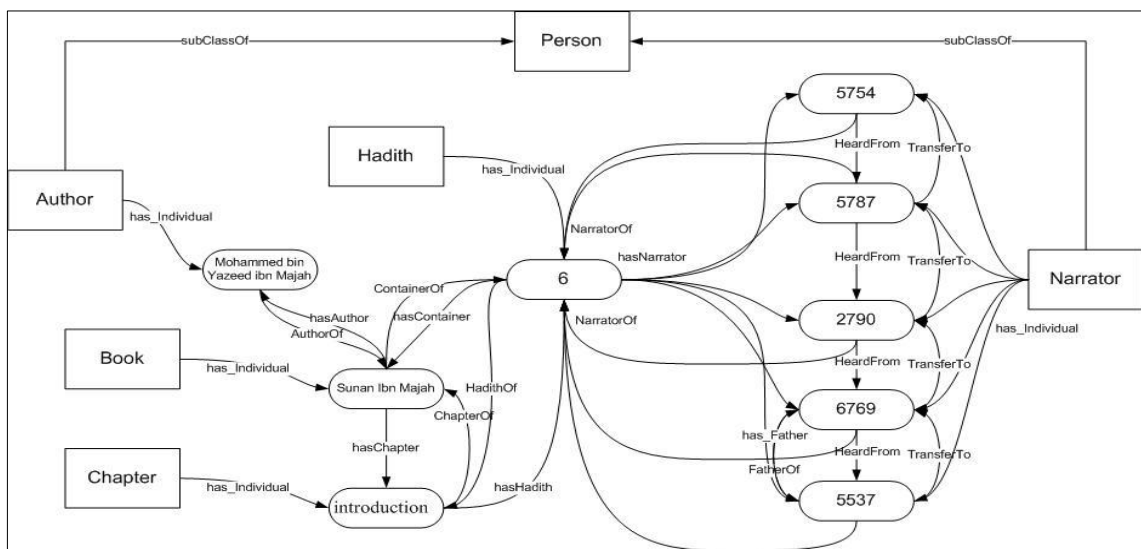


Figure 4.4: Ontology of Hadith Number 6

Figure 4.5 shows the terms, properties, and relations of Hadith that represented in the ontology using in protégée.

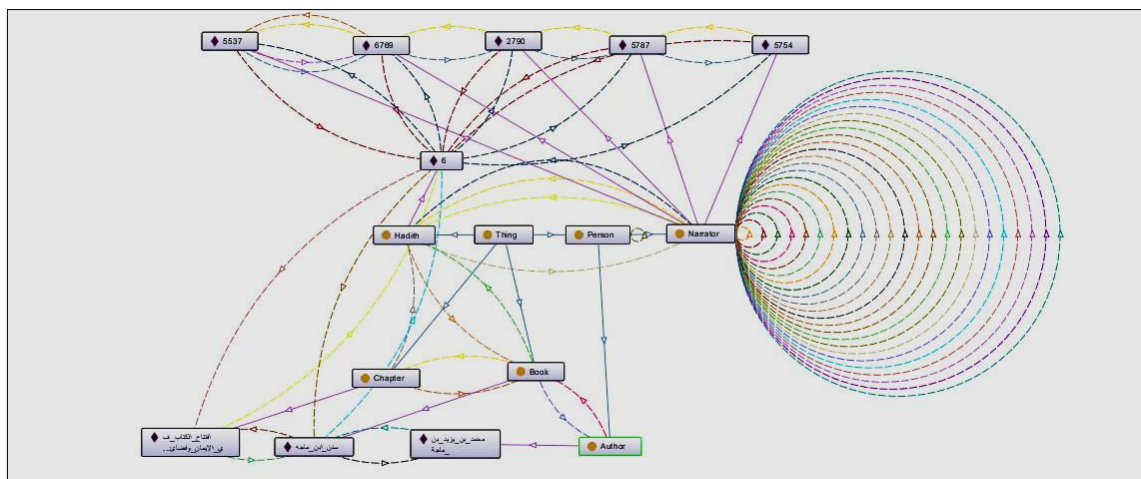


Figure 4.5 Ontology of Hadith Number 6 in Protégée

It shows all class of IJS ontology: *Book, Chapter, Hadith, Person, Author, and Narrator*. *Author and Narrator* are sub classes of *Person*.

Based on the figure, the relation and properties are explained as follow:

Book Class has individual: Sunnan Ibn Majah, Sunnan Ibn Majah book has Author (*hasAuthor* Property): Mohammed Ibn Yazed Ibn Majah, the book contain the chapter (*hasChapter* Property): Introduction and the book is contains the Hadith 6 (*ContainerOf* Property).

Chapter Class has individual: Introduction, the chapter is chapter of (*ChapterOf* Property): Sunnan Ibn Majah and the chapter is contain the Hadiths 6 (*hasHadith*).

Hadith Class has individual: Hadith number six, it found in chapter one of the Sunnan Ibn Majah book (*HadithOf* Property) which is introduction, the Hadith has many narrators (*hasNarrator* Property): 5754, 5787, 2790, 6769, and 5537.

Narrator Class has individuals: 5754, 5787, 2790, 6769, and 5537. These numbers represent the identifier of narrator identified by Ibn Hajar scholar as we explained earlier in Step 7 (Create IJS Instances). Each Narrator has heard from the next narrator (*HeardFrom* Property) and each narrator transfers to the previous narrator (*TransferTo* Property). All these narrators narrated the (*NarratorOf* Property) *Hadith 6*.

All properties that are needed in judging the Isnad of the Hadith are represented by the ontology, which are *Tadlees, Ikhtilat, Irsal, Bida'a, and Wahem* as shown in Figure 4.8, which illustrates these properties in OWL.

All narrators' properties that are needed to identify the narrators are represented by the ontology which is: *Id, Name, FirstName, FatherName, FatherNickName, GrandFatherName, GrandFatherNickName, FourthName, FourthNickName, FifthName, FifthNickName, hasFather, Nasab, BirthYear, DeathYear, Gender, LiveIn, OriginallyFrom, DenotedBy, Konia, Luqab, Rotba, Tabaqa, Tadlees, hasBedaa, hasIkthelat, hasIrsaland* and *hasWahem* as shown in Figure 4.6 which illustrates these properties are presented in Protégé ontology editor.

The example shows that all elements of Hadith Isnad judgment are set as classes. All properties of narrator are set as object and data properties. Slots were the various attributes of Hadith, also are set.

4.2.2 Quality through Ontology Querying

In order to verify and validate the ontology as regards to competency questions, we use the Description Logic Query (DL-Query) that is a standard Protégé 4 plugin and it based on the Manchester OWL syntax.

Protégé allows for querying the ontology. This allows the ontology to access easily and narrators to be search via their properties. This query function is an example of how IJS system could use the IJS ontology.

Three examples of the querying function are listed below. They answer the main questions that are asked in the development process of the ontology.

Example 1:

- The question: Who are the teachers and students of *Narrator* with Id = 2445?
- DL-Query: *Narrator* and HeardFrom 2445 or TransferTo 2445.
- The result of DL-Query is shown in Figure 4.5 which illustrates the individuals of *Narrator* class that are teachers and students of *Narrator* with Id = 2445.

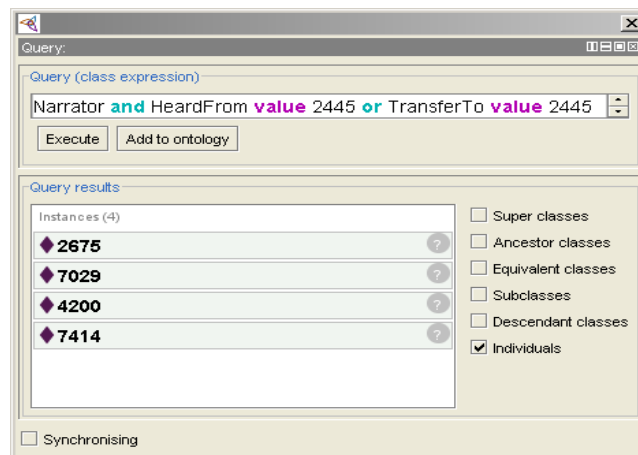


Figure 4.5: Query for Teachers and Student of Narrator

Example 2:

- The question: who is the Narrator that has Konia Abu Al Azhar “أبو الأزهر”?
- DL-Query: *Narrator* and Konia value “أبو الأزهر”.
- The result of DL-Query shown in Figure 4.6, which illustrates the individuals of *Narrator* class that is has Konia “أبو الأزهر”.

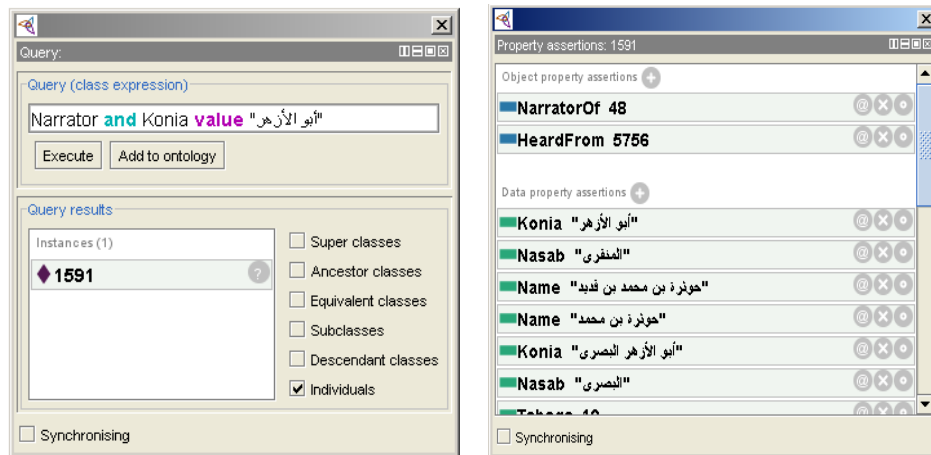


Figure 4.6: Query the Narrator by Konia.

Example 3:

- The question: who are the narrators of Hadith number six?
- DL-Query: Narrator and NarratorOf value 6.
- The results of DL-Query shown in Figure 4.7, which illustrates the individuals of *Narrator* class, which are narrator of Hadith, number six.

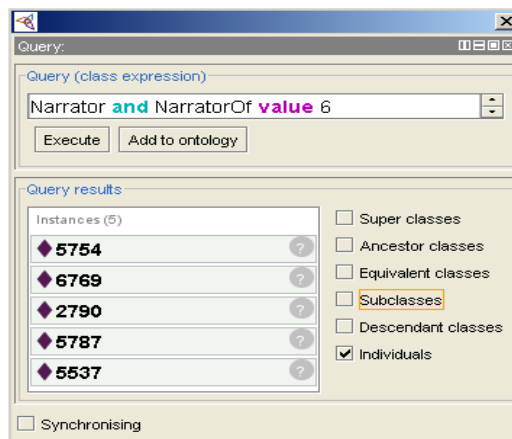


Figure 4.7: Query the Narrators of Hadith Number 6.

The results of DL-Query example show that the ontology are successfully portrays the body of knowledge related to Isnad judgment. Querying show, that IJS ontology provides an avenue for searching and manipulating the data within the ontology.

4.3 Summary

In this chapter, we have explained the development and evaluation of the IJS ontology. In the first section, we used the steps that we had explained in Section 3.2.2 (Ontology Development) in the process of building ontology. At the beginning, we identify domain and scope of the ontology.

Then we use the HadithRDF as bases for building the IJS ontology, where we add new properties and relation to achieve the goal of the research. We define the term and the properties which are listed in the Tables 4.1, 4.2, 4.3. In addition, we adding individuals to IJS ontology using Protégé and explained some of the factories that relate to the values of some properties, such as the *Rotba* and *Tabaqa*, narrators *Id* depending on the Ibn Hajar book.

In the second section, we have explained how to evaluate the IJS ontology and explain why Task-based cannot apply now, where it needs a full system to apply. We restricted ourselves in this section in two evaluating methods:

The first method is Hadith example to extract all elements of the IJS ontology and to make sure that all the elements are represented in the ontology. We proved that ontology had represented all the elements of the Hadith.

The second method is the queries from the IJS ontology to ensure that it answer the questions that are needed in the process of identifying narrators and returns their information. We proved that the ontology have answered all questions and return the correct results.

The results of the evaluation in both methods show that the development of the IJS ontology has achieved the purpose of its presence in the IJS system.

Chapter 5

Algorithm for Isnad Judgment

In this chapter, we present an algorithm that portrays the use of the IJS the ontology in the Isnad judgment process. The role of e-Narrator Web Service in this algorithm and the rules of Isnad judgment is also presented.

The process of Isnad judgment has four steps, which are separate the Isnad from Hadith, identify the narrator using the IJS ontology, check the continuity of Isnad, and applying the judgment rules. Each step has some inputs, generally taken from an earlier step, and produce outputs that fed into the next step. Each step will be explained in a separate section.

5.1 Separating the Isnad from Hadith (Matn)

In this section, we present the step of separating the Isnad from the Matn of Hadith (separating the narrators' chain from the words of Hadith). To do that we convert the e-Narrator [29] desktop application to a Web Service. This Web Service takes the Hadith as input and returns the narrators' chain and narration tool as output. We use it because it draws the Isnad tree, which is a basis of the Isnad judgment.

5.1.1 e-Narrator Overview

e-Narrator [29] focus on automatic generation and graphical representation of the Hadith narration tree. This process involves creating natural language lexer, performing shallow parsing, parsing, building a syntactic analyzer, and then applying semantic processing to identify narrators' names. Then generate an RDF representation of the narration chain based on HadithRDF ontology and finally a graph presenter that displays the narrators' chain graphically [29].

The e-Narrator system is divided into several small components. Each component has some inputs, generally taken from an earlier component, and produce outputs that fed into the next component in the line. Figure 4.1 shows the abstract level of the e-Narrator system components [29]. We will explain briefly each component.

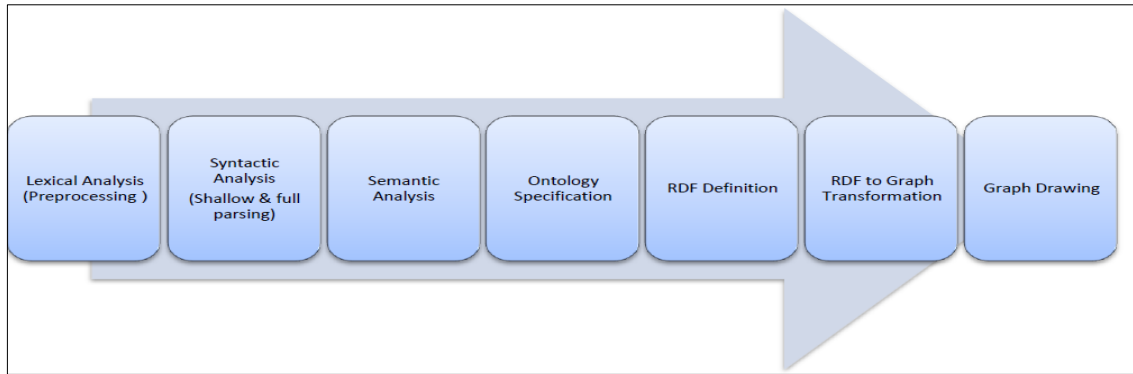


Figure 5.1: Abstract Level of the e-Narrator System Components [29]

- **Lexical Analysis:** This process is considered a preprocessing phase for the parser. The preprocessing phase includes removing punctuations, removing the diacritical markings, Join lines (remove new lines), remove extra white spaces, and replace the single white space that separates words by the symbol “␣”.
- **Syntactic Analysis:** This process groups and tags the sets of words into categories to use by shallow parsing. The shallow parser produces and categorizes the sentences into a series of words that have something in common. In addition, it contains accurate grammar that is used to parse the Hadith text both Isnad and Matn.
- **Semantic Analysis:** The main task is to identify the narrators' names and the connection between those names.
- **RDF Definition:** In this phase, the HadithRDF ontology is used. Building the system that leverages the previous phases that are concerned with semantically parsing and annotating the Hadith contents.
- **RDF Transformation:** In this phase, query RDF and then transform the XML RDF into a graph representing the narration flow diagram.
- **Graph Drawing:** Rendering the chain of Hadith transmitters graphically as nodes.

5.1.2 Creating the e-Narrator Web Service

For ethical issues we use the e-Narrator application as black box so the best method to do that adding interface to converting it to Web Service that gets Hadith text and return narrators list, Figure 5.2 shows the WSDL file of the created e-Narrator Web Service.

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://Hadith.ps/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://Hadith.ps/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schemaelementFormDefault="qualified" targetNamespace="http://Hadith.ps/">
      <s:element name="GetEsnad">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="HadithContent" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetEsnadResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetEsnadResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="GetEsnadSoapIn">
    <wsdl:part name="parameters" element="tns:GetEsnad" />
  </wsdl:message>
  <wsdl:message name="GetEsnadSoapOut">
    <wsdl:part name="parameters" element="tns:GetEsnadResponse" />
  </wsdl:message>
  <wsdl:portType name="ServiceSoap">
    <wsdl:operation name="GetEsnad">
      <wsdl:input message="tns:GetEsnadSoapIn" />
      <wsdl:output message="tns:GetEsnadSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ServiceSoap" type="tns:ServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetEsnad">
      <soap:operation soapAction="http://Hadith.ps/GetEsnad" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="ServiceSoap12" type="tns:ServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetEsnad">
      <soap12:operation soapAction="http://Hadith.ps/GetEsnad" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="Service">
    <wsdl:port name="ServiceSoap" binding="tns:ServiceSoap">
      <soap:address location="http://localhost/service.asmx" />
    </wsdl:port>
    <wsdl:port name="ServiceSoap12" binding="tns:ServiceSoap12">
      <soap12:address location="http://localhost/service.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

Figure 5.2: WSDL of e-Narrator Web Service

5.1.3 Testing e-Narrator Web Service

We will test the e-Narrator Web Services by calling web methods from unit tests, see Section 6.4.2: (Testing the e-Narrator Web Service) for more details. The Web Service run through the web browser by insert the Hadith number six as input then the output is the narrators' chain with narration tools as shown in table 5.1.

Table 5.1: Input and Output of the Web Service

Input:
<p>6- حَدَّثَنَا مُحَمَّدُ بْنُ بَشَّارٍ حَدَّثَنَا مُحَمَّدُ بْنُ جَعْفَرٍ حَدَّثَنَا شُعْبَةُ عَنْ مُعَاوِيَةَ بْنِ قُرَّةَ عَنْ أَبِيهِ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ لَا تَزَالُ طَائِفَةٌ مِنْ أُمَّتِي مَنْصُورِينَ لَا يَضُرُّهُمْ مَنْ خَذَلَهُمْ حَتَّى تَقُومَ السَّاعَةُ.</p> <p>It was narrated that Muhammad Ibn Bashar Tell us Muhammad Ibn Jaafar Tell us Shuba Tell us Mu'awiyah Ibn Qurrah narrated that his father said: The Messenger of Allah () said: "A group of my Ummah will continue to prevail and they will never be harmed by those who forsake them, until the Hour begins."</p>
Output:
<pre>[[{"name": "الاسناد", "tool": ""}, {"name": "محمد بن بشار", "tool": "حدثنا"}, {"name": "محمد بن أبيه-", "tool": "عن"}, {"name": "معاوية بن قره", "tool": "حدثنا"}, {"name": "شعبة", "tool": "حدثنا"}, {"name": "جعفر", "tool": "حدثنا"}, {"name": "قره", "tool": "عن"}]]</pre>

The e-Narrator system has limitation that it was design according to Sahih Muslim book and can't deal with all Isnad in another six books of Hadith that are represent scope of research as explained in Section 1.5 (Scope and Limitations of the Research), so we will not get the output from the e-Narrator Web Service for all inputs.

5.2 Identifying the Narrator Using the IJS Ontology

The narrator Identification usually pose a challenge and takes a long time from the researcher to determine and Identify that the narrator, may cannot each researcher due two reasons[52]:

1. **The various and different names manner:** the name of narrator that written in the Hadiths contains many manners. Table 5.2 shows the different manners of narrators' names as they are listed in Hadith.

Table 5.2: Name Manners of Narrators

Name Manners	Example	
First Name	Safina	سَفِينَةَ
First Name + Father	Ismail Ibn Ibrahim	إِسْمَاعِيلُ بْنُ إِبْرَاهِيمَ
First name + Father + Grandfather	Bakr Ibn Yahya Ibn Zaban	بَكْرُ بْنُ يَحْيَى بْنِ زَبَانَ
First Name + Father + Family	Abdul Rahman Ibn Ibrahim Damashcky	عَبْدُ الرَّحْمَنِ بْنُ إِبْرَاهِيمَ الدَّمَشَقِيُّ
First Name + Family	Khallad al-Saffar	خَلَّادُ الصَّفَّارِ
First Name + Father nickname	Yazed Ibn Abu Zaid	يَزِيدُ بْنُ أَبِي زَيْدٍ
First Name + Father nickname + Family	Said Ibn Abi Said Maqbari	سَعِيدُ بْنُ أَبِي سَعِيدِ الْمُقَبَّرِيِّ
First name + Father + Grandfather + Fourth Grandfather Nickname	Abdullah Ibn Mohammed Ibn Aqeel Ibn Abi Talib	عَبْدُ اللَّهِ بْنُ مُحَمَّدِ بْنِ عَقِيلِ بْنِ أَبِي طَالِبٍ
Surname	Abu Zubair	أَبُو الزُّبَيْرِ
Surname + Father	Umm Habiba bint Jahsh	أُمُّ حَبِيبَةَ بِنْتِ جَحْشٍ
Surname + Family	Abu Malik al-Ash'ari	أَبِي مَالِكِ الْأَشْعَرِيِّ
Surname + Father Surname	Abu Bakr Ibn Abi Shaybah	أَبُو بَكْرٍ بْنُ أَبِي شَيْبَةَ
Ibn + Father Name	Ibn Umar	ابْنِ عُمَرَ
Ibn + Father Nickname	Ibn Abi Mary	ابْنُ أَبِي مَرْيَمَ
Family Relation	His father, his grandfather, his master, his uncle, his brother, my father... etc.	أَبِيهِ، جَدِّهِ، مَوْلَاهُ، عَمِّهِ، أَخِيهِ، أَبِي ... الخ
Luqab	Al Amash, Nubayshah al Khair	الْأَعْمَشِيُّ، نُبَيْشَةُ الْخَيْرِ
Family	Al Ouzaii	الْأَوْزَاعِيُّ

These various name of the narrators that are listed in Hadith constituted a major challenge to the Hadith scholars to identify the narrators. To address this problem we set some rules to identify the narrator by using the ontology in Section 5.2.1 (Narrator Identification Rules).

2. **Similarity of names:** it pose a challenge in identifying the narrators such as if the narrator is heard from the two narrators Whose names are similar, for example the Hadith text contains the narrator name Sufyan “سُفْيَانُ” and we don't know if he Sufyan Ibn Oaina “سُفْيَانُ بْنُ عَيْنَةَ” or Sufyan Al Thawry "سُفْيَانُ الثَّوْرِيُّ".

5.2.1 Narrator Identification Rules

To solve the manners and similarity of names, we set some rules to identify the narrator using the ontology:

1. Insert all cases of the narrator manners in the *Name*, *Luqab*, *Konia*, and *DenotedBy* properties. This will solve the complexity of narrators' name.
2. The ontology querying executed in *Name*, *Konia*, *Luqab*, *Nasab*, *DenotedBy* properties.
3. Identify all narrator of Hadith by ontology querying of Hadith narrator.
4. Identify the first narrator in the chain using ontology querying of book author.
5. Identify the second narrator using the ontology querying for the teacher of the first narrator and so forth. Then query the *TransferTo* property of the first narrator.
6. Identify the narrator who has family relations using ontology querying for it relation such as *FatherOf* property.

5.2.2 Hadith Number Six Example (continued)

To show how the rules of identifying the narrators solve the manners and similarity of names, we continue the Hadith number six example that is introduced in Section 4.2.1 (Quality through Hadith Number Six Example). In Hadith number six example, we show that the extracted terms, properties, and relations of Hadith are represented in the ontology.

The First Step is to separate the Isnad from Hadith using e-Narrator Web Service. The output of the Web Service is the narrators' chain with narration tools as shown in Table 5.1. The narration tools are used in the judging algorithm (see Section 5.4 (Applying the Judgment Rules))

The Second Step identifying narrators in the chain and getting all information about them. In this step, we apply our rules to identify the narrator. See Section 5.2.1 (Narrator Identification Rules) as follows:

1. Querying the ontology about all narrators of Hadith number six is based on rule no.3 that will narrow the search for narrators instead of searching thousands of narrators therefor searching in the narrators of this Hadith only.

2. Querying the ontology about each narrator in the chain to identify the narrator and gets all information about him based on rule No.4 and rule No.5.
- **First Narrator** in the chain is Muhammad Ibn Bashar “محمد بن بشار”. We query narrator in *Name* property. The query executed on the Sunnan Ibn Majah book Author. Table 5.3 shows the result of the query and all returned information of the narrator.

Table 5.3: Result of Query about First Narrator of Hadith

Query:
Narrator and NarratorOf value 6 and Name value “محمد بن بشار” and TransferTo value “Mohammed Ibn Yazed Ibn Majah”
Results:
<pre> <!-- http://protege.org/ontologies/Hadith.owl#5754 --> <owl:NamedIndividual rdf:about="&ontologies;Hadith.owl#5754"> <rdf:type rdf:resource="&ontologies;Hadith.owl#Narrator"/> <hasIrsal rdf:datatype="&xsd;integer">-1</hasIrsal> <hasBedaa rdf:datatype="&xsd;integer">-1</hasBedaa> <Tadlees rdf:datatype="&xsd;integer">-1</Tadlees> <hasWahem rdf:datatype="&xsd;integer">-1</hasWahem> <hasIkthelat rdf:datatype="&xsd;integer">-1</hasIkthelat> <Tabaqa rdf:datatype="&xsd;integer">10</Tabaqa> <BirthYear rdf:datatype="&xsd;integer">167</BirthYear> <DeathYear rdf:datatype="&xsd;integer">252</DeathYear> <Rotba rdf:datatype="&xsd;integer">3</Rotba> <Id rdf:datatype="&xsd;integer">5754</Id> <Nasab>العبدى</Nasab> <Name>محمد بن بشار</Name> <Konia>أبو بكر البصرى</Konia> <GrandFatherName>عثمان</GrandFatherName> <Konia>أبو بكر</Konia> <Luqab>بندار</Luqab> <Nasab>البصرى</Nasab> <FirstName>محمد</FirstName> <Name>محمد بن بشار بن عثمان</Name> <FatherName>بشار</FatherName> <HeardFrom rdf:resource="&ontologies;Hadith.owl#5787"/> <NarratorOf rdf:resource="&ontologies;Hadith.owl#6"/> </owl:NamedIndividual> </pre>

- **Second Narrator** in the chain is Muhammad Ibn Jaafar “محمد بن جعفر”. The query is executed on the teachers of the first narrator. Table 5.4 shows the result of the query and all information of the narrator.

Table 5.4: Result of Query about Second Narrator of Hadith

Query:
Narrator and NarratorOf value 6 and Name value "محمد بن جعفر" and TransferTo value 5754
Results:
<pre> <!-- http://protege.org/ontologies/Hadith.owl#5787 --> <owl:NamedIndividual rdf:about="&ontologies;Hadith.owl#5787"> <rdf:type rdf:resource="&ontologies;Hadith.owl#Narrator"/> <hasIrsal rdf:datatype="&xsd;integer">-1</hasIrsal> <hasBadaa rdf:datatype="&xsd;integer">-1</hasBadaa> <hasWahem rdf:datatype="&xsd;integer">-1</hasWahem> <Tadlees rdf:datatype="&xsd;integer">-1</Tadlees> <hasIkthelat rdf:datatype="&xsd;integer">-1</hasIkthelat> <DeathYear rdf:datatype="&xsd;integer">294</DeathYear> <Rotba rdf:datatype="&xsd;integer">3</Rotba> <Id rdf:datatype="&xsd;integer">5787</Id> <Tabaqa rdf:datatype="&xsd;integer">9</Tabaqa> <Konia>أبو عبد الله</Konia> <FirstName>محمد</FirstName> <Nasab>البحري</Nasab> <DenotedBy>صاحب الكرابيين</DenotedBy> <Nasab>الهذلي</Nasab> <DenotedBy>غندر</DenotedBy> <FatherName>جعفر</FatherName> <Name>محمد بن جعفر</Name> <HeardFrom rdf:resource="&ontologies;Hadith.owl#2790"/> <TransferTo rdf:resource="&ontologies;Hadith.owl#5754"/> <NarratorOf rdf:resource="&ontologies;Hadith.owl#6"/> </owl:NamedIndividual> </pre>

- **Third Narrator** in the chain is Shuba "شعبة". The query is also executed on the teachers of the second narrator. Table 5.5 shows the result of the query and all information of the narrator.

Table 5.5: Result of Query about Third Narrator of Hadith

Query:
Narrator and NarratorOf value 6 and Name value "شعبة" and TransferTo value 5785
Results:
<pre> <owl:NamedIndividual rdf:about="&ontologies;Hadith.owl#2790"> <rdf:type rdf:resource="&ontologies;Hadith.owl#Narrator"/> <Tadlees rdf:datatype="&xsd;integer">-1</Tadlees> <hasIrsal rdf:datatype="&xsd;integer">-1</hasIrsal> <hasBadaa rdf:datatype="&xsd;integer">-1</hasBadaa> <hasWahem rdf:datatype="&xsd;integer">-1</hasWahem> <hasIkthelat rdf:datatype="&xsd;integer">-1</hasIkthelat> <DeathYear rdf:datatype="&xsd;integer">160</DeathYear> <Rotba rdf:datatype="&xsd;integer">2</Rotba> <Id rdf:datatype="&xsd;integer">2790</Id> <Tabaqa rdf:datatype="&xsd;integer">7</Tabaqa> <Name>شعبة بن الحجاج بن الورد</Name> <GrandFatherName>الورد</GrandFatherName> <FatherName>الحجاج</FatherName> <FirstName>شعبة</FirstName> <Konia>أبو بسطام</Konia> <Nasab>الأزدى</Nasab> <Nasab>الواسطي</Nasab> <Nasab>البحري</Nasab> <Nasab>العنكي</Nasab> <Name>شعبة بن الحجاج</Name> <Name>شعبة</Name> <TransferTo rdf:resource="&ontologies;Hadith.owl#5787"/> <NarratorOf rdf:resource="&ontologies;Hadith.owl#6"/> <HeardFrom rdf:resource="&ontologies;Hadith.owl#6769"/> </owl:NamedIndividual> </pre>

- **Fourth Narrator** in the chain is Mu'awiyah Ibn Qurrah "معاوية بن قرة". The query is also executed on the teachers of the third narrator. Table 5.6 shows the result of the query and all information of narrator.

Table 5.6: Result of Query about Fourth Narrator of Hadith

Query:
Narrator and NarratorOf value 6 and Name value "معاوية بن قرة" and TransferTo value 2790
Results:
<pre> <owl:NamedIndividual rdf:about="&ontologies;Hadith.owl#6769"> <rdf:type rdf:resource="&ontologies;Hadith.owl#Narrator"/> <Tadlees rdf:datatype="&xsd;integer">-1</Tadlees> <hasIrsal rdf:datatype="&xsd;integer">-1</hasIrsal> <hasBedaa rdf:datatype="&xsd;integer">-1</hasBedaa> <hasIktelat rdf:datatype="&xsd;integer">-1</hasIktelat> <hasWahem rdf:datatype="&xsd;integer">-1</hasWahem> <DeathYear rdf:datatype="&xsd;integer">113</DeathYear> <Rotba rdf:datatype="&xsd;integer">3</Rotba> <Tabaqa rdf:datatype="&xsd;integer">3</Tabaqa> <BirthYear rdf:datatype="&xsd;integer">36</BirthYear> <Id rdf:datatype="&xsd;integer">6769</Id> <Name>معاوية بن قرة بن اياس</Name> <Name>معاوية بن قرة بن هلال</Name> <FatherName>معاوية</FatherName> <FourthName>هلال</FourthName> <Nasab>المزني</Nasab> <GrandFatherName>اياس</GrandFatherName> <Nasab>البصري</Nasab> <Name>معاوية بن قرة</Name> <Name>معاوية بن قرة بن اياس بن هلال بن رثاب</Name> <Konia>ابو اياس البصري</Konia> <FirstName>معاوية</FirstName> <Konia>ابو اياس</Konia> <TransferTo rdf:resource="&ontologies;Hadith.owl#2790"/> <HeardFrom rdf:resource="&ontologies;Hadith.owl#5537"/> <hasFather rdf:resource="&ontologies;Hadith.owl#5537"/> <NarratorOf rdf:resource="&ontologies;Hadith.owl#6"/> </owl:NamedIndividual> </pre>

- **Fifth narrator** in the chain is father of Mu'awiyah, which is Qurrah "أبيه-قرة". The query is executed on *FatherOf* property of third narrator based on rule no.6. Table 5.7 shows the result of the query and all information of narrator.

Table 5.7: Result of Query about Fifth Narrator of Hadith

Query:
Narrator and NarratorOf value 6 and Name value "قرة" and TransferTo value 6769 and FatherOf value 6769
Results:
<pre> <!-- http://protege.org/ontologies/Hadith.owl#5537 --> <owl:NamedIndividual rdf:about="&ontologies;Hadith.owl#5537"> <rdf:type rdf:resource="&ontologies;Hadith.owl#Narrator"/> <hasWahem rdf:datatype="&xsd;integer">-1</hasWahem> <hasIrsal rdf:datatype="&xsd;integer">-1</hasIrsal> <Tadlees rdf:datatype="&xsd;integer">-1</Tadlees> <hasBedaa rdf:datatype="&xsd;integer">-1</hasBedaa> <hasIkthelat rdf:datatype="&xsd;integer">-1</hasIkthelat> <Rotba rdf:datatype="&xsd;integer">1</Rotba> <Tabaqa rdf:datatype="&xsd;integer">1</Tabaqa> <Id rdf:datatype="&xsd;integer">5537</Id> <DeathYear rdf:datatype="&xsd;integer">64</DeathYear> <Konia>أبو معاوية</Konia> <Name>قرة</Name> <FatherName>إيأس</FatherName> <FirstName>قرة</FirstName> <Nasab>المزني</Nasab> <Name>قرة بن إيأس بن هلال</Name> <GrandFatherName>هلال</GrandFatherName> <Name>قرة بن إيأس بن هلال بن رباب</Name> <Name>قرة بن إيأس</Name> <Konia>أبو معاوية البصري</Konia> <NarratorOf rdf:resource="&ontologies;Hadith.owl#6"/> <TransferTo rdf:resource="&ontologies;Hadith.owl#6769"/> <FatherOf rdf:resource="&ontologies;Hadith.owl#6769"/> </owl:NamedIndividual> </pre>

This example shows that we have identified narrators in the Isnad chain and have returned information of each of them using narrator identification rules. Next step is checking the continuity of Isnad to check that each narrator heard from its teacher and the teacher transfer the Hadith to its students by querying IJS ontology.

5.3 Check Continuity of Isnad

Continuity of Isnad means that:

1. Each narrator in the Isnad chain heard the Hadith from his teacher and the teacher had transferred the Hadith to his students.
2. Check if the narrator did not hear the chosen Hadith from his teacher or not. This case is satisfied when the Hadith scholar explicitly states that the narrator did not hear this Hadith from his teacher.

In our algorithm, we check the continuity of Isnad by means of the ontology reasoner. It checks whether each narrator heard from his teacher. The ontology objects properties: *HeardFrom*, *notHeardFrom*, and *TransferTo* are used for this purpose.

5.3.1 Hadith Number Six Example (continued)

We show how the continuity of Isnad can be checked based on Hadith Number Six Example. In Hadith Number Six Example we show that the extracted terms, properties and relations of Hadith are represented in the ontology, then we identified the five narrators of Hadith Number Six and returned information of each narrator using the IJS ontology.

To apply the continuity of Isnad, we use OWL-API reasoner to query the ontology about the properties: *HeardFrom*, *notHeardFrom* between each two narrators. The results of reasoning are illustrated in the Table 5.8.

Table 5.8: Results of Ontology Reasoning

No.	First Narrator	Second Narrator	Heard From	notHeard From
1	Muhammad Ibn Bashar	Muhammad Ibn Jaafar	Yes	No
2	Muhammad Ibn Jaafar	Shuba Ibn Alhajaj	Yes	No
3	Shuba Ibn Alhajaj	Mu'awiyah Ibn Qurrah	Yes	No
4	Mu'awiyah Ibn Qurrah	Qurrah Ibn Iyas	Yes	No

5.4 Applying the Judgment Rules

This section represents the rules that are in the HIJ algorithm. These rules are followed by Hadith scholars to judge the Hadith Isnad. See Table 3.1(Judging on Isnad).

These rules are looking at information narrator about:

1. Degree of Authenticity: we use the Rotba of Ibn Hajar to know the degree of authenticity of narrator; we represent it as (*Rotba* property) in ontology.
2. Condition of Trouble: that is looking at *Tadlees*, *Ikthelat*, *Irsal*, *Bedaa*, and *Wahem*. We represent it as (*hasIkthelat*, *ikthelatHadiths*, *hasIrsal*, *hasBedaa*, *bedaaHadiths*, *hasWahem*, and *wahemHadiths*) properties in ontology.

5.4.1 Hadith Isnad Judgment Algorithm (HIJ)

The HIJ algorithm represents the rules that are followed by Hadith scholars to judge the Hadith Isnad. It is used to produce the Hadith Isnad judgment based on IJS

ontology. The algorithm as a flowchart is illustrated in Figure 5.3 and is listed afterwards.

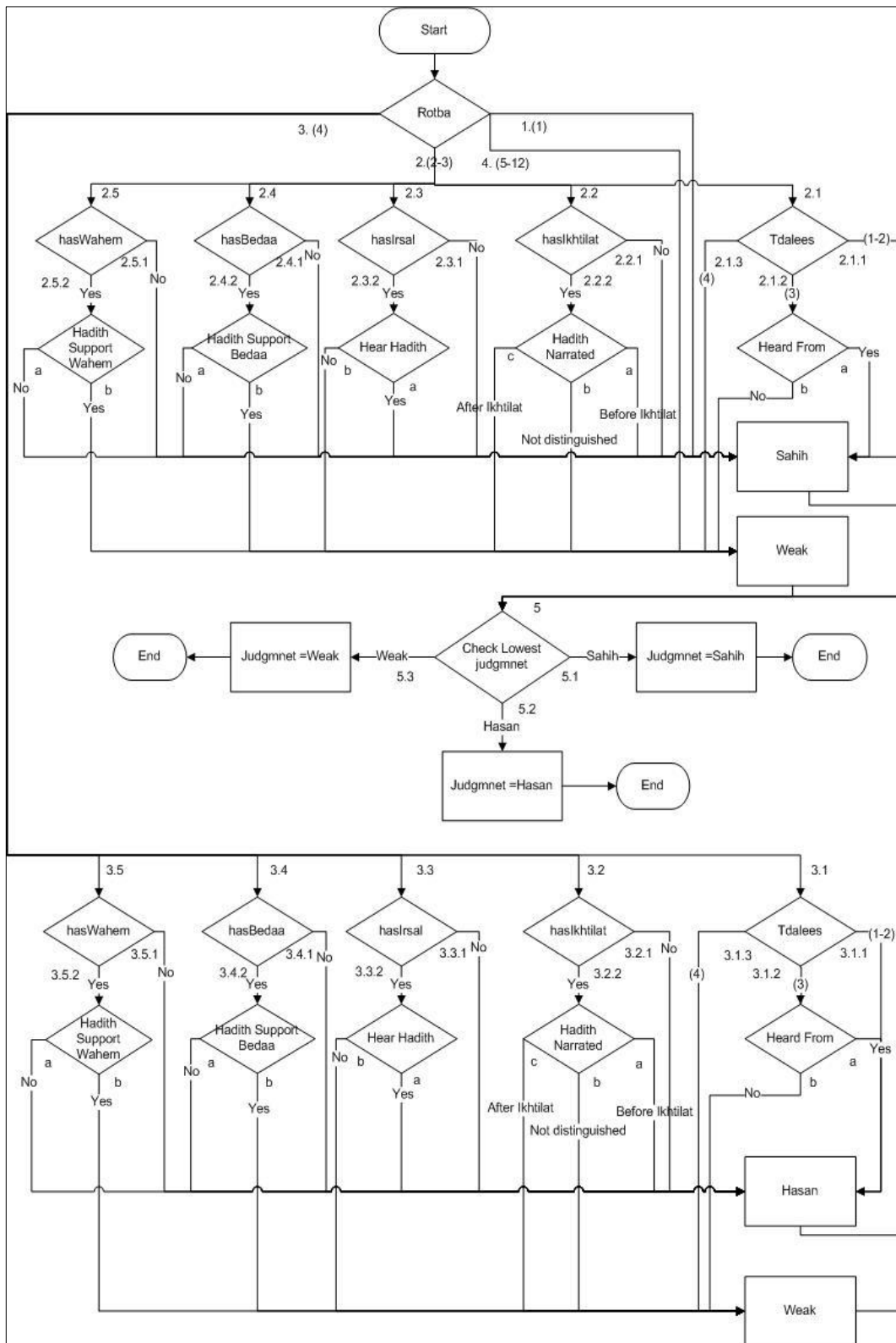


Figure 5.3: HIJ Algorithm Flow Chart

The HIJ algorithm checks the *Rotba* property (This property has values from one to twelve) as follows:

1. If the *Rotba* property value is one, then this narrator is a “Sahabi” and the Isnad takes the judgment (Sahih).
2. If the *Rotba* property value is 2-3, then the judgment will be taken from the following steps:
 - 2.1 Check the *Tadlees* property, this property has values 1-5.
 - 2.1.1 If the *Tadlees* property value is 1-2, then the Isnad will get judgment (Sahih).
 - 2.1.2 If the *Tadlees* property value is 3, then check that the narrator has *HeardFrom* the teacher narrator:
 - a. If *HeardFrom* = yes then the Isnad will get a judgment (Sahih).
 - b. If *HeardFrom* = no, then the Isnad will get a judgment (Weak).
 - 2.1.3 If the *Tadlees* property value is 4-5, then the narrator will get a judge (Weak).
 - 2.2 Check the “*hasIkthelat*” property:
 - 2.2.1 If *hasIkthelat* property value is = no, then the Isnad will get a judgment (Sahih).
 - 2.2.2 If *hasIkthelat* property value is yes, then check the cases of narrated this hadith *ikhtelatHadiths*
 - a. If the Hadith narrated before *Ikthelat* then the Isnad will get a judgment (Sahih).
 - b. If the Hadith is narrated is not distinguished then the Isnad will get a judgment (Weak).
 - c. If the Hadith narrated after *Ikthelat* then the Isnad will get a judgment (Weak).
 - 2.3 Check the *hasIrsal* property:
 - 2.3.1 If *hasIrsal* property value is = no, then the Isnad will get a judgment (Sahih).
 - 2.3.2 If *hasIrsal* property value is yes, then check that the narrator has *HeardFrom* the teacher narrator:
 - a. If *HeardFrom* = yes then the Isnad will get a judgment (Sahih).
 - b. If *HeardFrom* = no, then the Isnad will get a judgment (Weak).
 - 2.4 Check the *hasBedaa* property:
 - 2.4.1 If *hasBedaa* property value is = no then the Isnad will be judged (Sahih).
 - 2.4.2 If *hasBedaa* property value is = yes, then check that the if the Hadith support his *Bedaa*:
 - a. If the Hadith support *Bedaa* = no then the Isnad will get a judgment (Sahih).
 - b. If the Hadith support *Bedaa* = yes, then the Isnad will get a judgment (Weak).

- 2.5 Check the *hasBedaa* property:
 - 2.5.1 If *hasBedaa* property value is = no then the Isnad will be judged (Sahih).
 - 2.5.2 If *hasWahem* property value is = yes, then check that the if the Hadith support his Wahem:
 - a. If the Hadith support Wahem = no then the Isnad will get a judgment (Sahih).
 - b. If the Hadith support Wahem = yes, then the Isnad will get a judgment (Weak).
3. If the Rotba property value is 4, then the judgment will be taken from the following steps:
 - 3.1 Check the *Tadlees* property, this property has values 1-5.
 - 3.1.1 If the *Tadlees* property value is 1-2, then the Isnad will get judgment (Hasan).
 - 3.1.2 If the *Tadlees* property value is 3, then check that the narrator has *HeardFrom* the teacher narrator:
 - a. If *HeardFrom* = yes then the Isnad will get a judgment (Hasan).
 - b. If *HeardFrom* = no, then the Isnad will get a judgment (Weak).
 - 3.1.3 If the *Tadlees* property value is 4-5, then the narrator will get a judge (Weak).
 - 3.2 Check the *hasIkthelat* property:
 - 3.2.1 If *hasIkthelat* property value is = no, then the Isnad will get a judgment (Hasan).
 - 3.2.2 If *hasIkthelat* property value is yes, then check the cases of narrated this hadith *ikhtelatHadiths*:
 - a. If the Hadith narrated before *Ikthelat* then the Isnad will get a judgment (Hasan).
 - b. If the Hadith is narrated is not distinguished then the Isnad will get a judgment (Weak).
 - c. If the Hadith narrated after *Ikthelat* then the Isnad will get a judgment (Hasan).
 - 3.3 Check the *hasIrsal* property:
 - 3.3.1 If *hasIrsal* property value is = no, then the Isnad will get a judgment (Hasan).
 - 3.3.2 If *hasIrsal* property value is yes, then check that the narrator has *HeardFrom* the teacher narrator:
 - a. If *HeardFrom* = yes then the Isnad will get a judgment (Hasan).
 - b. If *HeardFrom* = no, then the Isnad will get a judgment (Weak).
 - 3.4 Check the *hasBedaa* property:
 - 3.4.1 If *hasBedaa* property value is = no then the Isnad will be judged (Hasan).
 - 3.4.2 If *hasBedaa* property value is = yes, then check that the if the Hadith support his *Bedaa*:

- a. If the Hadith support *Bedaa* = no then the Isnad will get a judgment (Hasan).
 - b. If the Hadith support *Bedaa* = yes, then the Isnad will get a judgment (Weak).
- 3.5 Check the *hasWahem* property:
- 3.5.1 If *hasWahem* property value is = no then the Isnad will be judged (Hasan).
 - 3.5.2 If *hasWahem* property value is = yes, then check that the if the Hadith support his Wahem:
 - a. If the Hadith support Wahem = no then the Isnad will get a judgment (Hasan).
 - b. If the Hadith support Wahem = yes, then the Isnad will get a judgment (Weak).
4. If the *Rotba* property value is 5-12, this Isnad will get judge (Weak).
5. Finally we will get the lowest judgment from the previous judgment
- 5.1 If the lowest judgment is Sahih then the final judgment of Isnad is (Sahih).
 - 5.2 If the lowest judgment is Hasan then the final judgment of Isnad is (Hasan).
 - 5.3 If the lowest judgment is Weak then the final judgment of Isnad is (Weak).

5.4.2 Hadith Number Six Example (continued)

We continue the Hadith Number Six example to show how we apply HIJ algorithm to get the judgment of Isnad. In the example, we show that the extracted terms, properties and relations of Hadith are represented in the ontology. Then we identified the five narrators of Hadith Number 6 and returned information of each narrator using the IJS ontology. Then we check the continuity of Isnad using IJS ontology.

In Applying, the HIJ algorithm we check the properties (*Rotba*, *hasIkthelat*, *ikthelatHadiths*, *hasIrsal*, *hasBedaa*, *bedaaHadiths*, *hasWahem* and *wahemHadiths*) of each narrator as shown in Table 5.9.

Table 5.9: Results of Applying HIJ Algorithm

No.	Property	Muhammad Ibn Bashar	Muhammad Ibn Jaafar	Shuba Ibn Al Hajaj	Mu'awiyah Ibn Qurrah	Qurrah Ibn Iyas
1	<i>Rotba</i>	3	3	2	3	1
2	<i>Tadlees</i>	0	0	0	0	0
3	<i>Ikthelat</i>	no	no	no	no	no
4	<i>Irsal</i>	no	no	no	no	no
5	<i>Bedaa</i>	no	no	no	no	no
6	<i>Wahem</i>	no	no	no	no	no

The algorithm depends on these properties in judging the Hadith Isnad. As result of applying the algorithm with these properties, the Isnad judgment of the said Hadith is Sahih.

5.5 Summary

In this chapter, we have explained the algorithm for Hadith Isnad judgment. This algorithm consists of four steps; we explained each step in separate section.

In the first section, we explained the process of separating the Isnad form the Matn. We use the e-Narrator system and converting it to Web Service. This converted to preserve the rights of others.

In this section, we explained the idea of e-Narrator system briefly then we explained how convert it to a Web Service which illustrated from the WSDL file. We also evaluate the results through calling it from the web and compare the results with real results. Since the system has applied to the Sahih Muslim book. We noticed some shortcomings when used with a Sunan Ibn Majah book this shortcoming is very simple and does not prevent its use in IJS system.

In the second section, we explained the second step of Isnad judgment, which is identifying Hadith narrators using the IJS ontology. In this step, we have a challenge in the names of the narrators, we explained the reasons for this challenge, and how we confronted and resolved. We set six rules for the use of the IJS ontology in identify the names of narrators , to make sure that rules is accurate, we complete example on Hadith No. 6 and confirm the accuracy of ontology in return narrators information.

In the third section, we explained the third step of Isnad judgment that is to check the continuity of Isnad. This step based on an ontology reasoner to check the following properties: *HeardFrom*, *notHeardFrom*, and *TransferTo* for each narrator. The Hadith No. 6 example shows that all the narrators heard from their teachers and each narrator transfer to their students.

In the fourth section, we explained the fourth step of Isnad judgment that is applying the judgment rules. We extracted all judgment rules and put the HIJ algorithm as shown in Figure 5.3 then we explained all the conditions of the algorithm. To make sure that algorithm is accurate we completed the Hadith No. 6 example and it proved the accuracy of algorithm in judging the Hadith Isnad.

Chapter 6

IJS Prototype

In this chapter, we present in details the proof-of-concept prototype to realize the proposed Isnad judgment system. The development of the prototype consists of four phases: system analysis, system design, system implementation, and system testing. In the following four sections, we describe each phase.

6.1 System Analysis

In this section, we present a complete description of the behavior of IJS prototype including a set of use cases that describe interactions, functional requirements and non-functional requirements, which impose constraints on the design or implementation.

6.1.1 Overall Description:

We develop a prototype for the ontology-based Isnad judgment system that automatically generates suggested judgment of Hadith Isnad. It is based on the rules Hadith scholars follow to produce a suggested judgment. To satisfy this goal we divide the prototype into the following six components:

1. User Web Interface: to allows the user to search Hadiths and show the results of judging the Hadith Isnad.
2. Hadith Database: to store the Hadith books.
3. e-Narrator Web Service: to extract all narrators' chains from Hadith.
4. IJS Ontology: to identify narrators' names and information that is needed in the process of Isnad judgment.
5. Judgment Rules: These rules are based on the HIJ Algorithm see Section 5.4.1(Hadith Isnad Judgment Algorithm).
6. Judgment System: to judge Hadith Isnad using the judgment rules.

6.1.2 System Functions

We visualize high-level system functions and requirements by drawing use case diagrams, which contain primarily actors and use cases. Actors are entities that interact with the systems, while use cases are the system functions actors involve. The IJS prototype supports the following use cases:

Table 6.1: System Use Cases

Use Case	Actor	Figure No.
Display the Hadith by browsing	researcher	1
Display the Hadith by searching	researcher	2
Verify the Isnad of Hadith	researcher	3
Add new narrator information	administrator	4
Update narrator information	administrator	5
Add new book information	administrator	6
Update book information	administrator	7
Add new Hadith information	administrator	8
Update Hadith information	administrator	9

- **User Characteristics**

- a. The user should be familiar with the IJS terminology respectively with Hadith terminology.
- b. The user should smatter of Hadith Isnad Judgment.

- **Principal Actors**

The two principal actors in IJS are the researcher and the administrator.

6.1.3 Specific Requirements

This section will present specific requirements of IJS that covers its various functions.

- **Functional Requirements**

- The system shall enable the researcher to display the Hadith by browsing.
- The system shall enable the researcher to display the Hadith by searching.
- The system shall enable the researcher to get the judgment of Hadith.
- The system shall enable the administrator to add new narrator information.
- The system shall enable the administrator to update narrator information.
- The system shall enable the administrator to add new book information.
- The system shall enable the administrator to update book information.
- The system shall enable the administrator to add new Hadith information.
- The system shall enable the administrator to update Hadith information.

We describe each functional requirement by giving various use cases that define interactions between a role and IJS system

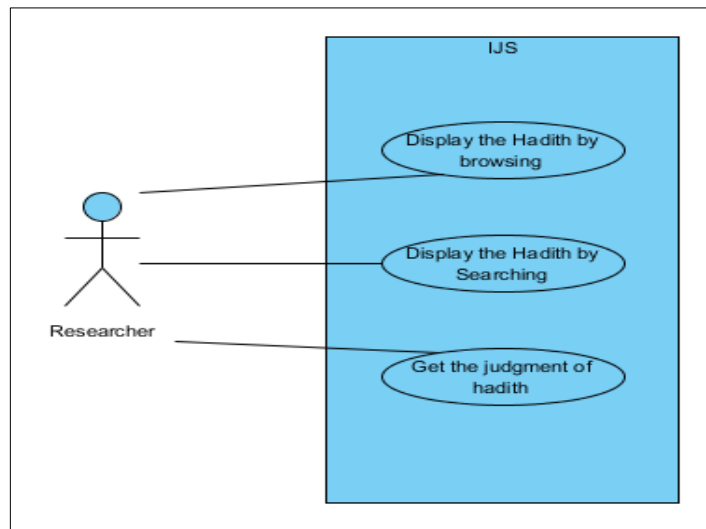


Figure 6.1: Use Cases of Researcher

Use Case 1: Display the Hadith by Browsing.

Primary Actor	Researcher
Main Scenario	<ol style="list-style-type: none"> 1- Researcher clicks on the book list. 2- System displays the Hadith books. 3- Researcher chooses the Hadith.

Use Case 2: Display the Hadith by Searching.

Primary Actor	Researcher
Main Scenario	<ol style="list-style-type: none"> 1- Researcher adds the Hadith number in the search box. 2- System displays the desired Hadith.
Alternate Scenario	<ol style="list-style-type: none"> 1- If the Hadith not found, the system should give hints to the researcher. 2- If the information not found, the system gives hints to the researcher.

Use Case 3: Get the Judgment of Hadith.

Primary Actor	Researcher
Main Scenario	<ol style="list-style-type: none"> 1- Researcher chooses any Hadith by exploring or searching. 2- System gets the judgment of Hadith. 3- System displays the judgment of Hadith.

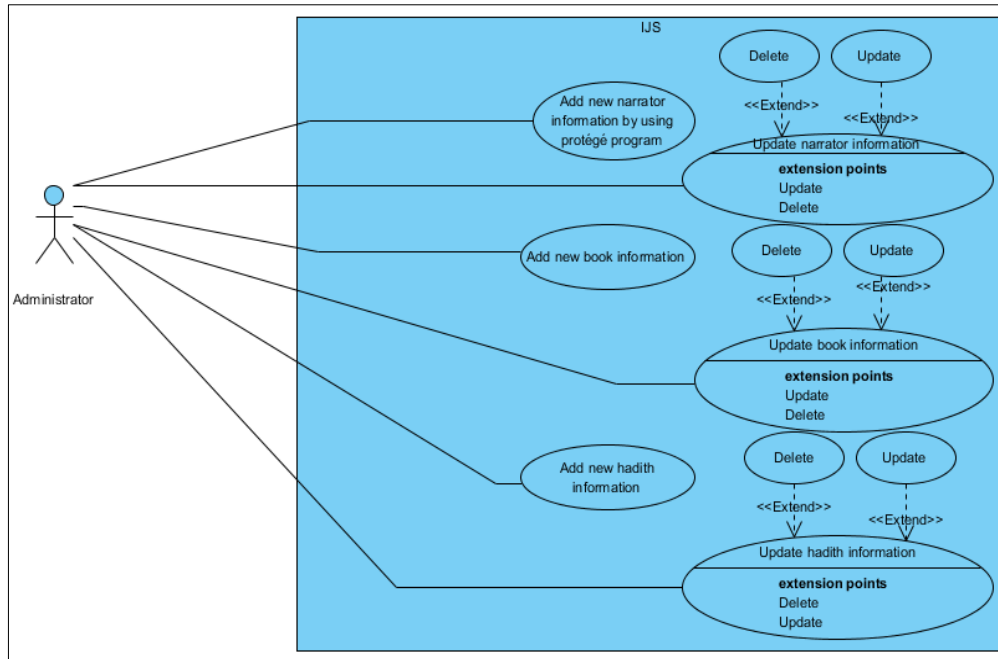


Figure 6.2: Use Cases of Administrator

Use Case 4: Add new narrator information by using protégé program.

Primary Actor	Administrator
Main Scenario	<ol style="list-style-type: none"> 1- Administrator click add new narrator information. 2- System displays the form to add the narrator information contains the following: Table 4.2 and Table 4.3. 3- Administrator enters the narrator information and submits.
Alternate Scenario	<ol style="list-style-type: none"> 1- If the narrator exists, system gives hints to the researcher such as “Narrator Already Exists”. 2- If the information error: system gives hints to the researcher.

Use Case 5: Update narrator information.

Primary Actor	Administrator
Main Scenario	<ol style="list-style-type: none"> 1- Administrator click updates the narrator information. 2- System displays the form to update the narrator information contains the following: Table 4.4. 3- Administrators enter the narrator information and submit.
Alternate Scenario	<ol style="list-style-type: none"> 1- If the information error: system gives hints to the researcher.

Use Case 6: Add new book information.

Primary Actor	Administrator
Main Scenario	<ol style="list-style-type: none"> 1- Administrator click add new book information. 2- System displays the form to add the narrator information contains the following: Table 3. 3- Administrator enters the book information and submits.
Alternate Scenario	<ol style="list-style-type: none"> 1- If the book exists, system gives hints to the researcher such as “Book Already Exists”. 2- If the information error: system gives hints to the researcher.

Use Case 7: Update book information.

Primary actor	Administrator
Main scenario	<ol style="list-style-type: none"> 1- Administrator click update book information. 2- System displays the form to update the book information contains the following: Table 4.4 3- Administrator enters the book information and submits.
Alternate Scenario	<ol style="list-style-type: none"> 1- If the information error: System gives hints to the researcher.

Use Case 8: Add new Hadith information.

Primary actor	Administrator
Main scenario	<ol style="list-style-type: none"> 1- Administrator click add new Hadith information. 2- System displays the form to add the Hadith information contains the following: Hadith number, Book name, chapter’s name, content 3- Administrator enters the Hadith information and submits.
Alternate Scenario	<ol style="list-style-type: none"> 1- If the Hadith exist, system gives hints to the researcher such as “Hadith Already Exists”. 2- If the information error: system gives hints to the researcher.

Use Case 9: Update Hadith information.

Primary actor	Administrator
Main scenario	<ol style="list-style-type: none"> 1- Administrator click update Hadith information. 2- System displays the form to update the Hadith information contains the following: Hadith number, Book name, Chapter’s name and Content 3- Administrator enters the Hadith information and submits.
Alternate Scenario	<ol style="list-style-type: none"> 1- If the information error: system gives hints to the researcher.

- **Nonfunctional Requirements:**

- Accessibility: The system will be developing as a Web-Based application using JSP programming language to be available to as many people.
- Response Time: the system will execute end user request in small interval. This will represent in response time of judgment the Hadith comparing with manual method.

6.1.4 Hadith Isnad Judgment Process

IJS process is a collection of related and structured tasks that produce Isnad judgment for a researcher is explained in chapter 5 (Algorithm for Isnad Judgment). We visualized it with a flowchart as a sequence of activities with interleaving decision points as shown in Figure 6.3.

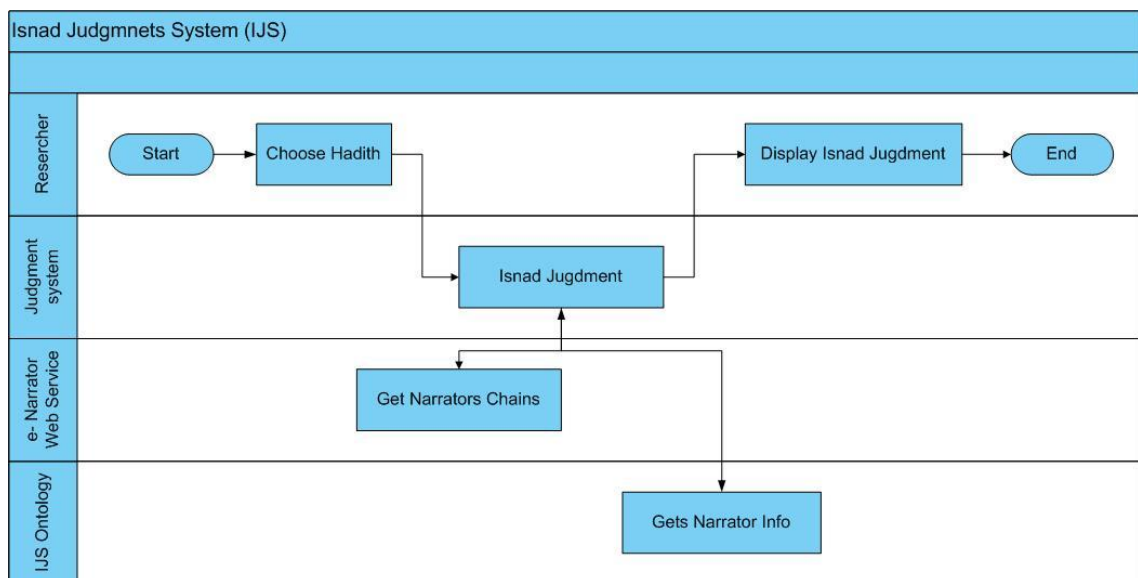


Figure 6.3: Hadith Isnad Judgment Process

The process of IJS system starts when the researcher selects the Hadith that he wants to judge his Isnad. Then the IJS will pass the Hadith to e-Narrator Web Service to extract the Isnad from Matn and then it will pass the narrators chain to IJS ontology to get all information about each narrator. Then it will apply the Hadith judgment algorithm see Section 5.4.1 (Hadith Isnad Judgment Algorithm), to get the judgment of Isnad that will display to the researcher.

6.2 System Design

In this section, we present how the system is designed to satisfy the requirements identified in the previous phase. The requirements identified in the system analysis phase are transformed into a system design that accurately describes the design of the system and that can be used as an input to system implementation in the next phase.

6.2.1 System Architecture

In this section, we provide high-level description of the IJS prototype and identify all its elements at some level of abstraction. This contrasts with low-level design, which elaborates the detailed design of each of these elements.

The IJS architecture in Figure 6.4 shows the components and the dependencies between them and shows the interaction between these components.

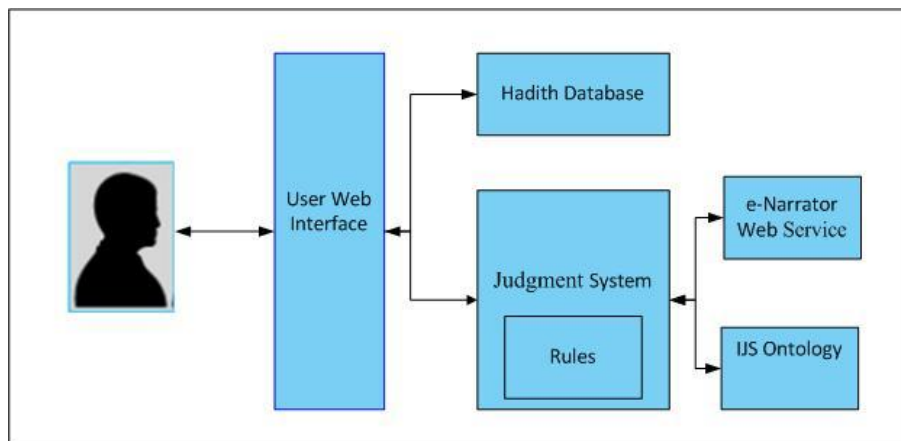


Figure 6.4: System Architecture

The system starts working once the user chooses a Hadith to judge from the user web interface. This request is directed to the e-Narrator Web Service that will analyze this Hadith to split the Isnad from Matn and extract the narrators to produce the narrators' chain.

The narrators' chain will process by the judging system that will process it using the IJS ontology, to get the Hadith narrators and match the chain name with its narrator. The system uses properties such as *FirstName*, *DenotedBy*, *Nasab*, *Konia*, *Luqab*, and *NazeelOf* to achieve that. Once all the chain's narrators are determined correctly, the judging process starts.

In the judging process, many properties taken into consideration see Section 5.4.1 (Hadith Isnad Judgment Algorithm), to generate the final Isnad judgment of the Hadith. The judgment will be one of three levels: Sahih, Hasan, and Weak.

6.2.2 IJS Components

The IJS system consists of six components shown in Figure 6.4. We explain the design of each component:

1. User Web Interface

This interface allows a user to search and explore Hadiths and choose among them for Isnad judgment. Figure 6.5 shows user interface, which contain three parts, first part is a list of Sunnan, books and chapters, second part for searching, third for choose and judge the Hadith.

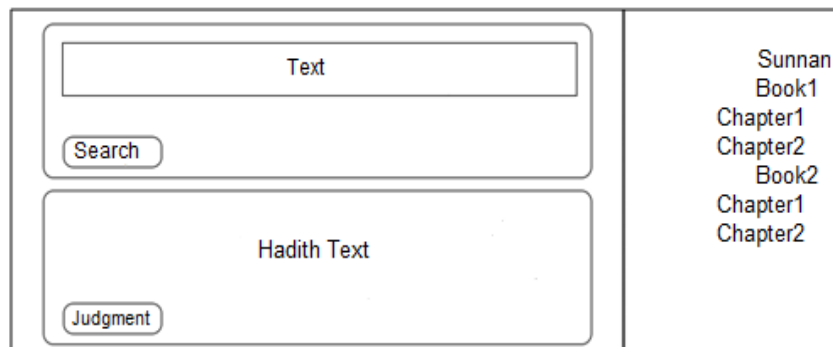


Figure 6.5: Searching and Exploring Hadiths Interface

In addition, the interface displays the judgment results to the user. Figure 6.6 shows the output interface, which contain the Hadith text, narrator chain properties, and the judgment of Isnad.

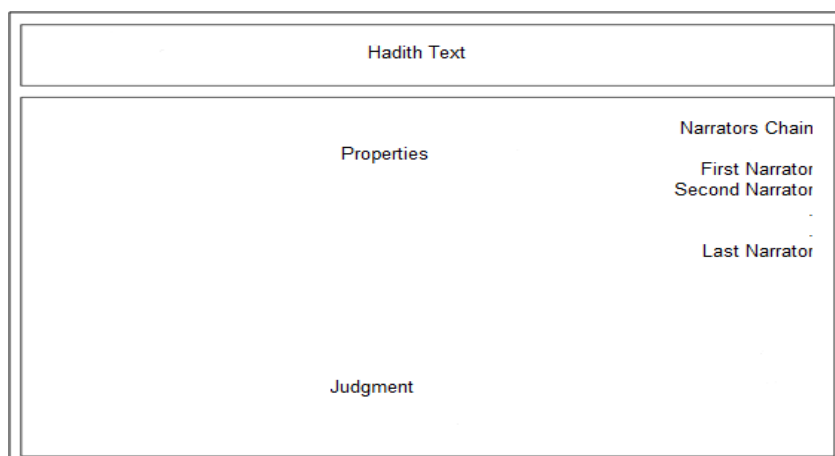


Figure 6.6: Output Interface

2. Hadith Database

This database is used to store the Hadith books. In this database, we have four tables that are Book, Sunan, Chapter, and Hadith. The relations between these tables are illustrated in Figure 6.7 that represents the ER-Diagram of the Hadith books database.

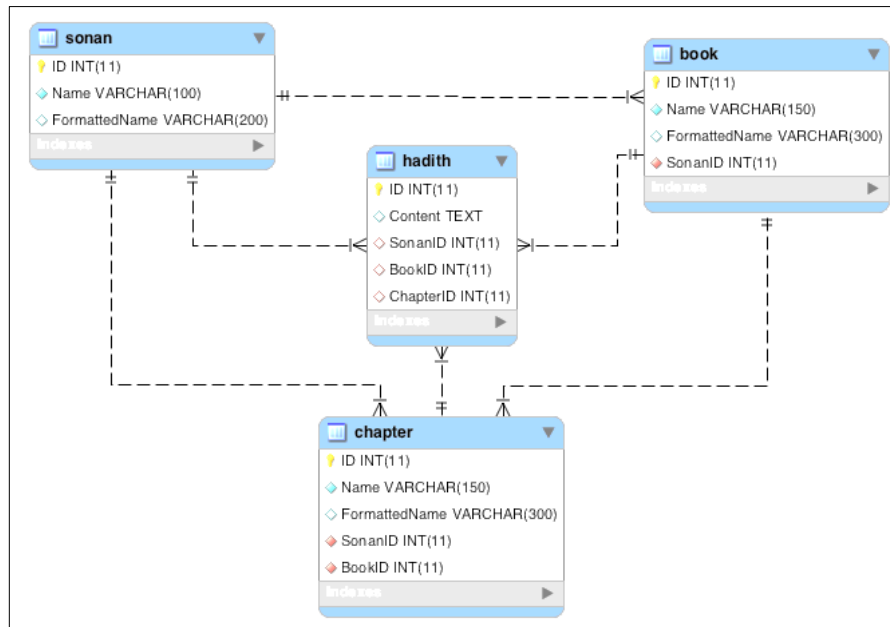


Figure 6.7: Hadith Books Database ER-Diagram

3. e-Narrator Web Service

This Web Service analyzes the Hadith by splitting its Isnad away from its Matn and extracting all narrators' chain of the Hadith. We explained the design of e-Narrator in Section 5.1 (Separating the Isnad from Hadith).

4. IJS Ontology

IJS ontology contains all narrator information and the relation between them. It is used to identify the narrator's name and information that is needed in the process of judging the Isnad. We explained the design of IJS ontology in Chapter 4: (IJS Ontology Development).

5. Judgment Rules

We use the HIJ algorithm to judge the Hadith Isnad, which is explained in Section 5.4 (Applying the Judgment Rules).

6. Judgment System

Ontology-based system uses the narrators' chains that are returned from e-Narrator Web Service and identify each narrator using IJS ontology, then applying the HIJ algorithm to judge the narrators chain.

6.2.3 IJS Diagrams

In this section, we show how the IJS system inters operates at a very high level and how processes of IJS operate with one another.

- **Deployment Diagram**

It describes the static deployment view of IJS prototype and it consists of nodes and their relationships as shown in Figure 6.8.

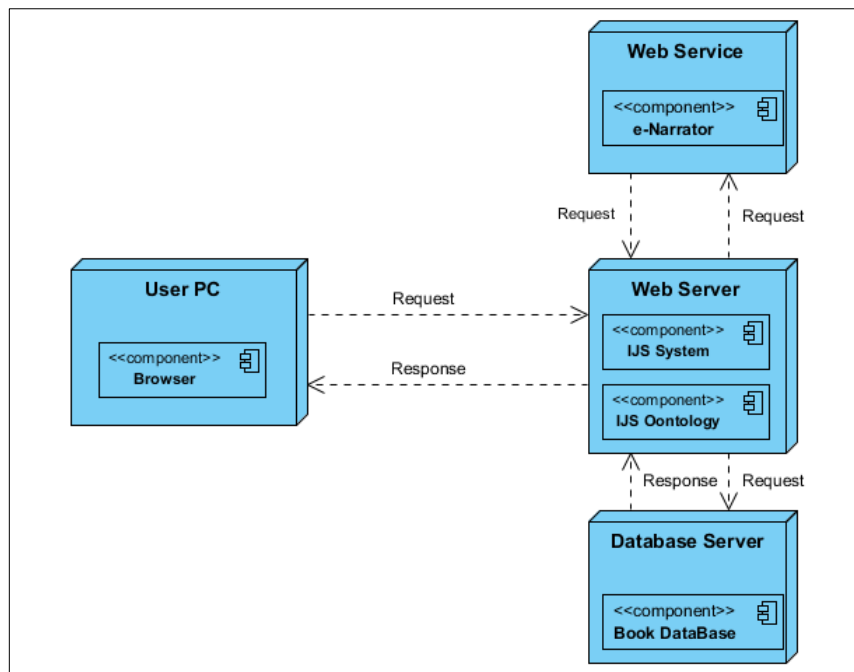


Figure 6.8: Deployment Diagram of IJS

The IJS needs three servers: the Web Server, which contains the IJS system and the IJS ontology, Database Server, which contains Hadith books database, Web Service server, which contains the e-Narrator service. The user uses his browser to interact with the system.

- **Sequence Diagram**

Figure 6.9 shows how processes of IJS operate with one another and in what order and shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects to carry out the functionality of the scenario. The sequence diagram presents the main use case in the IJS prototype. For the others use cases, you can refer to Appendix A: IJS Analysis and Design Components

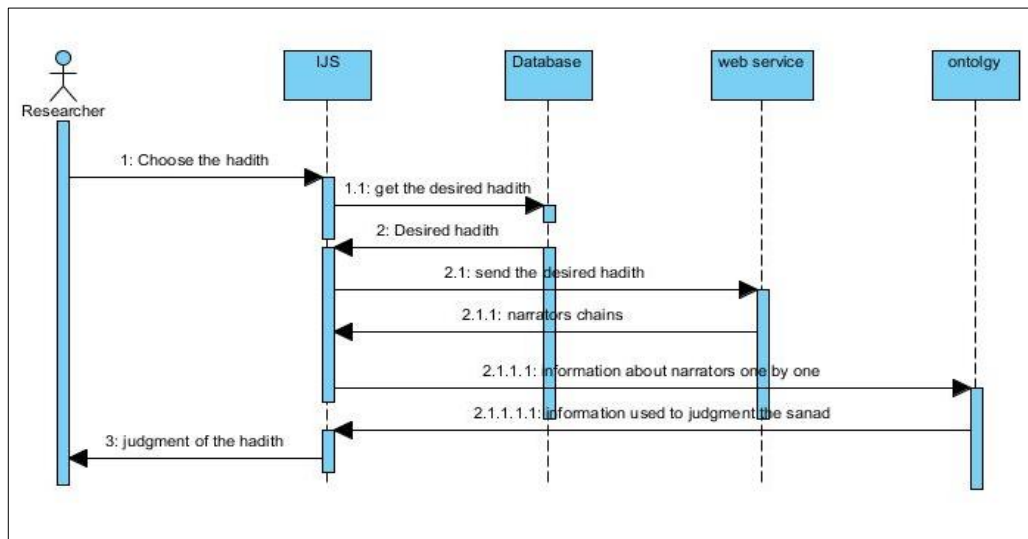


Figure 6.9: Get the Judgment of Hadith.

The scenario starts when the researcher chooses a Hadith to judge its Isnad. The IJS returns it from Hadith book database. Then IJS sends the desired Hadith to e-Narrator Web Service to extract the narrators' chains from Isnad. The IJS compares matches narrator name to IJS ontology to get all information about each narrator, then it will apply the algorithm to get the judgment of Isnad, which returns to the researcher.

6.3 System Implementation

In this section, we present how the system is implemented according to the previous design of IJS components that are designed in system design phase, see Section 6.2.2 (IJS Components). Designs are worked out during earlier steps and are translated into code.

6.3.1 User Web Interface

This component is implemented using JSP and MySQL database. JSP is connected to MySQL database server that contains Hadith books. It allows the user to search through these books and explore Hadiths. This would enable the user to choose the Hadith to judge its narrators' chain. The component loads data asynchronously using Asynchronous JavaScript and XML (AJAX) and JavaScript Object Notation (JSON). This method allows the system not to load all the books in the first page. The data will be loaded upon user request in very small HTTP requests. For coding implementation, see Appendix B: IJS Implementation). Figure 4.8 shows an example of the exploring interface.



Figure 6.10 Searches and Exploring Interface.

The design of the user web interface in Section 6.2.2 (IJS Components) illustrate that the user interface contain three parts. First part is a list of Sunnan, books and chapters. The second part is for searching. The third part for choose and judge the Hadith. In the implementation of this interface we create, a web page contains the three parts.

In the implementation of Hadith searching we have a limitation, which is could not search in Hadith text by any word, this limitation is due to the difficulties in Arabic language that contains formation on the characters. Therefore, we override this point in a prototype by searching the Hadith by id.

In addition, the design of the output interface contains the Hadith text, narrator chain properties, and the judgment of Isnad. The implemented parts are reflected in the web page shown in Figure 6.11.



The screenshot displays a web interface for Hadith results. At the top, it shows the title "سنة ابن ماجه - افتتاح الكتاب في الإيمان وفضائل الصحابة والعم - باب اتباع سنة رسول الله صلى الله عليه وسلم". Below the title, there is a paragraph of text in Arabic, followed by a note in brackets: "[أش (لم يجد) أي لم يجاوز بلزومه؛ على قدر فراد في الحديث ولا يفرق فيه ولم يتسر في التفسير منه] . صحیح".

Below the text is a table titled "السلسلة رقم (1)". The table has the following columns: "م", "رقم دروي", "اسم دروي", "درجۃ", "درجۃ للتكلیف", "مخلف", "مرسل", "له بدعة", "له أرقام", "حكم إسناده".

م	رقم دروي	اسم دروي	درجۃ	درجۃ للتكلیف	مخلف	مرسل	له بدعة	له أرقام	حكم إسناده
1	6053	مسند بن عبد الله بن نصر الخوافي لهداني الكوفي لحفظ	2	0	X	X	X	X	صحیح
2	3570	عبد الله بن عبد الملك بن واضح الخنطلي البرزنجي القنوي	2	0	X	X	X	X	صحیح
3	5942	مسند بن سورة الخنطلي لأميد الكوفي	2	0	X	X	X	X	صحیح
4	3490	عبد الله بن عمر بن الخطاب الحموي البغدادي للراشي لشي	1	0	X	X	X	X	صحیح

At the bottom right of the table, it says "حكم إسناده: صحیح".

Figure 6.11: Results Interface

6.3.2 Hadith Database

This database is created using phpMyAdmin tool that is used to manage MySQL database. The database creation done according to ER-Diagram, see Figure 6.7 in Section 6.2.2 (IJS Components). In addition, the database creation in IJS system is implemented using Java language.

We insert the data of Sunnan Ibn Majah because we choose from it all samples that is used from system evaluation. The data contains the books, chapters, and Hadiths.

6.3.3 e-Narrator Web Service

This Web Service is implemented as a .Net Web Service according to the design that is illustrated in Section 6.2.2 (IJS Components).

e-Narrator Web Service analyzes the Hadith to split the Isnad and then extracts the narrators to produce the narrators' chain. The result is formatted as a JSON string to make it easier to parse and transfer.

Figure 6.12 presents Hadith number six example as input to the Web Service, this example is used in four Sections 4.2.1, 5.2.2, 5.3.1, and 5.4.2.

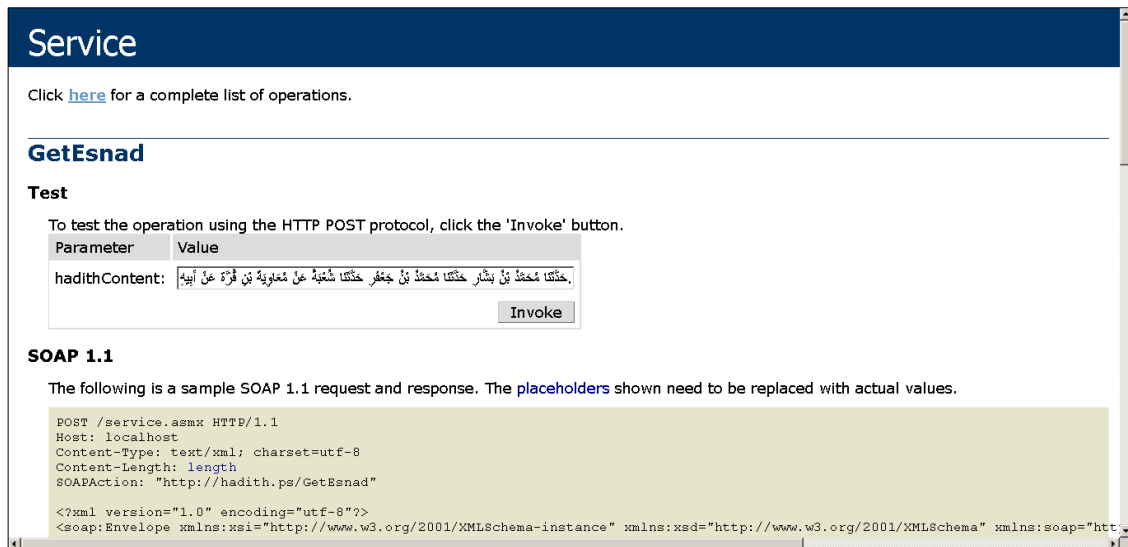


Figure 6.12: Hadith Input to e-Narrator Web Service

Figure 6.13 shows the expected output from analyzing this Hadith and showing the format of the output as a JSON array. To show the design of Web Services see (Section 5.1 Separating the Isnad from Hadith).



Figure 6.13: Narrator Chain as Output of e-Narrator Web Service

6.3.4 IJS Ontology

The ontology is stored in IJS Web Server and accessed by the system using OWL-API when the system needs to get the information of narrators. We explain the developments and implementation of IJS ontology using Protégé in Section 4.1 (IJS Ontology Development).

6.3.5 Judgment Rules

The HIJ algorithm is implemented using Java language as a part of Judgment rules explained in Section 5.4 (Applying the Judgment Rules).

Figure 6.14 shows part of implementing the HIJ algorithm. For the complete implementation, see Appendix B: IJS Implementation).

```

42
43 case 4:
44     judge = String.format , "s% من الرتبة ")levels[rotba]);
45     currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba], judge));
46     //Test Tadlees
47     int tadlees = currentNarrator.getTadlees();
48     if (tadlees == NEGATIVE_PROPERTY) {
49         judge = "لم يثبت عليه تدليس";
50         currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba], judge));
51     } else if (tadlees == 1 || tadlees == 2) {
52         judge = String.format , "s% من الدرجة ")levels[tadlees]);
53         currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba], judge));
54     } else if (tadlees == 3) {
55         if (currentNarrator.hasHeard()) {
56             judge = String.format , "'s%' ورجح بالسماع بقوله ")levels[tadlees], currentNarrator.
57             currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba], judge));
58         } else {
59             judge = String.format , "s% وقوله 's%' لا يفيد التمريح ")levels[tadlees], currentNarrator.
60             currentNarrator.getJudges().add(new Judge(3, judge));
61         }
62     } else {
63         judge = String.format , "s% من الدرجة ")levels[tadlees]);
64         currentNarrator.getJudges().add(new Judge(3, judge));

```

Figure 6.14: Part of the Implemented Algorithm

6.3.6 Judgment System

This component is the core of IJS system as illustrated in the system architecture; see Section 6.2.1(System Architecture). This part controls all components of the system as illustrated in sequence diagram, see Section 6.2.3 (IJS Diagrams).

The IJS ontology plays the major role in system that is store the data about Hadiths, the narrators and the relations between Hadiths and narrators, and the relations between the narrators themselves as illustrated in Section 3.2 (IJS Ontology Development).

In this part, A Java entity classes is implemented to realize and hold the data retrieved from the ontology. The class “PathsJudgment” is responsible for judging all the paths to produce judged paths. The class has two methods: The first is judgePaths method and it takes the chains as two-dimensional array and the Hadith to judge and returns the judgment of each chain. The second method is judgePath method and it takes an array of the chains and returns the final judgment for all chains is the Isnad contains more than one chain.

All these steps are managed by a Servlet, which receives the judgment request. This Servlet does all of these steps sequentially as follows:

Step 1: The e-Narrator Web Service is called through sending the Hadith to receive the narrators' chain as JSON string.

```
String jsonChain = HadithWS.getEsnad(aHadith.getContent());
```

Step 2: A two-dimensional array build as a JSON string that will hold the narrators chain.

```
ArrayList<ArrayList<Narrator>> allPaths =  
NarratorFullInfoHandler.getFullNarratorInfo(jsonChain, HadithId);
```

Step 3: The judging process will start judging these chains based on the algorithm of Isnad judgment see Chapter 5 (Algorithm for Isnad Judgment).

```
PathsJudgment.judgePaths(allPaths, HadithId);
```

Step 4: The judgment result is forwarded to the user Web Interface using these three lines of code.

```
request.setAttribute("Hadith", aHadith);  
request.setAttribute("AllPaths", allPaths);  
request.getRequestDispatcher("./WEB-INF/judgment.jsp").forward(request, response);
```

6.4 System Testing

In this section, we present the testing of each part of IJS system by using JUnit Testing to determine whether it behaves exactly as we expect. Each part is tested separately before integrating them into the IJS System to test the interfaces between modules. JUnit is a simple open source Java testing framework is used to write and run repeatable automated tests. It is an instance of the xUnit architecture for unit testing framework

We have tested the three components that can test by the JUnit test. We wrote the test cases using NetBeans IDE for the functions of user web interface, e-Narrator Web Service and the judgment system. Each of them have passed the required test. This proved that the prototype is implemented successfully see Appendix E: System Testing).

6.5 Summary

In this chapter, we have presented the phases of building the prototype of IJS system. It consists from the following four phases:

In the analysis phase, we analyzed and specified the requirements of the system and divided the system into six components: user web interface, Hadith database, e-Narrator Web Service, IJS ontology, judgment rules, and judgment system. We also described the functionality of the system through the use cases. Then we explained the IJS process through flowchart of sequence activities with interleaving decision points shown in Figure 6.3.

In the design phase, we explained the interaction and dependencies between these six components in the IJS architecture (Figure 6.4). Then we explained in details the design of these six components. Then we explained how the IJS system inters operates at high level and how processes of IJS operate with one another.

In the implementation phase, we explain in details the implementation of these six components according to the design that illustrated in the design phase. The implementation of the user interface contained items that have identified in the design phase, which is a list of Sunnan, books and chapter parts, the searching part, and the judgment part. The Hadith database implemented by phpMyAdmin tool, the e-Narrator Web Service implemented as a .Net Web Service, and the IJS ontology implemented using Protégé, The HIJ algorithm implemented using Java language, and judgment system implemented using Java language and managed by a Servlet.

In the testing phase, we tested the three components that can test by the JUnit test. We wrote the test cases using NetBeans IDE for the functions of user web interface, e-Narrator Web Service and the judgment system. Each of them passed the required test. This proved that the prototype implemented successfully.

Chapter 7

Results and Evaluation

In this chapter, we present the procedure we performed to evaluate the accuracy of the proposed IJS prototype and IJS ontology.

7.1 IJS Prototype Evaluation

To evaluate the system we have asked the expert in Isnad judgment to select Hadith sample, which contains the various cases to examine through the system including Sahih, Hassan, and weak Hadiths. In addition contains Ikhtelat, Irsal, tadlees, Wahem and Bedaa. Moreover, contains family relations.

After searching for the Hadith samples that match the criteria required in the Sunan Ibn Majah book because it include all cases of judgment like Sahih, Hassan, and weak, we found that these criteria is represented in sixteen Hadiths. Then we accredited them as a sample test to evaluate the accuracy of the system. To verify the accuracy of the system we performed two approaches:

- **Accuracy According to Al-Albani Scholar:** Where the Al-Albani judged the Hadith Isnad and Matn in Sunan Ibn Majah book and his judgment always is on top of the judgment on Isnad only.
- **Accuracy According to Hadith Specialist:** Because the Al-Albani judged on Hadith Isnad and Matn, we asked the expert in Isnad judgment to judge this Hadith sample.

7.1.1 Accuracy According to Al-Albani Scholar

This section explains how the experimental test is applied to evaluate the accuracy of the IJS prototype according to Al-Albani scholar, which judge the whole Hadith (Isnad and Matn)

Sheikh Nasser Eddin Al-Albani is one of the scientists who has worked on the science of Hadith and Sunnah. He knows the trusted men and others and became preoccupation tahkij of Hadith.

The approach of Sheikh Al-Albani in judging the Hadith is:

1. Diligence and innovation, not imitation, so we find his saying “*I do not imitate anybody while I’m judging those Hadiths, but follow scientific rules developed by the Hadith scholars to judgment on the Hadith of Sahih or weak*”.
2. He begins his judgment by mentioning the Hadith Matn.
3. Then he mentions his judgment.
4. Then he mentions who judge it in his book.
5. Then he mentions the judgment of imams like Nasaa'i.
6. He transferred from some manuscript books, and this is what distinguishes the work of Sheikh and great effort in tracking and Investigation,
7. Then he mentions the defendants narrator, or defendants narrators, and reviews the sayings of the Imams in them, and trying to survey and clear out by virtue of the narrators through the collection of the words of the imams, and discussed,
8. He mention other Hadith methods, and criticize if suitable rebounds and evidence or not. Here is where the science of Sheikh and his acquaintance, and his ability to criticize.

After this introduction about Sheikh Al Albani, we will present and clarify the results obtained after testing the IJS system according to Sheikh Al-Albani judgment as shown in Table 5.1.

Table 7.1: Accuracy According to Al albani

Hadith No.	Al-Albani Judgment	IJS Judgment
4, 6, 23	Sahih	Sahih
5, 48, 92	Hasan	Weak
21	Weak	No Judgment
30, 762	Sahih	Weak
54, 141	Weak	Weak
57	Sahih	Hasan
85	Hasan	Hasan
94	Sahih	No Judgment
439	Hasan	No Judgment
595	Weak	Hasan

The results shows that the numbers of same judgments on Hadiths are six and three cases are system error. These errors are due to the limitation on e-Narrator Web Service that is explained in Section 5.1.3 (Testing e-Narrator Web Service). While other seven is not same due to the reason mentioned earlier that the Al-Albani judgment done in terms of Isnad and Matn while the IJS system judgment done on the Isnad only.

Therefore, the judgment of Al-Albani is always more accurate than the judgment of the IJS system. If we recheck the results, we find six other judgments that can be identical where they are on top of the judgment and therefore the ratio of precision may be up to 75% as shown in Figure 5.1.

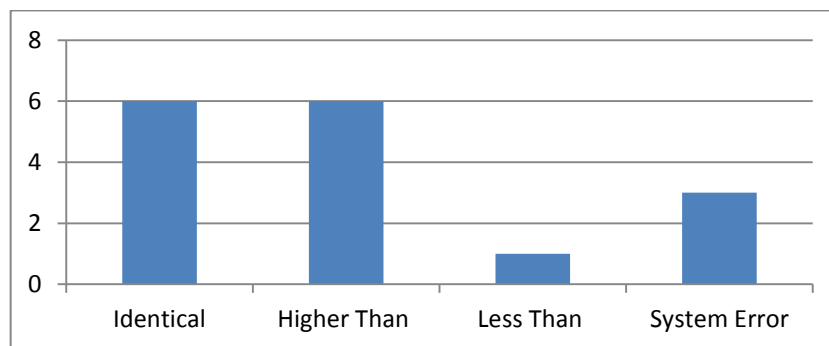


Figure 7.1: Results of IJS Compared to Al Albani

7.1.2 Accuracy According to Hadith Specialist

This section present the results of the experimental test applied to evaluate the accuracy of the IJS system according to a selected Hadith scholar.

To evaluate the prototype accurately we get the judgment of this sample from the Hadith specialist and the results were as in the Table 7.2, which shows the compatibility of the prototype judgment with the Hadith specialist judgment

Table 7.2: Accuracy According to Isnad Judgment by Hadith Scholar

Hadith No.	Hadith Specialist	IJS Judgment
4,6,23	Sahih	Sahih
57,85,595	Hasan	Hasan
5,30,48,54,92,141,762	Weak	Weak
21	Weak	No Judgment
94	Sahih	No Judgment
439	Hasan	No Judgment

These results show that that the numbers of same judgments on Hadiths are thirteen and three cases are system error. We have fully matched between the IJS System judgment and the Hadith Specialist judgment.

The percentage of accuracy compliance to 81%, this confirms the high accuracy enjoyed by the IJS System in Isnad judgment, beginning from analysis Hadith down to the result of judgment. This confirms success in proving the idea of this research in using ontology to support Hadith Isnad judgment.

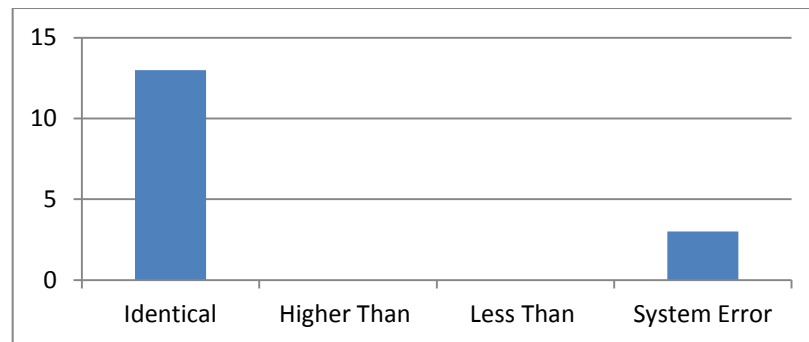


Figure 7.2: Results of IJS Judgment Compared to Hadith Specialist Judgment

7.2 IJS Ontology Evaluation

This section presents the evaluation of IJS ontology using the Task-Based framework that is illustrated in Section 3.2.3 (Ontology Evaluation). The elements of Task-Based are Task, Ontology, Application, and Gold Standard. In this evaluation, Isnad judgment represents the Task element, IJS ontology represents the Ontology element, IJS prototype represents the Application, and the judgment of Hadith specialist represents the Gold Standard.

For evaluating the performance of the IJS prototype, we have employed the semantic relation error types of the Task-Based framework is listed:

- **Deletions:** missing relations in places a relation ought to be identified. An example of deletion in this task is delete the (*TransferTo*) relation between the narrators, when we delete this relation the system could not identify the narrators.
- **Insertions:** postulating any relation to hold where none ought to be. An example of insertion in this task is insert new (*IkhtelatHadiths*) relation between the narrator and Hadith, when we insert this relation the judgment of Hadith Isnad will not be accurate.

- **Substitutions:** postulating a specific relation to hold where some other ought to been. For example if we substitute any relation with other relation, judgment of Hadith Isnad will not be accurate.

As compared to the Gold Standard, we obtained the accuracies, substitutions, deletions, and insertions as shown in Table 7.3.

Table 7.3: Results Overview

Overall Accuracy	81%
Substitutions	0%
Deletions	0%
Insertions	0%

These results provide a comparable measure for evaluating the performance of the IJS prototype system on this nontrivial task. They clearly indicate that the accuracy is 100% in the used IJS ontology. The 0% deletions indicate clear cases where a pertinent relation modeled in the ontology, the 0% of the substitution errors showed efficiencies in the model and the 0% of the insertions could be regard as superfluously model relations.

7.3 Summary

In this chapter, we have evaluated the system and have discussed the results. The evaluation divided into two sections:

In the first section, we evaluated the system according to following two approaches: according to Al-Albani scholar and according to Hadith specialist. The Results discussed in both approaches where the accuracy of the system in the first approach is 75% in the second approach is 81%. This is because Al-Albani scholar judge the Hadith Matn and Isnad, either we judge the Isnad only. These results prove that the ontology supports the process of Isnad judgment.

In the second section, we evaluated the ontology using Task-Based framework; this framework employed the semantic relation error types: deletions, insertions, and substitutions to validate the ontology. Then it compared the results with Gold Standard; if the accuracy of these semantic relation errors still 0% then it indicate that the accuracy of using the IJS ontology is 100%.

Chapter 8

Conclusion and Future Work

In this research, we have presented the proposed Isnad Judgment system (IJS). We first presented an overview of current electronic encyclopedias and Hadith applications, and showed that their primary shortcomings lie in the fact that they do not support the Isnad judgment.

We presented the development of our IJS ontology and the Hadith Isnad judgment algorithm, and we presented the role of ontology in the algorithm. We support that with example that contains all steps of algorithm.

IJS combines the ontology features with Hadith judgment rules to identify the narrators and get the Isnad judgment. The main components of the prototype are User Web Interface, Hadith Database, Judgment System, e-Narrator Web Service, and the IJS Ontology.

Two approaches are used to evaluate the accuracy of judgment of IJS and its prototype. An experimental method used to test the accuracy of the prototype. The first is based on Imam Al-Albani show that the accuracy is up to 75%. The second is based on Hadith specialist which shows that the accuracy is up to 81%. For evaluating the IJS ontology, we used the Task-Based framework that clearly indicates that the accuracy is 100% in using the IJS ontology.

The main contribution of this research is that the ontology can support the process of Hadith Isnad judgment with higher rate of accuracy and user satisfaction than manual Isnad judgment.

Since only a prototype of the proposed system is implemented, in a future work we look forward to implement a complete system. Success of our proposed IJS prototype encourages us to look for ways to increase the scope of this research to include more than the six books of Hadith and judge the Matn also.

In addition, we look to address the limitation in Arabic search that contains formation on the characters and address the limitation in the e-Narrator Web Service to include all Hadith books.

Bibliography

- [1] Aqil Azmi, Nawaf Badia, "iTree – Automating the Construction of the Narration Tree of Hadith", IEEE, 2010.
- [2] Lilac Al-Safadi, Mai Al-Badrani, Meshael Al-Junidey, "Developing Ontology for Arabic Blogs Retrieval", International Journal of Computer Applications (0975 – 8887) Volume 19– No.4, April 2011.
- [3] Natalya Noy, Deborah Guinness "Ontology Development 101: A Guide to Creating Your First Ontology Stanford University", Stanford, CA, 94305.
- [4] Kalfoglou, "Using Ontologies to Support and Critique Decisions", 1st International Conference on Knowledge Engineering and Decision Support, July 2004, Porto, Portugal .
- [5] Fouzi Harrag, Aboubekour Cherif, Eyas Qawasmeh, "Vector Space Model for Arabic Information Retrieval Application to Hadith Indexing", IEEE, 2008.
- [6] Kawther Aldhlan, Ahmed Zeki, AkramZeki, "Data mining and Islamic Knowledge Extraction: Hadith as A Knowledge Resource", Proceeding 3rd International Conference on ICT4M, 2010.
- [7] Nurazzah Rahman, Zainab Abu Bakar, Tengku Sembok, "Query Expansion using Thesaurus in Improving Malay Hadith Retrieval System", IEEE, 2010.
- [8] Saidah Saad, Naomie Salim, Hakim Zainal, Zurina Muda, "A process for building domain ontology: an experience in developing Solat ontology", 2011 International Conference on Electrical Engineering and Informatics, 17-19 July 2011, Bandung, Indonesia.
- [9] Shamela Library, <http://shamela.ws>, (2012, January 6).
- [10] Harf Library, <http://www.harf.com>, (2012, January 6).
- [11] Dorar Site, <http://www.dorar.net/enc/Hadith>, (2012, January 6).
- [12] Yevgen Biletskiy, Olga Vorochek, "Building and Integrating Ontologies to Assist Decision Making", IEEE, December 14-17, 2004, Atlantis, Paradise Island, Bahamas.
- [13] L. Zhang, L. Gao, X. Y. Shao, "Application of Decision-making Ontology for Automotive Body Assembly Design", IEEE, 2009
- [14] Junyi Chai and James Liu, "AN Ontology-Driven Framework for Supporting Complex Decision Process", World Automation Congress, 2010, TSI Press.
- [15] Rosmayati Mohamad, Abdul Razak Hamdan, "Ontological-based for Supporting Multi Criteria Decision-Making", IEEE, 2010.
- [16] Z. H. Deng, S. W. Tang, M. Zhang, D. Q. Yang, and J.Chen, "Overview of Ontology", Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 38, 2002.
- [17] D. Fensel, "Ontologies: Silver Bullet for Knowledge Management and Electronic Commerc", Springer 2001.
- [18] HA Jinbing, Wei Youna, JIN Ying, "Logistics Decision-making Support System Based on Ontology", International Symposium on Computational Intelligence and Design, 2008.

- [19] Guarino N, "Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction and Integration", Springer Verlag, 1997, pp.139-170.
- [20] Matthew Horridge, Simon Jupp, Georgina Moulton, Alan Rector, Robert Stevens, Chris Wroe, "A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.1", 2007
- [21] Jeffrey T. "Pollock Semantic Web for dummies, Wiley Publishing", Inc., Indianapolis, Indiana, 2009, ISBN: 978-0-470-39679-7
- [22] Protege Software <http://protege.stanford.edu/>, (2012, October 1).
- [23] OWL API, <http://owlapi.sourceforge.net/>, (2012, October 1).
- [24] Protégé-OWL API, <http://protege.stanford.edu/plugins/owl/api/>, (2012, October 1).
- [25] DL Query, <http://protegewiki.stanford.edu/wiki/DLQueryTab>, (2012, October 1).
- [26] Boyce, Pahl, "Developing Domain Ontologies for Course Content", Educational Technology & Society, 10 (3), 275-288, (2007).
- [27] Uschold, M. and Gruninger, M. "Ontologies: Principles, Methods and Applications", Knowledge Engineering Review, 1996.
- [28] Rebekah Gilmour, "An ontology for Hazard Identification in Risk Management", Department of Chemical Engineering, 2004.
- [29] Aqil M. Azmi, and Nawaf Ibn Badia, "e-NARRATOR – AN Application for Creating an Ontology of Hadiths Narration Tree Semantically and Graphically", The Arabian Journal for Science and Engineering, Volume 35, Number 2C, 2010.
- [30] Leo Obrst, Benjamin Ashpole, Werner Ceusters, Inderjeet Mani, Steve Ray, Barry Smith, "the Evaluation of Ontologies Toward Improved Semantic Interoperability", State University of New York at Buffalo, US National Institute of Standards and Technology, MITRE, Bashpole, Inc.
- [31] Janez Brank, Marko Grobelnik, Dunja Mladenić, "A Survey of Ontology Evaluation Techniques", Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia.
- [32] Robert Porzel, Rainer Malaka, "A task-based approach for ontology evaluation", ECAI Workshop Ont, Learning and Population, 2004.

Arabic Bibliography

- [33] صالح الرفاعي، "عناية العلماء بالإسناد وعلم الجرح والتعديل وأثر ذلك في حفظ السنة النبوية"، مجمع الملك فهد لطباعة المصحف الشريف بالمدينة المنورة، 2004.
- [34] محمد الخشوعي، "التخريج ودراسة الأسانيد، الجمعية الشرعية"، 2002.
- [35] إسلام درباله، "تخريج الحديث النبوي عن طريق الموسوعات الإلكترونية" دار المعرفة، 2009.
- [36] محمد الطحان، "تيسير مصطلح الحديث"، مركز الهدى للدراسات، 1994.
- [37] ولاء التويجري، "اللاختلاط وأثره في الحكم على الراوي"، جامعة الملك سعود، 2005.
- [38] محمد عبد الغفار، "التدليس في الحديث"، <http://audio.islamweb.net/audio/index.php?page=FullContent&audioid=178711&full=1>، (2012, September 29).
- [39] ولاء التويجري، "مبحث في مسألة تعارض الوصل والإرسال"، جامعة الملك سعود، 2009.
- [40] إسماعيل رضوان، طالب أبو شعر، "منهج الحكم على الأسانيد"، مكتبة وطبعة دار المنارة، 2006.

- [41] صبحي الصالح، "علوم الحديث ومصطلحه عرض ودراسة"، المكتبة الوقفية، 2012.
- [42] النووي، صحيح مسلم، دار الفكر للطباعة والنشر والتوزيع، 1981.
- [43] محمد النيسابوري، "معرفة علوم الحديث"، دار الكتب العلمية - بيروت، 1977.
- [44] محمد القاسمي، "قواعد التحديث من فنون مصطلح الحديث"، مؤسسة الرسالة ناشرون، 1914.
- [45] أكرم العمري، "بحوث في تاريخ السنة المشرفة"، بساط - بيروت، 1975.
- [46] شمس السخاوي، "فتح المغيـث بشرح الفية الحديث للعراقي"، مكتبة السنة - مصر، 2003.
- [47] عبد الله الجرحاني، "الكامل في ضعفاء الرجال"، الكتب العلمية - بيروت، 1997.
- [48] عبد الرحمن المعلمي، "التنكيل بما في تأنيب الكوثري من الأباطيل"، المكتب الإسلامي، 1986.
- [49] أحمد العسقلاني، "النكت على كتاب ابن الصلاح"، عمادة البحث العلمي بالجامعة الإسلامية، المدينة المنورة، المملكة العربية السعودية، 1984.
- [50] برهان الدين البقاعي، "النكت الوفية بما في شرح الألفية"، مكتبة الرشد ناشرون، 2007.
- [51] أحمد العسقلاني، "تقريب التهذيب"، دار الرشيد - سوريا، 1986.
- [52] محمد التركي، "البيان والتبيين لضوابط ووسائل تمييز الرواة المهملين"، المكتبة الشاملة، 2011.

Appendices

Appendix A: IJS Analysis and Design Components

Class Diagram:

Figure A.1 depicts the class diagram of book database and illustrate their attributes, operations (or methods), and the relationships among the classes.

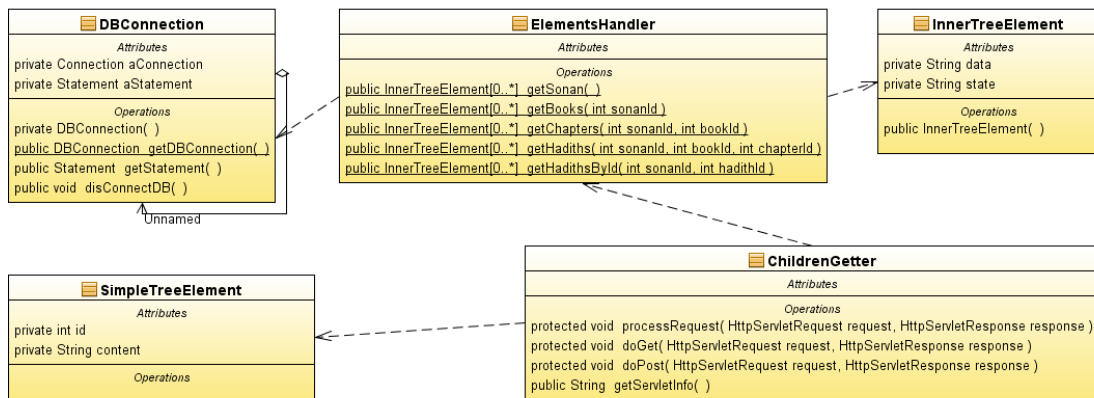


Figure A.1: Class Diagram of Book

Figure A.2 depicts the class diagram of IJS system by showing the system's and their attributes, operations (or methods), and the relationships among the classes.

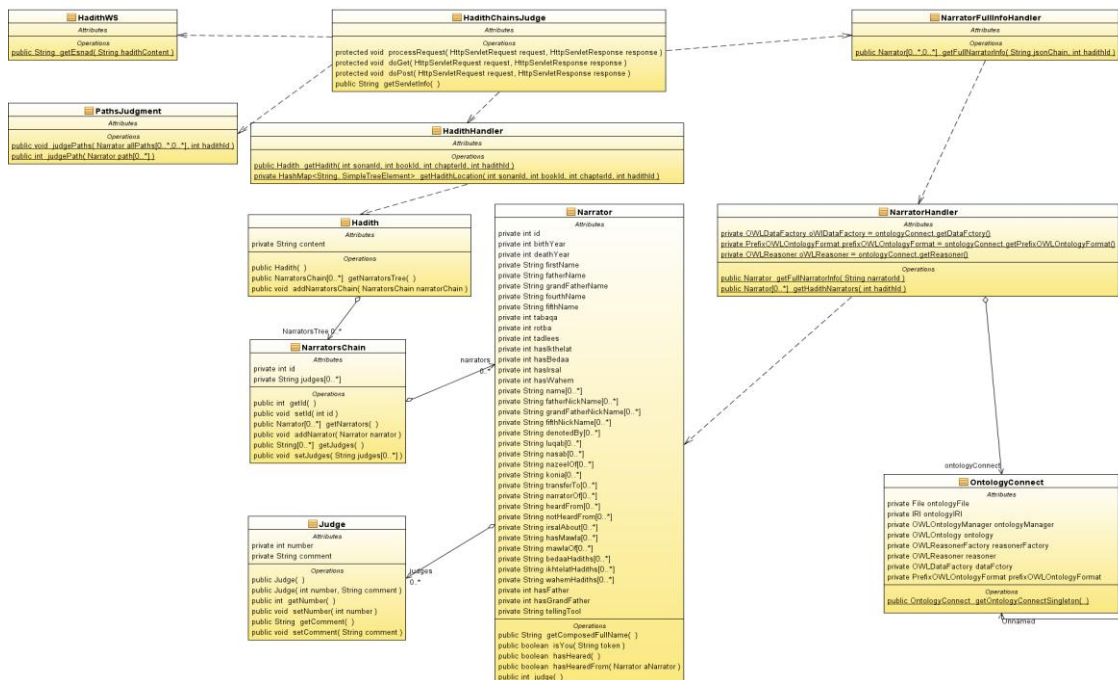


Figure A.2: Class Diagram of Narrator

Sequence Diagram:

It shows how processes operate with one another and in what order and shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case in IJS Prototype.

Usecase1: Figure A.3 displays the Hadith by exploring.

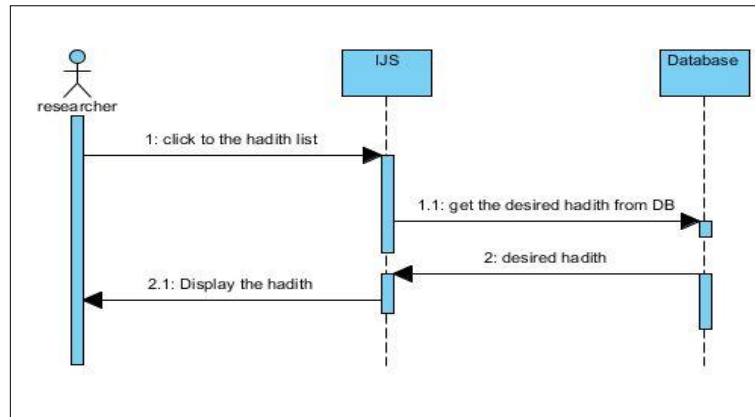


Figure A.3: Display the Hadith by Exploring.

Usecase2: Figure A.4 displays the Hadith by search

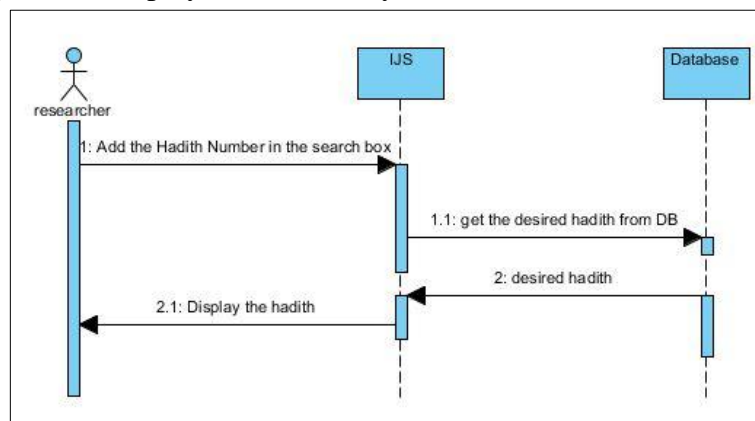


Figure A.4: Display the Hadith by Search.

Usecase3: Figure A.5 displays getting the judgment of Hadith.

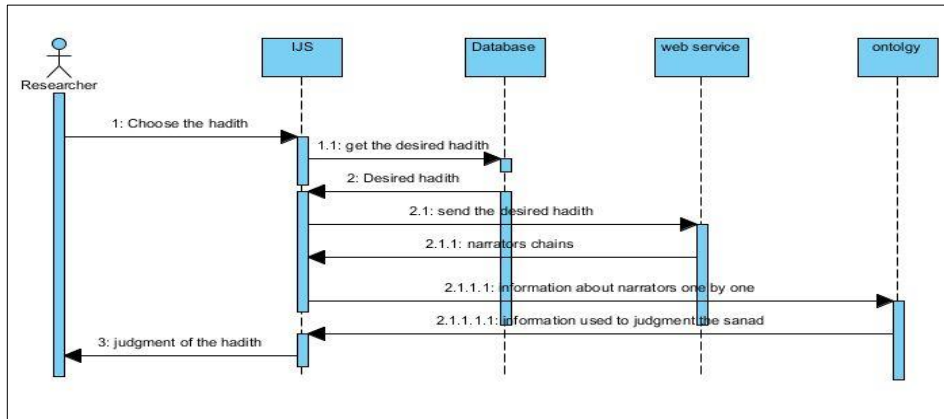


Figure A.5: Get the Judgment of Hadith.

Usecase4: Figure A.6 displays adding new narrator information.

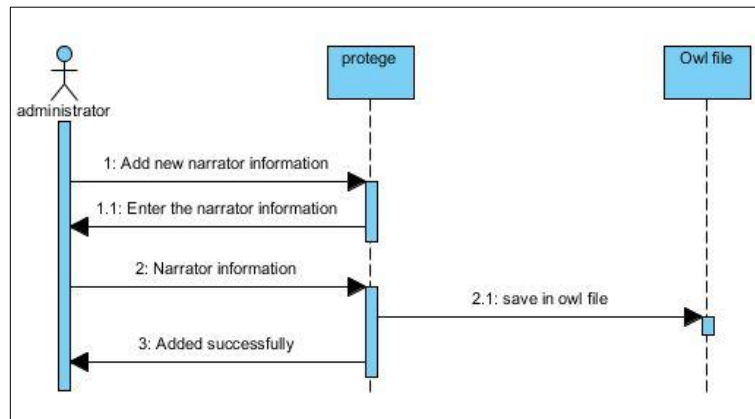


Figure A.6: Add New Narrator Information.

Usecase5: Figure A.7 displays updating the narrator information.

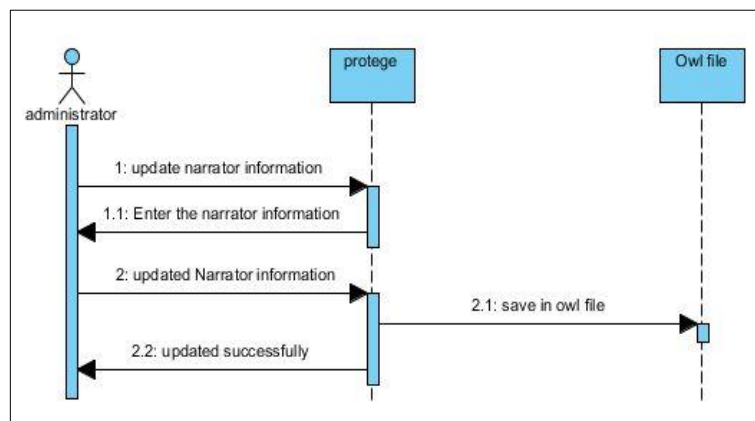


Figure A.7: Update Narrator Information.

Usecase6: Figure A.8 displays adding new book information.

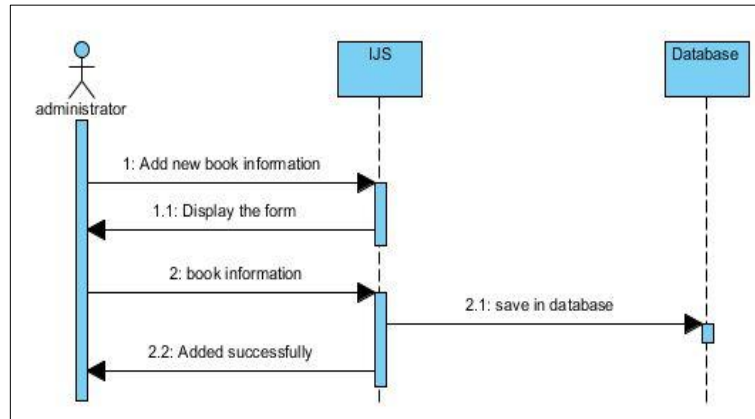


Figure A.8: Add New Book Information.

Usecase7: Figure A.9 displays updating the book information

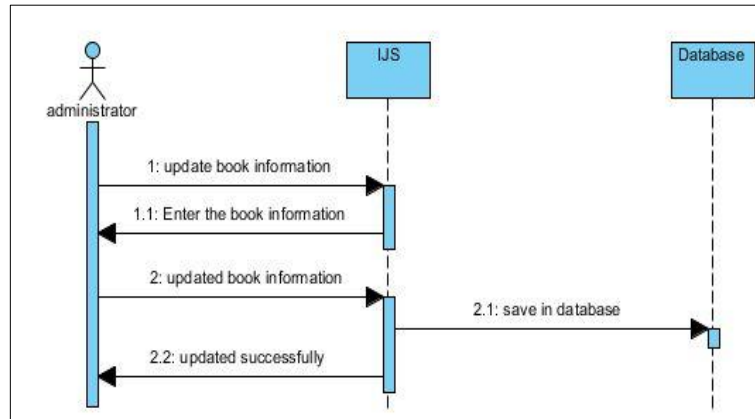


Figure A.9: Update Book Information

Usecase8: Figure A.10 displays adding new Hadith information.

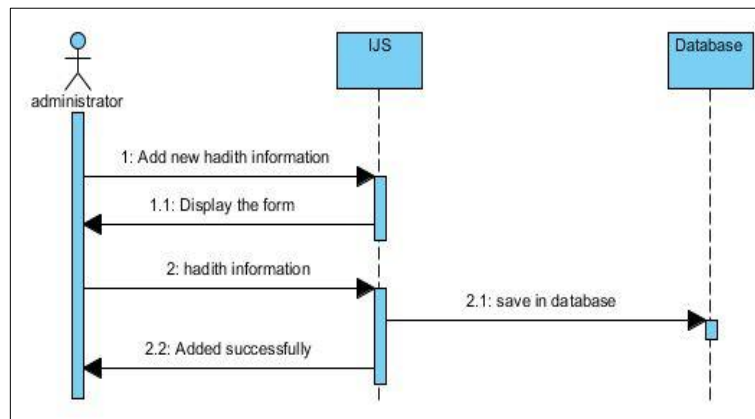


Figure A.10: Add New Hadith Information.

Usecase9: Figure A.11 displays updating the Hadith information.

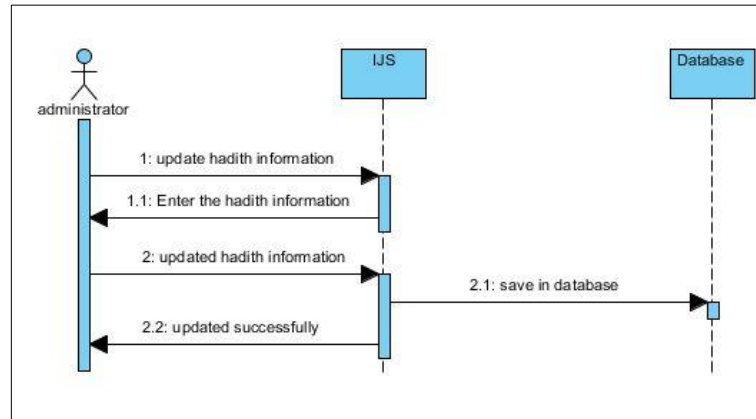


Figure A.11: Update Hadith Information

Appendix B: IJS Implementation

Listing B.1 depicts the code of main function that judges the Hadith chain.

```
@WebServlet(name = "HadithChainsJudge", urlPatterns = {"/HadithChainsJudge"})  
public class HadithChainsJudge extends HttpServlet {  
  
protected void processRequest(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    response.setContentType("text/html;charset=UTF-8");  
    PrintWriter out = response.getWriter();  
    try {  
        int sonanId = Integer.parseInt(request.getParameter("sonanId"));  
        int bookId = Integer.parseInt(request.getParameter("bookId"));  
        int chapterId = Integer.parseInt(request.getParameter("chapterId"));  
        int HadithId = Integer.parseInt(request.getParameter("HadithId"));  
        Hadith aHadith = HadithHandler.getHadith(sonanId, bookId, chapterId, HadithId);  
        String jsonChain = HadithWS.getEsnad(aHadith.getContent());  
        ArrayList<ArrayList<Narrator>> allPaths =  
NarratorFullInfoHandler.getFullNarratorInfo(jsonChain, HadithId);  
        PathsJudgment.judgePaths(allPaths, HadithId);  
        request.setAttribute("Hadith", aHadith);  
        request.setAttribute("AllPaths", allPaths);  
        request.getRequestDispatcher("/WEB-INF/judgment.jsp").forward(request, response);  
    } catch (ClassNotFoundException ex) {  
        out.write("DatabaseError");  
    } catch (SQLException ex) {  
        out.write("DatabasError");  
    } catch (ParserException ex) {  
        out.write("NoHadithInfo");  
    } catch (javax.xml.ws.WebServiceException ex) {  
        out.write("ErrorParsingEsnad");  
    } catch (java.net.SocketException ex) {  
        out.write("ErrorParsingEsnad");  
    } finally {  
        out.close();  
    }  
    }  
  
// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the  
left to edit the code.">  
protected void doGet(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    processRequest(request, response);  
    }  
  
protected void doPost(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    processRequest(request, response);  
    }  
  
public String getServletInfo() {  
    return "Short description";  
    }  
} // </editor-fold>  
}
```

Listing B.1: Hadith Chains Judge

Listing B.2 depicts the code that gets the Hadith from database.

```

public class HadithHandler {
    public static Hadith getHadith(int sonanId, int bookId, int chapterId, int HadithId) throws
    ClassNotFoundException, SQLException {
        DBConnection adBConnection = null;
        try { Hadith aHadith = null;
            adBConnection = DBConnection.getDBConnection();
            Statement asStatement = adBConnection.createStatement();
            String query = "Select content from Hadith where sonanId = " + sonanId + " and bookID
= " + bookId + " and ChapterID = " + chapterId + " and id = " + HadithId;
            ResultSet aResultSet = asStatement.executeQuery(query);
            while (aResultSet.next()) {
                aHadith = new Hadith();
                aHadith.setContent(aResultSet.getString("content"));
                HashMap<String, SimpleTreeElement> HadithLocation = getHadithLocation(sonanId,
bookId, chapterId, HadithId);
                aHadith.setHadithLocation(HadithLocation); }
            return aHadith;
        } finally { adBConnection.disconnectDB();}}
    private static HashMap<String, SimpleTreeElement> getHadithLocation(int sonanId, int bookId,
int chapterId, int HadithId) throws ClassNotFoundException, SQLException {
        DBConnection adBConnection = null;
        try { HashMap<String, SimpleTreeElement> HadithLocation = null;
            adBConnection = DBConnection.getDBConnection();
            Statement asStatement = adBConnection.createStatement();
            String query = "SELECT sonan.Name as sonan, book.name as book, chapter.name as
chapter, Hadith.content as Hadith FROM"
                + " sonan, chapter, book, Hadith "
                + "WHERE sonan.id = book.sonanid AND "
                + "chapter.bookid = book.id AND chapter.sonanid = sonan .id AND "
                + "Hadith.sonanid = sonan.id AND book.id = Hadith.bookid AND chapter.id =
Hadith.chapterid AND "
                + "sonan.id = " + sonanId + " AND book.id = " + bookId + " AND chapter.id = " +
chapterId + " AND Hadith.id = " + HadithId;
            ResultSet aResultSet = asStatement.executeQuery(query);
            if (aResultSet.next()) {
                HadithLocation = new HashMap<String, SimpleTreeElement>();
                HadithLocation.put("sonan", new SimpleTreeElement(sonanId,
aResultSet.getString("sonan")));
                HadithLocation.put("book", new SimpleTreeElement(bookId,
aResultSet.getString("book")));
                HadithLocation.put("chapter", new SimpleTreeElement(chapterId,
aResultSet.getString("chapter")));
                HadithLocation.put("Hadith", new SimpleTreeElement(HadithId,
aResultSet.getString("Hadith")));
            } return HadithLocation;
        } finally {
            adBConnection.disconnectDB();
        }
    }
}

```

Listing B.2: Hadith Handler

Listing B.3 depicts the code that Send the Hadith to the Web Service.

```
public class HadithWS {
    public static String getEsnad(String HadithContent) {
        ps.Hadith.Service service = new ps.Hadith.Service();
        ps.Hadith.ServiceSoap port = service.getServiceSoap();
        return port.getEsnad(HadithContent) } }
```

Listing B.3: Hadith Web Service Handler

Listing B.4 depicts the code that Handler Narrator Full Information.

```
public class NarratorFullInfoHandler {
    public static ArrayList<ArrayList<Narrator>> getFullNarratorInfo(String jsonChain, int
    HadithId) throws ParseException{
        Gson gsonObject = new Gson();
        Type type = new TypeToken<TempNarratorsChain[][]>() {
        }.getType();
        ArrayList<ArrayList<Narrator>> allPaths = new ArrayList<ArrayList<Narrator>>();
        TempNarratorsChain[][] HadithTempChain = gsonObject.fromJson(jsonChain, type);

        ArrayList<Narrator> HadithNarrators = NarratorHandler.getHadithNarrators(HadithId);
        Narrator lastNarrator = null;
        for (TempNarratorsChain[] narratorsChain : HadithTempChain) {
            ArrayList<Narrator> chain = new ArrayList<Narrator>();
            for (TempNarratorsChain chainNarrator : narratorsChain) {
                String chainName = chainNarrator.getName();
                if (chainName.equals("الاسناد")) {
                    continue; } chainName = chainName.replaceAll("ى", "$ي");
                chainName = chainName.replaceAll("ى", " ي ");
                if (chainName.indexOf("1- < ("أبيه) {
                    Narrator HadithNarrator =
                    NarratorHandler.getFullNarratorInfo(String.valueOf(lastNarrator.getHasFather()));
                    HadithNarrator.setTellingTool(chainNarrator.getTool());
                    chain.add(HadithNarrator);
                    lastNarrator = HadithNarrator;
                    continue;
                } else if (chainName.indexOf("1- < ("جده) {
                    Narrator HadithNarrator =
                    NarratorHandler.getFullNarratorInfo(String.valueOf(lastNarrator.getHasGrandFather()));
                    HadithNarrator.setTellingTool(chainNarrator.getTool());
                    chain.add(HadithNarrator);
                    lastNarrator = HadithNarrator;
                    continue; } for (Narrator HadithNarrator : HadithNarrators) {
                    if (HadithNarrator.isYou(chainName)) {
                        HadithNarrator.setTellingTool(chainNarrator.getTool());
                        chain.add(HadithNarrator);
                        lastNarrator = HadithNarrator;
                        break; }} }
                allPaths.add(chain);}
        return allPaths; }}
```

Listing B.4: Narrator Full Info Handler

Listing B.5 depicts the code that judges all paths of Hadith Isnad.

```

public class PathsJudgment {
    private static final String[] levels = {"", "الاولى", "الثانية", "الثالثة", "الرابعة", "الخامسة", "السادسة", "السابعة", "الثامنة عشر", "الثانية عشر", "الحادية عشر", "العاشرة", "التاسعة", "الثامنة"};
    private static final int[] goodJudgeNumber = {0, 0, 1, 1, 2};
    private static final int NEGATIVE_PROPERTY = -1;
    public static void judgePaths(ArrayList<ArrayList<Narrator>> allPaths, int HadithId) {
        Map<Integer, Narrator> checkedNarrators = new HashMap<Integer, Narrator>();
        String HadithIdString = String.valueOf(HadithId);
        for (int i = 0; i < allPaths.size(); i++) {
            for (int j = 0; j < allPaths.get(i).size(); j++) {
                Narrator currentNarrator = allPaths.get(i).get(j);
                if (checkedNarrators.containsKey(currentNarrator.hashCode())) {
                    continue; }
                int rotba = currentNarrator.getRotba();
                String judge;
                switch (rotba) {
                    case 1:
                        judge = "لأنه صحابي";
                        currentNarrator.getJudges().add(new Judge(1, judge));
                        break;
                    case 2: case 3: case 4:
                        judge = String.format("من الرتبة %s", levels[rotba]);
                        currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba], judge));
                        //Test Tadlees
                        int tadlees = currentNarrator.getTadlees();
                        if (tadlees == NEGATIVE_PROPERTY) {
                            judge = "لم يثبت عليه تدليس";
                            currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba],
                                judge)); } else if (tadlees == 1 || tadlees == 2) {
                            judge = String.format("مدلس من الدرجة %s", levels[tadlees]);
                            currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba],
                                judge)); } else if (tadlees == 3) { if (currentNarrator.hasHeard()) {
                            judge = String.format("%s' وصرح بالسماع بقوله %s مدلس من الدرجة %s", levels[tadlees],
                                currentNarrator.getTellingTool());
                            currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba],
                                judge)); } else { judge = String.format("لا يفيد التصريح '%s' وقوله %s مدلس من الدرجة %s", levels[tadlees],
                                currentNarrator.getTellingTool());
                            currentNarrator.getJudges().add(new Judge(3, judge)); }
                            } else { judge = String.format("مدلس من الدرجة %s", levels[tadlees]);
                            currentNarrator.getJudges().add(new Judge(3, judge)); }
                        //Test Ikhtelat
                        int hasIkhtelat = currentNarrator.getHasIkhtelat();
                        if (hasIkhtelat == NEGATIVE_PROPERTY) {
                            judge = "لم يخلط في أحاديثه من قبل";
                            currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba],
                                judge)); } else { boolean found = false;
                            for (String HadithIdtemp : currentNarrator.getIkhtelatHadiths()) {
                                if (HadithIdString.equals(HadithIdtemp)) {
                                    judge = "قد ثبت أنه اخلط في رواية هذا الحديث";
                                    currentNarrator.getJudges().add(new Judge(3, judge));
                                    found = true;
                                }
                            }
                        }
                }
            }
        }
    }
}

```



```

        break; } if (!found) {
        judge = "لم يثبت أنه اخلط في رواية هذا الحديث";
        currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba],
judge)); } } //Test Irsal
        int hasIrsal = currentNarrator.getHasIrsal();
        Narrator nextNarrator = null;
        if (j < allPaths.get(i).size() - 1) {
            nextNarrator = allPaths.get(i).get(j + 1); }
        if (hasIrsal == NEGATIVE_PROPERTY) {
            judge = "لم يثبت عليه الإرسال";
            currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba],
judge)); } else if (currentNarrator.hasHeardFrom(nextNarrator)) {
            judge = String.format("عرف عنه الإرسال ولكنه ثبت أنه سمع عن %s",
nextNarrator.getComposedFullName());
            currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba],
judge)); } else { judge = String.format("عرف عنه الإرسال ولم يثبت أنه سمع عن %s",
nextNarrator.getComposedFullName());
            currentNarrator.getJudges().add(new Judge(3, judge)); }
        //Test Bedaa
        int hasBedaa = currentNarrator.getHasBedaa();
        if (hasBedaa == NEGATIVE_PROPERTY) {
            judge = "لم تثبت عليه بدعة";
            currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba],
judge)); } else {boolean found = false;
            for (String HadithIdtemp : currentNarrator.getBedaaHadiths()) {
                System.out.println(HadithIdtemp);
                if (HadithIdString.equals(HadithIdtemp)) {
                    judge = "قد ثبت أن له بدع وهذا الحديث يدعم بدعته";
                    currentNarrator.getJudges().add(new Judge(3, judge));
                    found = true;
                    break; } }
            if (!found) { judge = "قد ثبت أن هذا الحديث لا يدعم بدعته";
                currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba],
judge)); } }
        //Test Wahem
        int hasWahem = currentNarrator.getHasWahem();
        if (hasWahem == NEGATIVE_PROPERTY) {
            judge = "لم تثبت له أوهام";
            currentNarrator.getJudges().add(new Judge(goodJudgeNumber[rotba],
judge)); } else { for (String HadithIdtemp : currentNarrator.getWahemHadiths()) {
            if (HadithIdString.equals(HadithIdtemp)) {
                judge = "قد ثبت أن الراوي له أوهام وهذا الحديث أحد أوهامه";
                currentNarrator.getJudges().add(new Judge(3, judge));
                break; } else { if (allPaths.size() == 1) {
                    judge = "قد ثبت أن الراوي له أوهام وقد تفرد بالرواية في هذا السند";
                    currentNarrator.getJudges().add(new Judge(3, judge));
                } else {for (int k = 0; k < allPaths.size(); k++) {
                    Narrator narrator = allPaths.get(k).get(j);
                    if (narrator.getId() == currentNarrator.getId()) {
                        judge = "قد ثبت أن الراوي له أوهام وقد تفرد بالرواية في هذا السند";
                        currentNarrator.getJudges().add(new Judge(3, judge));
                        break;
                    }
                }
            }
        }
    }

```

```

        } else {
            judge = " قد ثبت أن الراوي له أوهام ولكنه الحديث لا يدعم أوهامه ولم يتفرد بروايته في "
            currentNarrator.getJudges().add(new Judge(3, judge));
            }}}} } break;
        case 5:case 6:case 7:case 8:case 9:case 10:case 11: case 12:
            judge = String.format("من الرتبة %s", levels[rotba]);
            currentNarrator.getJudges().add(new Judge(3, judge));
            break; }
        checkedNarrators.put(currentNarrator.hashCode(), currentNarrator);
    } }
public static int judgePath(ArrayList<Narrator> path) {
    int level = -1;
    for (Narrator narrator : path) {
        if (level == 3) {
            break; }
        for (Judge judge : narrator.getJudges()) {
            level = Math.max(level, judge.getNumber());
            if (level == 3) {
                return 3; } } }
    return level;
}

```

Listing B.5: Paths Judgment

Appendix C: OWL Source Code

Figure C.1 depicts the OWL source code of IJS ontology.

```

<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <ENTITY ontologies "http://protege.org/ontologies/" >
  <ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <ENTITY Hadith "http://protege.org/ontologies/Hadith.owl#" >
  <ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF xmlns="&ontologies;Hadith.owl#"
  xml:base="&ontologies;Hadith.owl"
  xmlns:Hadith="&ontologies;Hadith.owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:ontologies="http://protege.org/ontologies/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about="&ontologies;Hadith.owl"/>
  <!--
  //////////////////////////////////////
  //
  // Datatypes
  //
  //////////////////////////////////////
  -->
  <!-- http://www.w3.org/2000/01/rdf-schema#Literal -->
  <rdfs:Datatype rdf:about="&rdfs;Literal"/>
  <!-- http://www.w3.org/2001/XMLSchema#integer -->
  <rdfs:Datatype rdf:about="&xsd;integer"/>
  <!--
  //////////////////////////////////////
  //
  // Object Properties
  //
  //////////////////////////////////////
  -->
  <!-- http://protege.org/ontologies/Hadith.owl#AuthorOf -->
  <owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#AuthorOf">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Author"/>
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Book"/>
    <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#hasAuthor"/>
  </owl:ObjectProperty>

  <!-- http://protege.org/ontologies/Hadith.owl#BedaaHadiths -->
  <owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#BedaaHadiths">
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Hadith"/>
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
  </owl:ObjectProperty>

  <!-- http://protege.org/ontologies/Hadith.owl#BrotherOf -->
  <owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#BrotherOf">

```

```

    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#ChapterOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#ChapterOf">
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Book"/>
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Chapter"/>
    <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#hasChapter"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#ContainerOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#ContainerOf">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Book"/>
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Hadith"/>
    <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#hasContainer"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#CousinOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#CousinOf">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#DaughterHusbandOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#DaughterHusbandOf">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#FatherOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#FatherOf">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#GrandFatherOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#GrandFatherOf">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Person"/>
    <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#hasGrandFather"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#GrandSonOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#GrandSonOf">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#HadithOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#HadithOf">
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Chapter"/>
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Hadith"/>
    <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#hasHadith"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#HeardFrom -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#HeardFrom">
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#TransferTo"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#IkhtelatHadiths -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#IkhtelatHadiths">
    <rdfs:range rdf:resource="&ontologies;Hadith.owl#Hadith"/>
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>

```

```

</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#IrsalAbout -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#IrsalAbout">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#MawlaOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#MawlaOf">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#hasMawla"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#MotherOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#MotherOf">
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#NarratorOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#NarratorOf">
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Hadith"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#hasNarrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#NephewOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#NephewOf">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#TransferTo -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#TransferTo">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#UncleOf -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#UncleOf">
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#WahemHadiths -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#WahemHadiths">
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasAuthor -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasAuthor">
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Author"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Book"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasBrother -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasBrother">
  <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#BrotherOf"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasChapter -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasChapter">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Book"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Chapter"/>

```

```

</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasContainer -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasContainer">
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Book"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Hadith"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasCousin -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasCousin">
  <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#CousinOf"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasDaughterHusband -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasDaughterHusband">
  <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#DaughterHusbandOf"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasFather -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasFather">
  <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#FatherOf"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasGrandFather -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasGrandFather">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasGrandSon -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasGrandSon">
  <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#GrandSonOf"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasHadith -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasHadith">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Chapter"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Hadith"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasMawla -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasMawla">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasMother -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasMother">
  <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#MotherOf"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasNarrator -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasNarrator">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Hadith"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>

```

```

<!-- http://protege.org/ontologies/Hadith.owl#hasNephew -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasNephew">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#NephewOf"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasUncle -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#hasUncle">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <owl:inverseOf rdf:resource="&ontologies;Hadith.owl#UncleOf"/>
</owl:ObjectProperty>
<!-- http://protege.org/ontologies/Hadith.owl#notHeardFrom -->
<owl:ObjectProperty rdf:about="&ontologies;Hadith.owl#notHeardFrom">
  <rdfs:range rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
</owl:ObjectProperty>
<!--
////////////////////////////////////
//
// Data properties
//
////////////////////////////////////
-->
<!-- http://protege.org/ontologies/Hadith.owl#AttributedToGrandfather -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#AttributedToGrandfather">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#BirthYear -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#BirthYear">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Content -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Content">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Hadith"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#DeathYear -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#DeathYear">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#DenotedBy -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#DenotedBy">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#FatherName -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#FatherName">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#FatherNickName -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#FatherNickName">

```

```

    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#FifthName -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#FifthName">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#FifthNickName -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#FifthNickName">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#FirstName -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#FirstName">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#FourthName -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#FourthName">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#FourthNickName -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#FourthNickName">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Gender -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Gender">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#GrandFatherName -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#GrandFatherName">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
    <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#GrandFatherNickName -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#GrandFatherNickName">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
    <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Id -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Id">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Konia -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Konia">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Liveln -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Liveln">

```



```

    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Luqab -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Luqab">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Name -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Name">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Nasab -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Nasab">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#NazeelOf -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#NazeelOf">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#NickName -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#NickName">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#NoOfPages -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#NoOfPages">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Book"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#OriginallyFrom -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#OriginallyFrom">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Rotba -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Rotba">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
    <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Tabaqa -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Tabaqa">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Tadlees -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Tadlees">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Title -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Title">
    <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Hadith"/>

```

```

    <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Topic -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Topic">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Hadith"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#Who -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#Who">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Person"/>
  <rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#YearOfPublication -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#YearOfPublication">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Book"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasBedaa -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#hasBedaa">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#haslthelat -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#haslthelat">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#haslrsal -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#haslrsal">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<!-- http://protege.org/ontologies/Hadith.owl#hasWahem -->
<owl:DatatypeProperty rdf:about="&ontologies;Hadith.owl#hasWahem">
  <rdfs:domain rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<!--
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
-->
<!-- http://protege.org/ontologies/Hadith.owl#Author -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Author">
  <rdfs:subClassOf rdf:resource="&ontologies;Hadith.owl#Person"/>
</owl:Class>
<!-- http://protege.org/ontologies/Hadith.owl#Book -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Book"/>
<!-- http://protege.org/ontologies/Hadith.owl#Chapter -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Chapter"/>
<!-- http://protege.org/ontologies/Hadith.owl#Hadith -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Hadith"/>
<!-- http://protege.org/ontologies/Hadith.owl#Narrator -->

```

```

<owl:Class rdf:about="&ontologies;Hadith.owl#Narrator">
  <rdfs:subClassOf rdf:resource="&ontologies;Hadith.owl#Person"/>
</owl:Class>
<!-- http://protege.org/ontologies/Hadith.owl#Person -->
<owl:Class rdf:about="&ontologies;Hadith.owl#Person"/>
<!--
////////////////////////////////////
//
// Individuals
//
////////////////////////////////////
-->
<!-- http://protege.org/ontologies/Hadith.owl#1125 -->
<owl:NamedIndividual rdf:about="&ontologies;Hadith.owl#1125">
  <rdf:type rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <hasWahem rdf:datatype="&xsd;integer">-1</hasWahem>
  <hasBedaa rdf:datatype="&xsd;integer">-1</hasBedaa>
  <Tadlees rdf:datatype="&xsd;integer">-1</Tadlees>
  <hasIkthelat rdf:datatype="&xsd;integer">-1</hasIkthelat>
  <hasIrsal rdf:datatype="&xsd;integer">-1</hasIrsal>
  <Id rdf:datatype="&xsd;integer">1125</Id>
  <Rotba rdf:datatype="&xsd;integer">4</Rotba>
  <Tabaqa rdf:datatype="&xsd;integer">7</Tabaqa>
  <Nasab><الواسطي/>Nasab
  <FatherName><دينار/>FatherName
  <Nasab><السلمي/>Nasab
  <Name><حجاج بن دينار/>Name
  <FirstName><حجاج/>FirstName
  <Nasab><الأشجعي/>Nasab
  <NarratorOf rdf:resource="&ontologies;Hadith.owl#48"/>
  <TransferTo rdf:resource="&ontologies;Hadith.owl#5756"/>
  <TransferTo rdf:resource="&ontologies;Hadith.owl#6227"/>
  <HeardFrom rdf:resource="&ontologies;Hadith.owl#8298"/>
</owl:NamedIndividual>
<!-- http://protege.org/ontologies/Hadith.owl#1156 -->
<owl:NamedIndividual rdf:about="&ontologies;Hadith.owl#1156">
  <rdf:type rdf:resource="&ontologies;Hadith.owl#Narrator"/>
  <hasIrsal rdf:datatype="&xsd;integer">-1</hasIrsal>
  <Tadlees rdf:datatype="&xsd;integer">-1</Tadlees>
  <hasWahem rdf:datatype="&xsd;integer">-1</hasWahem>
  <hasBedaa rdf:datatype="&xsd;integer">-1</hasBedaa>
  <hasIkthelat rdf:datatype="&xsd;integer">-1</hasIkthelat>
  <Tabaqa rdf:datatype="&xsd;integer">1</Tabaqa>
  <Rotba rdf:datatype="&xsd;integer">1</Rotba>
  <Id rdf:datatype="&xsd;integer">1156</Id>
  <DeathYear rdf:datatype="&xsd;integer">36</DeathYear>
  <FirstName><حذيفة/>FirstName
  <GrandFatherName><جابر/>GrandFatherName
  <FatherName><اليمان/>FatherName
  <Name><حذيفة بن اليمان/>Name
  <Nasab><العبيسي/>Nasab
  <Konia><أبو عبد الله العبيسي/>Konia
  <FatherName><حسل/>FatherName
  <FourthName><أسيد/>FourthName
  <Konia><أبو عبد الله/>Konia
  <FatherName><حسيل/>FatherName
  <TransferTo rdf:resource="&ontologies;Hadith.owl#1879"/>

```

```

    <NarratorOf rdf:resource="&ontologies;Hadith.owl#94"/>
  </owl:NamedIndividual>
  <!-- http://protege.org/ontologies/Hadith.owl#141 -->
  <owl:NamedIndividual rdf:about="&ontologies;Hadith.owl#141">
    <rdf:type rdf:resource="&ontologies;Hadith.owl#Hadith"/>
    <Content> حدثنا عبد الوهاب بن الضحاك قال: حدثنا إسماعيل بن عياش، عن صفوان بن عمرو، عن عبد الرحمن بن جبير بن
    نفيير، عن كثير بن مرة الحضرمي، عن عبد الله بن عمرو، قال: قال رسول الله صلى الله عليه وسلم: «إن الله اتخذني خليلاً كما اتخذ إبراهيم
    «خليلاً، فمنزلي ومنزل إبراهيم في الجنة يوم القيامة تجاهين، والعباس بيننا مؤمن بين خليلين
  </Content>
    <Topic>فضل العباس بن عبد المطلب رضي الله عنه</Topic>
    <hasNarrator rdf:resource="&ontologies;Hadith.owl#2938"/>
    <hasNarrator rdf:resource="&ontologies;Hadith.owl#3499"/>
    <hasNarrator rdf:resource="&ontologies;Hadith.owl#3827"/>
    <hasNarrator rdf:resource="&ontologies;Hadith.owl#4257"/>
    <hasNarrator rdf:resource="&ontologies;Hadith.owl#473"/>
    <hasNarrator rdf:resource="&ontologies;Hadith.owl#5631"/>
    <HadithOf rdf:resource="&ontologies;Hadith.owl#العلم_والصحابة_وفضائل_الإيمان_وفضائل_الصحابة_والعلم"/>
    <hasContainer rdf:resource="&ontologies;Hadith.owl#سنن_ابن_ماجه"/>
  </owl:NamedIndividual>

```

Figure C.1: OWL Source Code of IJS Ontology

Appendix D: Hadith Samples

4- حَدَّثَنَا مُحَمَّدُ بْنُ عَبْدِ اللَّهِ بْنِ نُمَيْرٍ حَدَّثَنَا زَكَرِيَّا بْنُ عَدِيٍّ عَنِ ابْنِ الْمُبَارَكِ عَنْ مُحَمَّدِ بْنِ سُوْقَةَ عَنْ أَبِي جَعْفَرٍ قَالَ كَانَ ابْنُ عُمَرَ إِذَا سَمِعَ مِنْ رَسُولِ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ حَدِيثًا لَمْ يَعْذُهُ وَلَمْ يَقْصُرْ ذُوْنَهُ.

It narrated that Abu Jaafar said, "*Whenever Ibn Umar heard a Hadiths from the Messenger of Allah he would not do more than it said, and he would not do less.*"

5- حَدَّثَنَا هِشَامُ بْنُ عَمَّارٍ الدَّمَشْقِيُّ حَدَّثَنَا مُحَمَّدُ بْنُ عِيْسَى بْنِ سَمِيعٍ حَدَّثَنَا إِبْرَاهِيمُ بْنُ سُلَيْمَانَ الْأَفْطَسُ عَنِ الْوَلِيدِ بْنِ عَبْدِ الرَّحْمَنِ الْجُرَشِيِّ عَنْ جُبَيْرِ بْنِ نُفَيْرٍ عَنْ أَبِي الدَّرْدَاءِ قَالَ خَرَجَ عَلَيْنَا رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ وَنَحْنُ نَذْكُرُ الْفَقْرَ وَنَتَخَوَّفُهُ فَقَالَ الْفَقْرُ تَخَافُونَ وَالَّذِي نَفْسِي بِيَدِهِ لَتُنْصَبَنَّ عَلَيْكُمْ الدُّنْيَا صَبًّا حَتَّى لَا يُرِيغَ قَلْبَ أَحَدِكُمْ إِزَاعَةً إِلَّا هِيَهُ وَإِيْمَ اللَّهِ لَقَدْ تَرَكْتُكُمْ عَلَى مِثْلِ الْبَيْضَاءِ لَيْلُهَا وَنَهَارُهَا سَوَاءٌ قَالَ أَبُو الدَّرْدَاءِ صَدَقَ وَاللَّهُ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ تَرَكْنَا وَاللَّهُ عَلَى مِثْلِ الْبَيْضَاءِ لَيْلُهَا وَنَهَارُهَا سَوَاءٌ.

It narrated that Abu Darda said, "*The Messenger of Allah came out to us when we were speaking of poverty and how we feared it. He said 'Is it poverty that you fear? By the One in whose hand is my soul (the delights and luxuries of) this world will come to you in plenty, and nothing will cause the heart of anyone of you to deviate except that'. By Allah, I am leaving you upon something like Bayda (white, bright, dear path) the night and day of which are the same.*"

6- حَدَّثَنَا مُحَمَّدُ بْنُ بَشَّارٍ حَدَّثَنَا مُحَمَّدُ بْنُ جَعْفَرٍ حَدَّثَنَا شُعْبَةُ عَنْ مُعَاوِيَةَ بْنِ قُرَّةَ عَنْ أَبِيهِ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ لَا تَزَالُ طَائِفَةٌ مِنْ أُمَّتِي مُنْصُورِينَ لَا يَضُرُّهُمْ مَنْ خَدَلَهُمْ حَتَّى تَقُومَ السَّاعَةُ.

Mu'awiyah bin Qurrah narrated that his father said, "*The Messenger of Allah group of my ummd will continue to prevail and they will never be harmed by those who forsake them, until the Hour begins.*"

21- حَدَّثَنَا عَلِيُّ بْنُ الْمُنْذِرِ حَدَّثَنَا مُحَمَّدُ بْنُ الْفُضَيْلِ حَدَّثَنَا الْمُقْبِرِيُّ عَنْ جَدِّهِ عَنْ أَبِي هُرَيْرَةَ عَنِ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ أَنَّهُ قَالَ لَا أَعْرِفَنَّ مَا يُحَدِّثُ أَحَدُكُمْ عَنِّي الْحَدِيثَ وَهُوَ مُتَكَيِّئٌ عَلَى أَرِيكَتِهِ فَيَقُولُ أَفْرَأُ قُرَأْنَا مَا قِيلَ مِنْ قَوْلِ حَسَنِ فَأَنَا قُلْتُهُ.

It narrated from Abu Hurairah that the prophet said "*I do not want to hear of anyone of you who, upon hearing a Hadith narrated from me, says while reclining on his pillow: 'Recite Quran (to verify this Hadith).'*" (Here the Prophet said) *Any excellent word that is said, it is I who have said it.*" [How then can you reject what I have said?]

23- حَدَّثَنَا أَبُو بَكْرِ بْنُ أَبِي شَيْبَةَ حَدَّثَنَا مُعَاذُ بْنُ مُعَاذٍ عَنِ ابْنِ عَوْنٍ حَدَّثَنَا مُسْلِمُ الْبَطِينُ عَنْ إِبْرَاهِيمَ النَّيْمِيِّ عَنْ أَبِيهِ عَنْ عَمْرِو بْنِ مَيْمُونٍ قَالَ مَا أَخْطَأَنِي ابْنُ مَسْعُودٍ عَشِيَّةَ حَمِيْسٍ إِلَّا أَنْبَيْتُهُ فِيهِ قَالَ فَمَا سَمِعْتُهُ يَقُولُ بِشَيْءٍ قَطُّ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ فَلَمَّا كَانَ ذَلِكَ عَشِيَّةً قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ فَتَنَكَّسَ قَالَ فَتَنَظَّرْتُ

إِلَيْهِ فَهُوَ قَائِمٌ مُحَلَّلَةٌ أَزْرَارٌ قَمِيصِهِ قَدْ اغْرُورَقَتْ عَيْنَاهُ وَانْتَفَخَتْ أَوْدَاجُهُ قَالَ أَوْ دُونَ ذَلِكَ أَوْ فَوْقَ ذَلِكَ أَوْ قَرِيبًا مِنْ ذَلِكَ أَوْ شَبِيهَا بِذَلِكَ.

'Amr bin Maimun said: "I used to visit Ibn Mas'ud every Thursday afternoon, but he never uttered the words: 'The Messenger of Allah said.' Then one evening, he said: 'The Messenger of Allah said,' then he let his head hang down." He said, "I looked at him and saw that his shirt was unfastened; his eyes were swelled with tears, and his veins were bulging out (with fear). He said 'Or more than that, or less than that, or dose to that or something similar."

30- حَدَّثَنَا أَبُو بَكْرِ بْنُ أَبِي شَيْبَةَ وَسُوَيْدُ بْنُ سَعِيدٍ وَعَبْدُ اللَّهِ بْنُ عَامِرِ بْنِ زُرَّارَةَ وَإِسْمَاعِيلُ بْنُ مُوسَى قَالُوا حَدَّثَنَا شَرِيكٌ عَنْ سِمَاكِ عَنْ عَبْدِ الرَّحْمَنِ بْنِ عَبْدِ اللَّهِ بْنِ مَسْعُودٍ عَنْ أَبِيهِ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ مَنْ كَذَبَ عَلَيَّ مُتَعَمِّدًا فَلْيَتَّبِعُوا مَعَهُ مِنَ النَّارِ.

It was narrated from 'Abdurrahman bin 'Abdullah bin Mas'ud that his father said: "The Messenger of Allah said 'Whoever tells lies about me deliberately, let him take his place in Hell."

48- حَدَّثَنَا عَلِيُّ بْنُ الْمُنْذِرِ حَدَّثَنَا مُحَمَّدُ بْنُ فَضِيلٍ ح وَحَدَّثَنَا حُوَيْرَةُ بْنُ مُحَمَّدٍ حَدَّثَنَا مُحَمَّدُ بْنُ بَشْرٍ قَالَ حَدَّثَنَا حَجَّاجُ بْنُ دِينَارٍ عَنْ أَبِي غَالِبٍ عَنْ أَبِي أُمَامَةَ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ مَا ضَلَّ قَوْمٌ بَعْدَ هُدَى كَانُوا عَلَيْهِ إِلَّا أَلُتُوا الْجَدَلَ ثُمَّ تَلَا هَذِهِ آيَةَ بَلْ هُمْ قَوْمٌ خَصِمُونَ الْآيَةَ.

It narrated that Abu Umamah said "The Messenger of Allah said: 'No people go astray after having followed right guidance, but those who indulge in disputes.' Then he recited this Verse: "Nay! But they are quarrelsome people."

54- حَدَّثَنَا مُحَمَّدُ بْنُ الْعَلَاءِ الْهَمْدَانِيُّ قَالَ حَدَّثَنِي رَشْدِيُّ بْنُ سَعْدٍ وَجَعْفَرُ بْنُ عَوْنٍ عَنْ ابْنِ أَنْعَمٍ هُوَ الْإِفْرِيقِيُّ عَنْ عَبْدِ الرَّحْمَنِ بْنِ رَافِعٍ عَنْ عَبْدِ اللَّهِ بْنِ عَمْرٍو قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ الْعِلْمُ ثَلَاثَةٌ فَمَا وَرَاءَ ذَلِكَ فَهُوَ فَضْلٌ آيَةٌ مُحْكَمَةٌ أَوْ سُنَّةٌ قَائِمَةٌ أَوْ قَرِيضَةٌ عَادِلَةٌ.

It was narrated that 'Abdullah bin Amr said: "The Messenger of Allah said: 'Knowledge is based on three things, and anything beyond that is superfluous: a clear Verse, an established Sunnah, or the rulings by which the inheritance is divided fairly."

57- حَدَّثَنَا عَلِيُّ بْنُ مُحَمَّدٍ الطَّنَافِسِيُّ حَدَّثَنَا وَكَيْعٌ حَدَّثَنَا سُفْيَانُ عَنْ سُهَيْلِ بْنِ أَبِي صَالِحٍ عَنْ عَبْدِ اللَّهِ بْنِ دِينَارٍ عَنْ أَبِي صَالِحٍ عَنْ أَبِي هُرَيْرَةَ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ الْإِيمَانُ بَضْعٌ وَسِتُّونَ أَوْ سَبْعُونَ بَابًا أَدْنَاهَا إِمَاطَةُ الْأَذَى عَنِ الطَّرِيقِ وَأَرْفَعُهَا قَوْلُ لَا إِلَهَ إِلَّا اللَّهُ وَالْحَيَاءُ شُعْبَةٌ مِنَ الْإِيمَانِ.

It was narrated that Abu Hurairah said: "The Messenger of Allah said: 'Faith has sixty-some or seventy parts, the least of which is to remove a harmful thing from the road the greatest of which is to say La ilaha illallah (none has the right to be worshipped but Allah).and modesty is a branch of faith."

85- حَدَّثَنَا عَلِيُّ بْنُ مُحَمَّدٍ حَدَّثَنَا أَبُو مُعَاوِيَةَ حَدَّثَنَا دَاوُدُ بْنُ أَبِي هِنْدٍ عَنْ عَمْرِو بْنِ شُعَيْبٍ عَنْ أَبِيهِ عَنْ جَدِّهِ قَالَ خَرَجَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ عَلَى أَصْحَابِهِ وَهُمْ يَخْتَصِمُونَ فِي الْقَدْرِ فَكَانَ مَا يُفْقَأُ فِي وَجْهِهِ حَبُّ الرُّمَّانِ مِنَ الْعَضْبِ فَقَالَ بِهَذَا أَمِرْتُمْ أَوْ لِهَذَا خُلِقْتُمْ تَضْرِبُونَ الْقُرْآنَ بَعْضَهُ بِنَعْضِ بَهَذَا هَلَكَتِ الْأُمَّةُ قَبْلَكُمْ قَالَ فَقَالَ عَبْدُ اللَّهِ بْنُ عَمْرٍو مَا عَبَطْتُ نَفْسِي بِمَجْلِسٍ تَخَلَّفْتُ فِيهِ عَنْ رَسُولِ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ مَا عَبَطْتُ نَفْسِي بِذَلِكَ الْمَجْلِسِ وَتَخَلَّفِي عَنْهُ.

Amr bin Shu'aib narrated from his father that his grandfather said: "The Messenger of Allah came out to his Companions when they were disputing about the Divine Decree, and it was as if pomegranate seeds had burst on his face (i.e., it turned red) because of anger. He said 'Have you been commanded to do this, or were you created for this purpose? You are using one part of the Quran against another part, and this is what led to the doom of the nations who came before you.'" 'Abdulla bin Amr said: "I was never so happy to have missed a gathering with the Messenger of Allah as I was to have missed that gathering".

92- حَدَّثَنَا مُحَمَّدُ بْنُ الْمُصَفَّى الْحَمَصِيُّ حَدَّثَنَا بَقِيَّةُ بْنُ الْوَلِيدِ عَنْ الْأَوْزَاعِيِّ عَنْ ابْنِ جُرَيْجٍ عَنْ أَبِي الزُّبَيْرِ عَنْ جَابِرِ بْنِ عَبْدِ اللَّهِ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ إِنَّ مَجُوسَ هَذِهِ الْأُمَّةِ الْمُكَذِّبُونَ بِأَقْدَارِ اللَّهِ إِنْ مَرَضُوا فَلَا تَعُودُوا لَهُمْ وَإِنْ مَاتُوا فَلَا تَسْهَدُوا لَهُمْ وَإِنْ لَقَيْتُمُوهُمْ فَلَا تُسَلِّمُوا عَلَيْهِمْ.

It narrated that Jabir bin Abdullah said: "The Messenger of Allah said: 'The Magians of this Ummah are those who deny the decrees of Allah. If they fall sick, do not visit them; if they die, do not attend their funerals; and if you meet them, do not greet them with Salim'".

94- حَدَّثَنَا عَلِيُّ بْنُ مُحَمَّدٍ حَدَّثَنَا وَكَيْعُ ح وَحَدَّثَنَا مُحَمَّدُ بْنُ بَشَّارٍ حَدَّثَنَا مُؤَمَّلٌ قَالَ حَدَّثَنَا سُفْيَانُ عَنْ عَبْدِ الْمَلِكِ بْنِ عُمَيْرٍ عَنْ مَوْلَى لِرَبْعِيِّ بْنِ جِرَاشٍ عَنْ رَبْعِيِّ بْنِ جِرَاشٍ عَنْ حَدِيفَةَ بْنِ الْيَمَانِ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ إِنِّي لَا أَدْرِي مَا قَدْرُ بَقَائِي فِيكُمْ فَاقْتَدُوا بِاللَّذِينَ مِنْ بَعْدِي وَأَشَارَ إِلَى أَبِي بَكْرٍ وَعُمَرَ.

It was narrated that Hudhaifah bin Yaman said: "The Messenger of Allah said: 'I do not know how long I will stay among you, so follow the example of these two after I am gone, and he pointed to Abu Bakr and Umar.'"

141- حَدَّثَنَا عَبْدُ الْوَهَّابِ بْنُ الضَّحَّاكِ حَدَّثَنَا إِسْمَاعِيلُ بْنُ عِيَّاشٍ عَنْ صَفْوَانَ بْنِ عَمْرٍو عَنْ عَبْدِ الرَّحْمَنِ بْنِ جُبَيْرِ بْنِ نُفَيْرٍ عَنْ كَثِيرِ بْنِ مَرَّةٍ الْحَضْرَمِيِّ عَنْ عَبْدِ اللَّهِ بْنِ عَمْرٍو قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ إِنَّ اللَّهَ اتَّخَذَنِي خَلِيلًا كَمَا اتَّخَذَ إِبْرَاهِيمَ خَلِيلًا فَمَنْزِلِي وَمَنْزِلُ إِبْرَاهِيمَ فِي الْجَنَّةِ يَوْمَ الْقِيَامَةِ تُجَاهَيْنِ وَالْعَبَّاسُ بَيْنَنَا مَوْمِنٌ بَيْنَ خَلِيلَيْنِ.

It narrated that 'Abdullah bin Amr said: "The Messenger of Allah said: 'Allah has taken me as a close friend (Khalil) as He took Ibrahim as a close friend. Therefore, my house and the house of Ibrahim will be opposite to one another on the Day of Resurrection. And Abbas will be in between us, a believer between two dose friends.'"

493- حَدَّثَنَا مُحَمَّدُ بْنُ عَبْدِ الْمَلِكِ بْنِ أَبِي الشَّوَّارِبِ حَدَّثَنَا عَبْدُ الْعَزِيزِ بْنُ الْمُخْتَارِ حَدَّثَنَا سُهَيْلٌ عَنْ أَبِيهِ عَنْ أَبِي هُرَيْرَةَ أَنَّ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ أَكَلَ كَنْفَ شَاةٍ فَمَضْمَضَ وَغَسَلَ يَدَيْهِ وَصَلَّى.

It narrated from Abu Hurairah that the Messenger of Allah ate meat from the shoulder of a sheep, then he rinsed his mouth and washed his hands, then he prayed.

595- حَدَّثَنَا هِشَامُ بْنُ عَمَّارٍ حَدَّثَنَا إِسْمَاعِيلُ بْنُ عَيَّاشٍ حَدَّثَنَا مُوسَى بْنُ عُقْبَةَ عَنْ نَافِعٍ عَنْ ابْنِ عُمَرَ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ لَا يَفْرَأُ الْقُرْآنَ الْجُنُبُ وَلَا الْحَائِضُ.

It narrated that Ibn 'Umar said " *The Messenger of Allah said: 'No one who is sexually impure and no woman who is menstruating should recite Quran.'*"

Appendix E: System Testing

Testing the User Web Interface

We write a JUnit test case using NetBeans IDE to ensure that the code of `getHadiths`, `getBooks`, `getSonan`, `getHadithsById` and `getChapters`, functions is working as intended as shown in the Figure 6.15.

```

64      * Test of getHadiths method, of class ElementsHandler.
65      */
66      @Test
67      public void testGetHadiths() throws Exception {
68          System.out.println("getHadiths");
69          int sonanId = 1198;
70          int bookId = 863;
71          int chapterId = 905;
72          String expectedResult = "[{"attr":{"id":"1198_863_905_699"},"data":"أَنَّ النَّبِيَّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ: «أَدْرَكَ مِنَ النَّبِيِّ رَجُلًا فَبَدَأَ بِأَقْدَارِهَا، حَيْثُومًا وَحَرْعًا، خَلُومًا وَفَلَمًا»}";
73          ArrayList result = ElementsHandler.getHadiths(sonanId, bookId, chapterId);
74          String jsonResult = gson.toJson(result);
75          assertEquals(expResult, jsonResult);
76      }
77      /**
78      * Test of getHadithsById method, of class ElementsHandler.
79      */
80      @Test
81      public void testGetHadithsById() throws Exception {
82          System.out.println("getHadithsById");
83          int sonanId = 1198;
84          int hadithId = 87;
85          String expectedResult = "[{"attr":{"id":"1198_87"},"data":"وَأَنَّ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ بَدَأَ بِأَقْدَارِهَا، حَيْثُومًا وَحَرْعًا، خَلُومًا وَفَلَمًا"}";
86          ArrayList result = ElementsHandler.getHadithsById(sonanId, hadithId);
87          String jsonResult = gson.toJson(result);
88          assertEquals(expResult, jsonResult);
89      }

```

Figure E.1: Code of JUnit Test for User Web Interface

When we run a JUnit test, the results displayed in the Test Results window of the IDE showing that these functions work as intended as shown in Figure 6.16.

The screenshot shows the NetBeans IDE interface. The top pane displays the source code for `ElementsHandlerTest.java`, which includes two JUnit test methods: `testGetHadiths()` and `testGetHadithsById()`. The bottom pane shows the Test Results window, indicating that all 5 tests passed successfully. The test results are as follows:

Test Method	Duration (s)	Status
testGetBooks	0.826	passed
testGetSonan	0.063	passed
testGetHadithsById	0.015	passed
testGetChapters	0.016	passed
testGetHadiths	0.031	passed

Figure E.2: JUnit Test Results Window of User Web Interface

Testing the e-Narrator Web Service

We test the main function of this.Net Web Service that is analyzing the Hadith to extract the narrators' chain. This subsystem gets Hadith data in JSON string and returns the narrators' chain in a JSON string. We write a JUnit test case using NetBeans IDE to ensure that the code of GetEsnad function is working as intended as shown in Figure 6.17.

```

28 public void setUp() {
29 }
30
31 @After
32 public void tearDown() {
33 }
34
35 /**
36  * Test of getEsnad method, of class HadithWS.
37  */
38 @Test
39 public void testGetEsnad() {
40     System.out.println("getEsnad");
41     String hadithContent = "4 - كَانَ ابْنُ عُمَرَ «إِذَا سَمِعَ مِنْ رَسُولِ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ خَيْرًا لَمْ يَغْدَهُ، وَلَمْ يَغْمَرْ ذُوئَهُ - 4»
42     String hadithContent2 = "2 - وَإِذَا تَهَيَّأَ عَنْ شَيْءٍ فَأَتَتْهُمَا - 2»
43     String expectedResult = "[[{"name": "\u0627\u0644\u0627\u0633\u0646\u0627\u062f", "tool": ""}, {"name": "\u0627\u0645\u0645\u062f \u0628\u0646 \u0639\u0628\u062f \u0627\u0644\u0644\u0647 \u0628\u0646 \u0646\u0645\u064a\u0631", "tool": "\u0627\u062d\u062f\u0627\u062f"}]]\n"
44     String expectedResult2 = "[[{"name": "\u0627\u0644\u0627\u0633\u0646\u0627\u062f", "tool": ""}, {"name": "\u0627\u0645\u0645\u062f \u0628\u0646 \u0639\u0628\u062f \u0627\u0644\u0644\u0647 \u0628\u0646 \u0646\u0645\u064a\u0631", "tool": "\u0627\u062f\u0627\u062f"}, {"name": "\u0627\u0645\u0645\u062f \u0628\u0646 \u0639\u0628\u062f \u0627\u0644\u0644\u0647 \u0628\u0646 \u0646\u0645\u064a\u0631", "tool": "\u0627\u062f\u0627\u062f"}]]\n"
45     String result = HadithWS.getEsnad(hadithContent);
46     String result2 = HadithWS.getEsnad(hadithContent2);
47     assertEquals(expResult, result);
48     assertEquals(expResult2, result2);
49 }
50

```

Figure E.3: Code of JUnit Test for e-Narrator Web Service Function

When we run a JUnit test, the results displayed in the Test Results window of the IDE showing that the GetEsnad function works as intended as shown in the Figure 6.18.

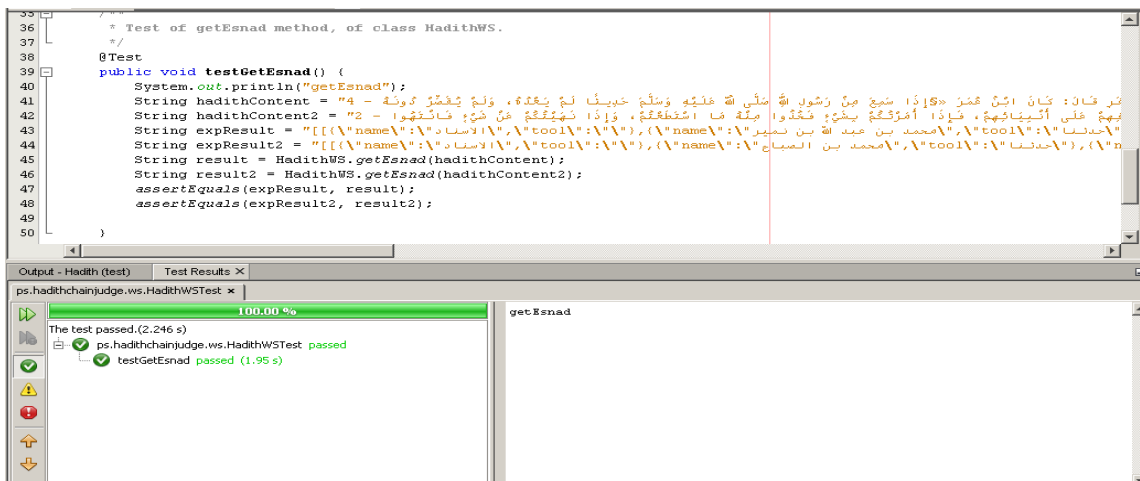


Figure E.4: JUnit Test Results Window of e-Narrator Web Service

Testing the Judgment System

The main function of this component is to produce the Hadith Isnad judgment based on the IJS ontology and HIJ algorithm. The component gets narrators' chain as a JSON string from the e-Narrator Web Service, and then by using the DL-Query gets all narrator information from IJS ontology. Using HIJ algorithm the judgment produced. To ensure the code of judgePaths and judgePath functions is working as intended, we write a JUnit test case using NetBeans IDE as shown in the Figure 6.19.

```

27  /**
28   * Test of judgePaths method, of class PathsJudgment.
29   */
30  @Test
31  public void testJudgePaths() throws ParseException {
32      System.out.println("judgePaths");
33      int hadithId = 141;
34      String jsonChain = "[[{\"name\":\"الاسناد\",\"tool\":\"\"},{\"name\":\"عبد الوهاب بن الحجاج\",\"tool\":\"احدنا\"}]";
35      ArrayList<ArrayList<Narrator>> allPaths = NarratorFullInfoHandler.getFullNarratorInfo(jsonChain, hadithId);
36      String expectedResult = "[{\"id\":4257,\"birthYear\":0,\"deathYear\":245,\"firstName\":\"عبد الوهاب\",\"fatherName\":\"\"}]";
37      PathsJudgment.judgePaths(allPaths, hadithId);
38      String jsonResult = gson.toJson(allPaths);
39      assertEquals(expResult, jsonResult);
40  }
41
42
43  /**
44   * Test of judgePath method, of class PathsJudgment.
45   */
46  @Test
47  public void testJudgePath() throws ParseException {
48      System.out.println("judgePath");
49      int hadithId = 141;
50      String jsonChain = "[[{\"name\":\"الاسناد\",\"tool\":\"\"},{\"name\":\"عبد الوهاب بن الحجاج\",\"tool\":\"احدنا\"}]";
51      ArrayList<ArrayList<Narrator>> allPaths = NarratorFullInfoHandler.getFullNarratorInfo(jsonChain, hadithId);
52      ArrayList<Narrator> path = allPaths.get(0);
53      int expectedResult = -1;
54      int result = PathsJudgment.judgePath(path);
55      assertEquals(expResult, result);
56  }

```

Figure E.5: Code of JUnit Test for Isnad Judgment System Function

When we run this JUnit test, the results displayed in the Test Results window of the IDE proving that judgePath and judgePaths functions works as intended as shown in Figure 4.15.

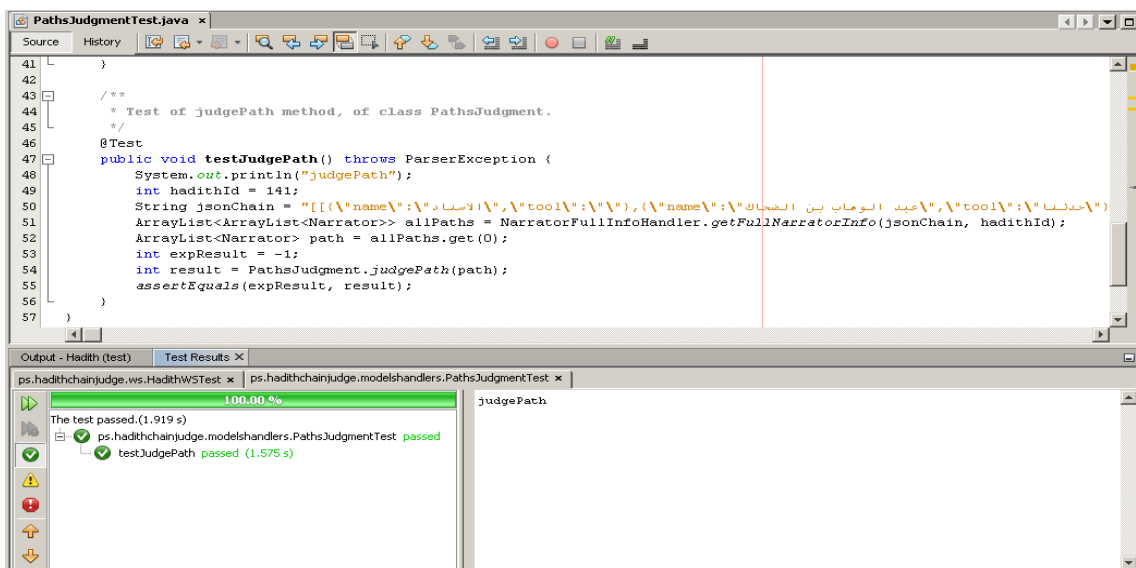


Figure E.6: JUnit Test Results Window of Isnad Judgment System