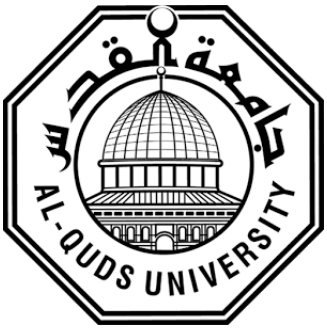
Deanship of Graduate Studies

Al-Quds University



Standard Based Exchange of Learning Objects:

Towards Outcome Based Learning

Ahmad Ali Mohammad Shukr

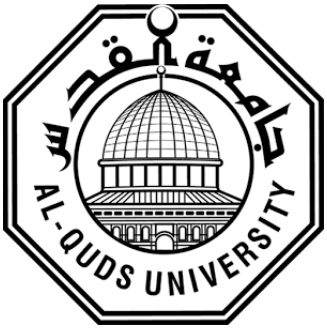
M.Sc. Thesis

Jerusalem – Palestine

1438 / 2017

Deanship of Graduate Studies

Al-Quds University



Standard Based Exchange of Learning Objects: Towards Outcome Based Learning

Ahmad Ali Mohammad Shukr

M.Sc. Thesis

Jerusalem – Palestine

1438 / 2017

Standard Based Exchange of Learning Objects:

Towards Outcome Based Learning

Prepared By:

Ahmad Ali Mohammad Shukr

B.Sc: Computer Science Al-Quds University Palestine

Supervisor: Dr. Jad Najjar

Co-supervisor: Dr. Badie Sartawi

A Thesis submitted in Partial fulfillment of requirements for the degree of Master of Computer Science – Al-Quds University.

1438 / 2017

Al-Quds University

Master of Computer Science

Computer Science & Information Technology

Thesis Approval

Standard Based Exchange of Learning Objects: Towards Outcome Based Learning

Prepared By: Ahmad Ali Shukr
Registration No. 21120124

Supervisor: Dr. Jad Najjar
Co-supervisor: Dr. Badie Sartawi

Master thesis submitted and accepted, Date: 1/12/2016

The names and signatures of the examining committee members are as follows:

1- Head of Committee (Supervisor): Dr.Jad Najjar .....

2- Co-Supervisor: Dr.Badie Sartawi .....

3- Internal Examiner: Dr. Rashid Jayousi .....

4- External Examiner: Dr. Ahmad Ewais .....

Jerusalem – Palestine

1438 / 2017

بسم الله الرحمن الرحيم

وَالْعَصْرِ (2) إِنَّ الإِنسَانَ لَفِي خُسْرٍ (3) إِلاَّ الَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ وَتَوَاصَوْا بِالْحَقِّ وَتَوَاصَوْا بِالصَّبْرِ

سورة العصر

In the name of Allah, the Beneficent, the Merciful

By time, Indeed, mankind is in loss, Except for those who have believed and done righteous deeds and advised each other to truth and advised each other to patience. –

AL-'Asr (The Declining Day) - The Holy Quran

Dedication

To my Mother, Mother, Mother
To my Father
To my Wife
To my Son Ali & future sons
To my Sisters
To my Brothers

Declaration

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed: .....

Ahmad Ali Shukr

Date: 15 January 2017

Acknowledgment

I would first like to thank my thesis advisors Dr. Jad Najjar and Dr. Badie Sartawi for the help, guidance and support throughout thesis project. I also would like to thank the members of the Computer Science Faculty at Al-Quds University.

Thanks for other master program members, Dr. Rashid Jayousi, Dr. Nedal Kafri, Dr. Raid Zaghal, and Dr. Wael Hassouneh. This thesis is a result of knowledge I grasped from all master courses.

Special appreciation to the external examiner Dr. Ahmad Ewais, and the internal examiner Rashid Jayousi. Thank you for your time.

Not to forget my mac laptop and every cup of coffee helped me to stay awake.

Finally, I must express my very profound gratitude to my parents and to my wife and my family for providing me with unfailing support and continuous encouragement throughout the years of my study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Abstract

There are a huge number of learning objects available on the web, still it is not easy to determine and select what to learn at the next level to master knowledge or a skill. Some techniques were introduced for recommending learning objects to learners, and one of these techniques is to use learning outcomes as basis for the recommendation. This means that we have to store learning objects and capture a link with learning outcomes, to recommend learning objects based on learning outcomes effectively.

In this study, we designed a software system to link learning objects with learning outcomes, and we implemented a learning objects repository linked with learning outcomes repository. We then validate our assumption by importing real learning objects and link them with real learning outcomes from different sources. To proof the concept, we conducted an experiment on a group of teachers and students to measure the usefulness of the system for actual users, also we show auto linking between learning objects and learning outcomes results to expert to determine results accuracy.

التبادل القياسي للمواد التعليمية :نحو تعليم مبني على مخرجات التعلم

اسم الطالب :احمد علي شكر

المشرف الرئيسي :د.جهاد النجار

المشرف المساعد :د.بديع السرطاوي

الملخص

هنالك عدد كبير من المواد التعليمية متوفرة على شبكة الانترنت، وحتى هذه اللحظة ليس من السهل تحديد ما الذي يجب تعلمه، هناك بعض الطرق تم ايجادها لعرض المواد التعليمية التي تهم المتعلمين، ومن هذه الطرق عرض المواد التعليمية بناءً على مخرجات التعليم. ولتحقيق هذا الهدف يجب في البداية ان نقوم بربط مخرجات التعليم والمواد التعليمية.

في هذه الدراسة، قمنا بتصميم نظام برمجي يقوم بربط مخرجات التعليم والمواد التعليمية، وقمنا بتطبيق ربط بين أماكن تخزين المواد التعليمية ومخرجات التعليم. و قمنا بالتحقق من صحة فرضياتنا من خلال إدخال مواد تعليمية حقيقة وربطها بمخرجات تعلم حقيقية من مصادر مختلفة. وقمنا بعقد تجربة على مجموعة من الطلاب والأساتذة لقياس حجم الاستفادة من خلال مستخدمين حقيقين و قمنا بعرض نتائج الربط الأوتوماتيكي بين مخرجات التعليم والمواد التعليمية على خبير تعليمي بهدف تقييم دقة الربط الاوتوماتيكي.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Full Name
XML	eXtensible Markup Language
PALO	Personal Achieved Learning Outcomes
ILT	Interactive Learning Technology
JSON	JavaScript Object Notation
QLearn	Quality Learn
UNESCO	The United Nations Educational, Scientific and Cultural Organization
ACM	Association for Computing Machinery
IEEE	The Institute of Electrical and Electronics Engineers
LOM	Learning Object Metadata
PDF	Portable Document Format
CEN/ISSS	The European Committee for Standardization/Information Society Standardization System
LOD	Learning Outcome Definitions
AQU	Al-Quds University
RDBMS	Relational database management system
MYSQL	Database Engine
IMS GLC	The IMS Global Learning Consortium
API	Application Programming Interface
IEEE RCD IEEE	Reusable Competency Definitions (RCD
XSLT	Extensible Stylesheet Language Transformations
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
DNS	Domain Name System
URI	Uniform Resource Identifier
CRUD	Create Read Update Delete
AJAX	Asynchronous JavaScript and XML

Chapter One

Introduction

In this chapter, we will introduce what this work is about, learning objects, learning outcomes, and the problem that this research is trying to solve.

1.1 Introduction

World Wide Web (WEB) was invented in 1989 [31], since that time, we started to use it for sharing all types of content such as books, presentations, videos, images, and scientific papers, etc., and this open new possibilities for learning, mastering new skills, and sharing information. For example, through WEB you can learn by yourself programming, mathematics, physics, and other practical things such as cooking, connecting cables or any other thing you have ever imagined and for free.

Because of the huge number of learning material available throughout the WEB (see Figure 1), it became harder to find the most appropriate material, so that new algorithms, methods and standards were invented to recommend materials for users based on their browsing history or preferences, also several standards were proposed to store and exchange different types of digital materials.

Learning objects term is used to describe different types of learning material, it was defined as “Any entity, digital or non-digital, that may be used for learning, education or training”

[11], for example, multimedia content, instructional content, Portable Document Format (PDF) files, presentations, etc.

Different standards have been proposed to store, publish, and exchange learning objects between different organizations, universities and learning institutes, the customization of these standards called Application Profile (AP [9]).

The European Committee for Standardization/Information Society Standardization System (CEN/ISSS [9]) defines Application Profile as: *"an assemblage of metadata elements selected from one or more metadata schemas and combined in a compound schema".*

Application Profiles provide the means to express principles of modularity and extensibility. The purpose of an application profile is to adapt or combine existing schemas into a package; that it was tailored for the functional requirements of a particular application, while retaining interoperability with the original base schemas [1].

Prophet Mohammed said *"I seek refuge in you from knowledge which does not benefit"*, the ultimate goal of learning and education is to make our life better and easier and reflecting what we learn in our life, formally the value that we get from learning called *"Learning Outcomes".*

According to Tuning Educational Structures Learning Outcomes [14] defined as "what students should know and be able to do in response to a learning experience."[13], for example (Algorithmic Strategies): *"For each of the strategies (brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming), identify a practical example to, which it would apply. [Familiarity]"* [14], another example, if you are studying JavaScript Programming, the outcome will be the ability to program using JavaScript programming language.

1.2 Motivation

The growth of the World Wide Web made the number of learning resources even bigger than before, as an example, if you search for keywords "JAVA tutorials" in Google search engine, the number of results is about 17500000, see the Figure 1.

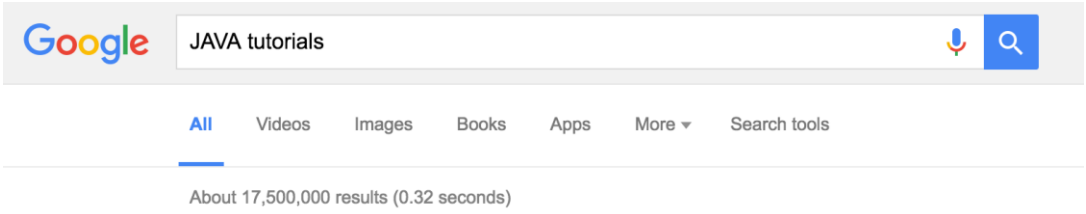


Figure 1 - Google results for "JAVA tutorials" keywords.

This huge number of learning objects will confuse learners, and to solve this problem we should recommend more relevant learning objects based on students needs, and to facilitate the process of acquiring the recommended learning objects, there are standards that allow us to exchange and store learning objects across different repositories around the world, thus different standards have been introduced such as IEEE Learning Object Metadata(LOM) [6], Dublin core [7] and others; these standards are concerned only with learning objects.

On the other hand, there are learning outcomes standards and specifications such as the Reusable Competency Definitions (RCDs) [16] and the ICOPER Learning Outcome Definitions (LOD) [5]), However, these standards do not tackle and enable outcome based learning properly.

Therefore, what needs to be done actually is a proper linkage of metadata records of the learning objects and the learning outcomes, to enables relevant recommendations and delivery of outcome based learning experience.

There are several benefits from linking learning objects to learning outcomes, such as:

- Providing learner with learning objects based on his learning outcomes, previous researches show that dynamic linkage between content and student learning profile could enhance the adequacy of the learning objects [8].
- Open new possibilities to build smarter search engines, which means the results will be more relevant and useful to learner.

It is impossible to set static group of metadata for every application, and meet various needs for different users groups, the obvious solution for these problems is making application profiles (AP), which means customized each group of users to set of metadata according to their purpose [32].

To help users in building LOM Application Profile, several tools were developed, either if learners are computer professionals or not, and embedding learning outcomes data in the LOM record, in the classification category [3], while this technique enable us to link learning object with learning outcomes, it limit us in term of scalability, usability, search inside theses outcomes, and link an outcome with different learning objects.

Our work aims to link learning outcomes with learning objects in a way that enable us to retrieve learning objects based on learning outcomes, as different learning objects can be linked to different learning outcomes and vice versa.

1.3 Research Questions

In order to enable the outcome based learning; in this thesis, we will work to answer the following questions:

- What is the most effective approach to link learning outcomes to learning objects, and assessment?
- How can we build a framework that support modular approach to enable outcome based learning?

- How to retrieve learning objects given learning outcome.
- How to retrieve learning outcome given learning object.
- How to design schema flexible enough to represent learning object and learning outcome metadata.

1.4 Objectives

Innovative Learning Technologies (ILT) lab established at AL-Quds University (AQU) has the goal of enabling the delivery of outcome and competence based learning experience, QLearn platform will be developed to address the pitfalls in current online learning platforms.

This work aims to develop an IEEE LOM [22] Application profile to support linking learning outcomes to learning objects, that enable system to store, search, retrieve different learning resources, and to link different learning objects with learning outcomes.

To achieve these goals, we will do the following:

- Design a schema for Learning Object repository and Learning Outcomes repository based on well-known standards.
- Write software implementation for these schemas and map it to a Relational Database Management System (RDBMS).
- Build a web service Representational State Transfer (RESTful) Web Service on top of it to enable different component of the system to connect to each other (see Figure 2) below.
- Design and implement algorithm to auto link learning objects with learning outcomes.
- Validate the objectives above (empirically).

As a case study, we wrote the software code using Python programming language and we used MYSQL as RDBMS to implement learning objects and learning outcomes repositories, the linking algorithm, and the RESTful web services on top of them as in Figure 2.

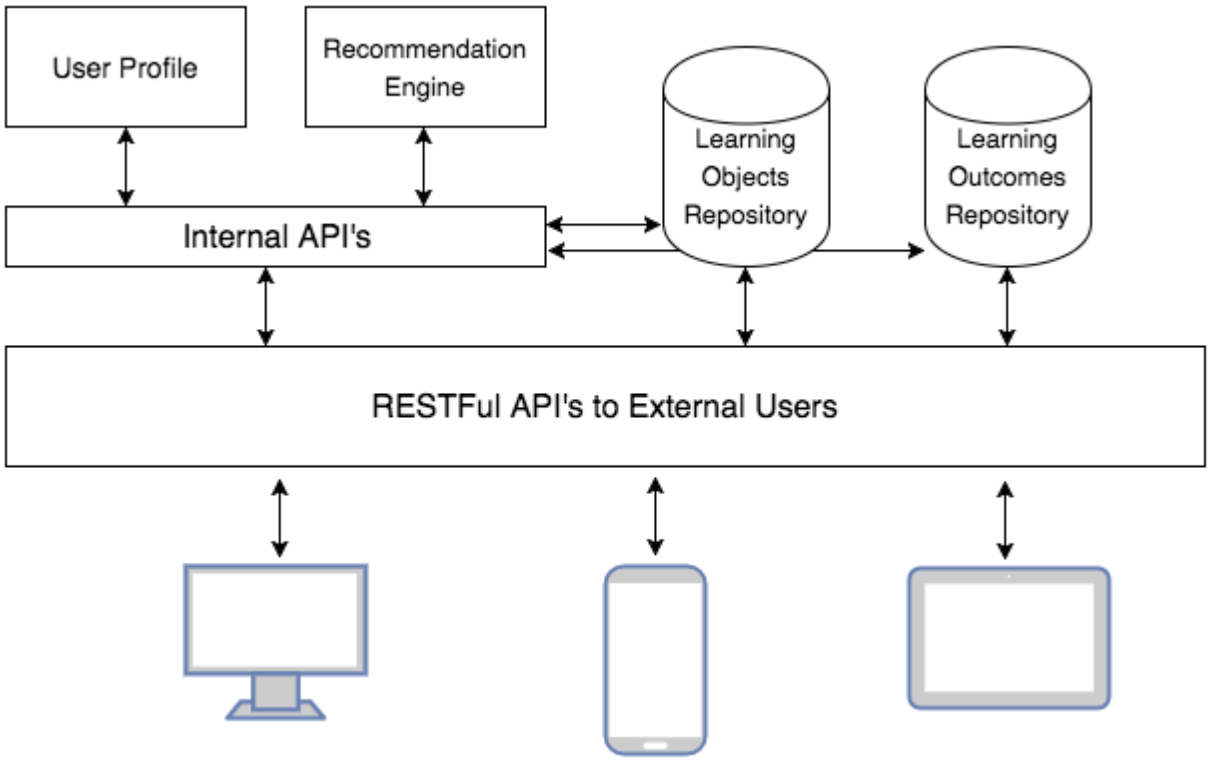


Figure 2 – QLearn system components

1.5 Methodology

We will develop Application Profile (AP) based on IEEE LOM, these are the guidelines for the development of AP's with specific focus on the IEEE LOM Standard [3]:

- Step one – Selection of data elements.
- Step two – Size and smallest permitted maximum.
- Step three – Check if application profile can be based on more than one base metadata schema.
- Step four – Adding local data elements.

- Step five – Obligation of data elements.
- Step six – Value space.
- Step seven – Relationship and dependency.
- Step eight – Data type profiling.
- Step nine – Application profile binding.

International Organizations such as IMS GLC and European Committee publish these guidelines for Standardization.

The next step is to link the IEEE LOM Application profile with learning outcomes, so we can get the required learning objects for a certain learning outcome. To achieve this goal, we will create new entity contain references for learning objects and learning outcomes, these references will enable us to retrieve learning objects from learning outcomes and vice versa, then we will design and implement an algorithm to auto link learning objects and learning outcomes.

Finally, to make sure that the implemented work is integrated and can be used within other system components, we will expose RESTFul API and clear internal API Calls to get the access to learning objects and learning outcomes repositories.

1.6 Thesis Structure

In the first chapter, we show a general overview about the research. Chapter 2 presents background the ILT project, QLearn system, and metadata standards. In chapter 3, the literature review and the major previous researches are introduced. The architecture for learning outcomes, learning objects repositories; their linkage and implementation of a real use case of the system are in Chapter 4. In Chapter 5, we evaluate our implementation. Conclusions and future directions drawn in Chapter 6.

Chapter Two

Background

In this chapter, we will introduce the ILT research project objectives and goals and QLearn platform as this work is part of it, and then we will show different metadata standards related to our work.

2.1 ILT Project

ILT is the driver of research and development around Learning Technologies and the relevant learning platforms, standards and practices at Al-Quds University. QLearn platform will be developed to address the pitfalls in current online learning platforms like (Learning (Course) Management Systems). Qlearn is an outcome based system and will enable effective mapping between learning objectives, learning objects and assessment using keywords mapping between learning objects and learning outcomes.

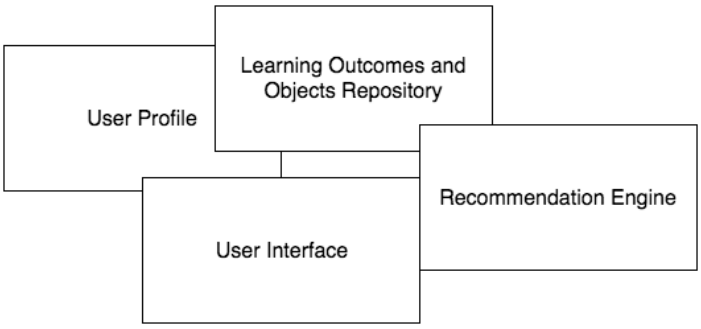


Figure 3- QLearn research project parts

Qlearn proposes the following approach to enable outcome-based learning:

- Subject matter experts collect and/or develop Learning Outcomes across the fields of education and training. Outcomes are analyzed and agreed upon by educators each in his knowledge area or expertise. The outcomes are formulated according to common taxonomies; like Bloom's [33].

- Then proper Learning Objects that satisfy levels of mastery of relevant learning outcomes are identified and selected. Learning Objects are selected from the relevant repositories worldwide or developed locally. Available repositories are rich of resources that are collected and indexed according to standards to enable integrating them with other systems. Learning objects can include concepts, descriptions, structures, application areas, example from real life, animation, demos, solving examples, outlines of processes and/or algorithms, programs in various languages, lab outlines, proofs, comparative analysis, alternatives, solved problems, quizzes, test cases, etc.
- The Qlearn is a Learning Environment that shall visualize and provide excellent mapping between Learning Outcomes and Learning Objects in one integrated setup directed by the learner intended goals. Relevant available resources and assessment tools will be offered to the learners to make sure outcomes are achieved. Learner achievement and over all objectives will guide the learner exploration of learning objects. The Qlearn retrieves resources and assessment items from the proper container of Learning Objects. This container is local and will be connected to the environment but should heavily base on including links to available resources all over. Having repository of links will reduce search time, better utilize available resources, encourage educators to develop Learning Objects that support listed Learning Outcomes, and avoid us worry about copyright issues.

2.1.1 ILT Goals

To support and conduct applied research in fields recognized to gain importance for development in technology enhanced learning and education and learning (content) management platforms.

- Develop young investigators committed to computer science and IT in Learning Technologies-related research.
- Develop flexible and open outcomes-based learning environment that provides efficient mapping between competences, outcomes, learning resources and objects.

2.1.3 ILT Objectives

The major objectives of ILT are the following:

- To promote research in areas that is relevant to learning technologies science.
- To promote interactions among scientists and educators exploring diverse fields which share relevance to education and learning technologies
- To attract and support educators in developing their teaching methods and identify learning resources that help learners achieve goals of learning
- To promote adoption and implementations of standards in exchanging data such as learning objects, and learning outcomes.
- To develop and maintain a learning platform to support education and learning across the fields of knowledge.
- To develop resources and training programs to support educators throughout the cycle of online learning
- To provide Educators and professionals in all fields of education and training the tools to access, manipulate and maintain curricula guidelines profiles as far as level of learning outcomes.
- To provide learners flexibility and achievement based exploration of knowledge and resources.

2.2 Learning Objects Data Models

In this section we will review current available learning objects data models and standards.

2.2.1 IEEE LOM

The IEEE Learning Object Metadata (LOM) standard is a data model, usually encoded in XML or JSON, and used to describe a learning object and similar digital resources used to support learning.

According to IEEE LOM standard draft *"The purpose of this multi-part Standard is to facilitate search, evaluation, acquisition, and use of learning objects, for instance by learners or instructors or automated software processes. This multi-part standard also facilitates the sharing and exchange of learning objects, by enabling the development of catalogs and inventories while taking into account the diversity of cultural and lingual contexts in which the learning objects and their metadata are reused."* [23].

Additionally *"By specifying a common conceptual data schema, this part of this standard ensures that bindings of Learning Object Metadata have a high degree of semantic interoperability. As a result, transformations between bindings will be straightforward."* [23].

Finally *"This Part of this standard specifies a base schema, which may be extended as practice develops, e.g., facilitating automatic, adaptive scheduling of learning objects by software agents"* [23].

Figure 4 below shows the main attributes of the IEEE LOM standard:

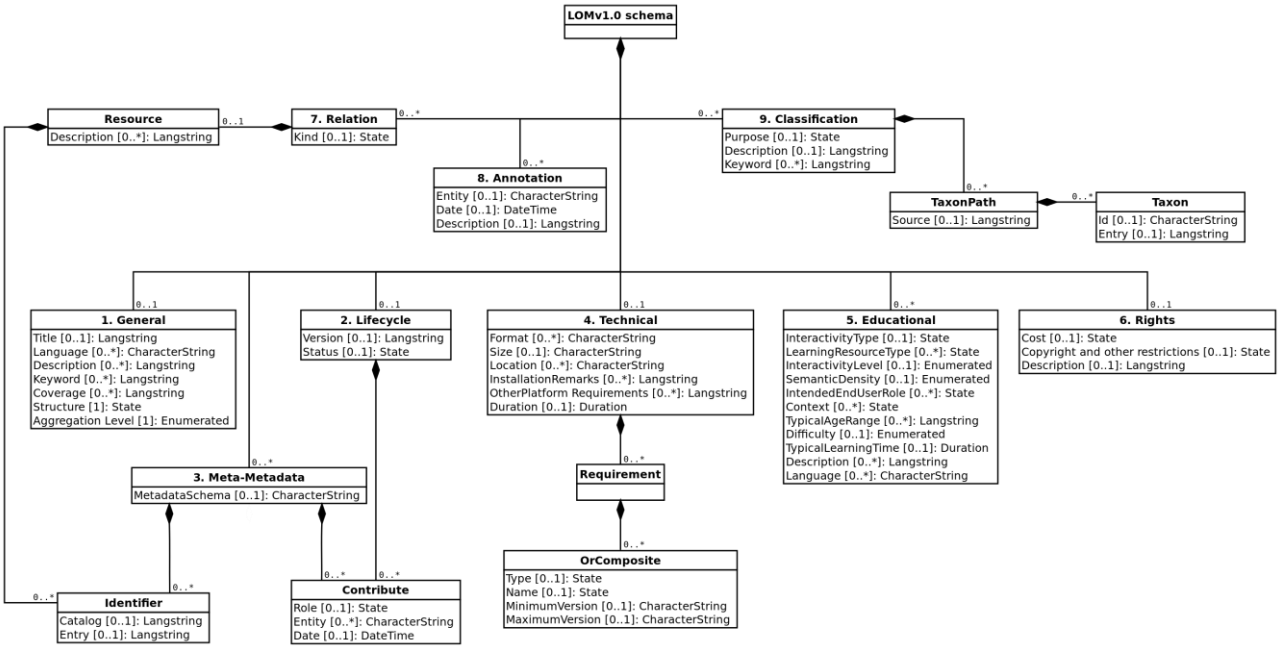


Figure 4 - A schematic representation of the hierarchy of elements in the LOM data model [27]

Table 3 in Appendix summarizes the main IEEE LOM attributes [23]. IEEE LOM is a standard that contain everything related to a learning object, implementer's usually only use subset of IEEE LOM attributes A.K.A. Application Profile.

2.2.2 Dublin Core

While IEEE LOM designed to cover every part related to learning object, for some projects (most of them) need only small part of these attributes, because it will require too much of work to get, store and maintain all these data, a simpler standard has been proposed: Dublin Core [24].

Below is an example of XML Meta data used in Dublin Core standard:

```xml
<?xml version="1.0" encoding="UTF-8"?>

    <metadata
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:title>My Learning Object Title</dc:title>
    <dc:creator>Ahmad Shukr</dc:creator>
    <dc:subject> My Learning Object Subject</dc:subject>
    <dc:description>My Learning Object Description</dc:description>
    <dc:publisher>Al-Quds University</dc:publisher>
    <dc:contributor>Jad Najjar</dc:contributor>
    <dc:type>PDF</dc:type>
    <dc:format>PDF</dc:format>
    <dc:identifier>123</dc:identifier>
    <dc:source>http://alquds.edu</dc:source>
    <dc:language>english</dc:language>
    <dc:relation>IsPartOf</dc:relation>
    <dc:coverage>2014-2020</dc:coverage>
    <dc:rights>CC</dc:rights>

</metadata>
```

Figure 5 – Dublin core meta XML data example

Table 4 in Appendix describes Dublin core main attributes.

Dublin Core is simpler and more straightforward standard than IEEE LOM, while IEEE LOM is widely used across many projects around the world such UK LOM Core [25], CanCore [26].

The table below summarizes the difference between IEEE LOM and Dublin CORE.

IEEE LOM	**Dublin CORE**
Cover wide range of learning objects attributes	Cover fewer number learning object attributes
Can be encoded in XML or JSON	Can be encoded in XML or JSON
Entities cover in details the properties of each in entity.	Entities cover in brief the properties of each in entity.
Designed for educational purposes.	Designed for generic usage purposes.

Table 1 - IEEE LOM and Dublin CORE compare table.

Table 1 show that IEEE LOM Data model cover more attributes than Dublin CORE standard with more details for each entity, and that's why we adopted it for QLearn project.

2.2.3 IEEE RCD

The IEEE Reusable Competency Definitions (RCD) standard is the only widely accepted standard for describing competencies. It is a continuation (and replacement) of the early efforts on the development of IMS Reusable Definition of Competency or Educational Objective (RDCEO) [16].

This specification profiles IEEE RCD with one metadata element defining the type of the learning outcome and its associated value domain to capture whether a learning outcome refers to knowledge, skill or competence following the definitions of the European Qualification Framework [16].

Instances that conform to the Learning Outcomes Definition (LOD) specification also conform to the IEEE RCD standard, which ensures interoperability between the OICS and other systems [16].

Chapter Three

Literature Review

Learning objects and learning outcomes standardization and how to link them are the main topics of this thesis, we will introduce related works to learning objects and learning outcomes metadata standards.

2.1 Learning objects metadata

In this section we will review works related to learning objects metadata, and the standards they used to store and exchange learning objects.

2.1.1 Reusable Learning Objects: a Survey of LOM-Based Repositories (2002)

They surveyed the field of learning objects on eight repositories (*ARIADNE, SMETE, Learning Matrix, iLumina, MERLOT, HEAL, CAREO, Learn-Alberta, EdnA, Lydia),* seven of these repositories were using IEEE LOM stanard Application profile, while only one repository used Dublin Core standard. [45]

2.1.2 Interoperability of Learning Object Repositories: Complications and Guidelines (2004)

In this paper they presented an approach for interoperable metadata using application profiles, they transformed ARIADNE XML instances into IEEE LOM using Extensible Stylesheet Language Transformations(XSLT) which is a programming language used for processing XML files, the major contribution in this paper is that the mapping between different metadata specifications is not always a simple one-to-one process.[46]

2.1.3 eMAP: Design and Implementation of Educational Metadata Application Profiles (2004)

The architectural components and the fundamental functionalities of a software toolkit (eMAP) aims to assist and facilitate the process of designing and implementing an

educational metadata application profile by modifying and combining one or more educational metadata standards.

First they introduce why metadata is important, as its created and used in correspondence with online learning objects in order to facilitate the process of describing, indexing, searching, reusing and retrieving educational resources. According to them there is limitation in the existing tools, so they designed and implemented eMap tool, and offered a graphical user interface for easy metadata authoring.

2.1.4 The LOM application profile for agricultural learning resources of the CGIAR (2009)

LOM was used across different domains; the best use of LOM is the Application Profile of the international agricultural research centers of the CGIAR project. CGIAR developed CG LOM Core based on IEE Learning Object Metadata (LOM) standard, CG LOM Core defines the collection of metadata elements and value spaces to meet CGIAR requirements, and it preserves the interoperability with other repositories outside of the CGIAR.

The goal of this project was to make CGIAR learning resources available to the intended users, one of the challenges they faced was to choose which standard they should use in the profile, such as LOM or Dublin Core, etc. [18]

The authors studied different Applications Profiles; these profiles are based on IEEE Learning Object Metadata (LOM) standard to *"identify mandatory elements used in the implementation of other organizations."* [18], then they go through the steps of the recommendations guidelines to develop Application profile.

2.1.5 Supporting the Process of Developing and Managing LOM Application Profiles: The ASK-LOM-AP Tool (2012)

Introducing a new web-based tool (ASK-LOM-AP) [3] facilitate the process of developing and manage LOM Application Profiles for different educational communities.

First: The authors show the basic guidelines for developing Application profiles based on The IMS Global Learning Consortium recommendations [3], then they showed examples of existing application profiles such as The CELEBRATE Application Profile, UK-LOM Core Application Profile and others, then they mention existing tools for development and management of IEEE LOM applications profiles, for example eMAP Tool [19] and others, then they show the tool they have developed through showing its Software architecture and some screenshots from the software they have developed.

They evaluated their research, by conduct workshops to develop Applications profile using the tool they developed, The authors suggest future work on the research which is*: "the development of new functionalities that will enable the learners of the tool to build communities around the different Application Profiles developed with ASK-LOM-AP offering their comments and experiences regarding the usage of each Application Profile and the development of new functionalities that will facilitate collaboration and consensus building among the members of educational communities and e-learning experts for translating conceptual APs into concrete representations and bindings."* [3].

2.2 Outcome based learning

In this section we will review works used learning objects in context of learning outcomes.

2.2.1 Repository Services for Outcome-based Learning (2010)

In their work on the Outcome-based Learning Repository [34], the authors present the concept and prototypical implementation of an open architecture that aims to remedy (fix) these issues by providing a unified metadata and service layer for making key educational resources sharable, storable, findable, and interoperable. The reference model and its supporting technology architecture are tested by a family of prototypes implemented as

extensions to or adaptations of existing mainstream systems like Moodle, .LRN, Elgg and Facebook.

2.2.2 Learning Outcome Based Higher Education: iCoper Use Cases (2009)

iCooper is European project that enable online learning outcomes based on education at university environments.[15], The learner use search terms for browsing particular knowledge, skill or competency taxonomy.

- **Knowledge** is recognition and recall of facts and specifics.
- **Skill** is the application of certain knowledge.
- **Competencies** "are the effective application of skills" [17],

For example, knowing how to program is a skill, being a programmer is a competency because you are applying the skill in effective manner.

The learner then select the context where the outcome should be obtained from, a context is the domain where the needed outcome located such as math, biology, lab, classroom..etc., then the learner select the type of outcomes he is looking for (knowledge, skills, competences), then the system return the units of learning e.g. learning objects, that can be used to achieve the desired qualification, other use cases enabled in iCooper project depicts in Figure 7.

2.2.3 A Data Model for Describing and Exchanging Personal Achieved Learning Outcomes PALO (2012)

The Personal Achieved Learning Outcomes (PALO) data model is simple schema that link learning outcomes information (knowledge, skill or competency) with a learner profile [16].

- The PALO data model enables to capture the **Relations** no matter if its taxonomies or ontologies belong to [16].
- **The Context** in the domain where the learner achieved the outcome.

- **Evidence and assessments** are proving of the achievement of the learning outcomes.
- **Information about levels and ranking** of achieved learning outcomes [16].

The Personal Achieved Learning Outcomes (PALO) specification went through at least three iterative expert evaluations by the ICOPER consortium and at relevant international workshops by standards experts, teachers and learners to make sure that it captures data needed for increased employability of learners and higher interoperability with different learning systems.

Prototypes of outcome based learning applications like widgets and modules of Learning Management Systems (LMS) are developed, to produce and import data about achieved learning outcomes of learners in systems like Moodle [35], Elgg [36] and Clix [37].

The data of learner achieved learning outcomes profiles are stored in ICOPER's PALO repository. These data can be consumed by learning systems to provide learners with relevant material, recommendation of other teachers and learners based on similarity of learning outcome profiles, or to enable learners to share their achievement profiles with social or recruitment systems.

The authors propose a schema that facilitates interoperable storage and management of Personal Achieved Learning Outcomes (PALO).

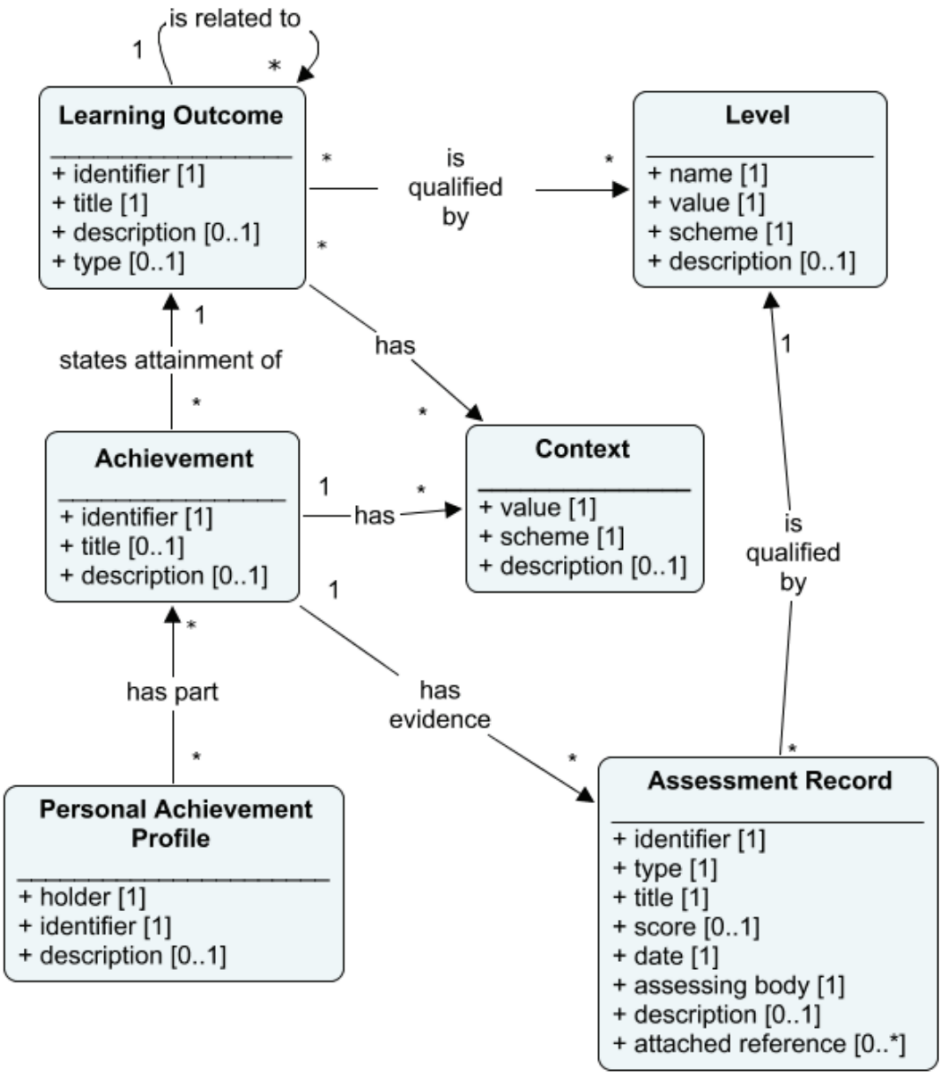


Figure 6 – PALO data model [16]

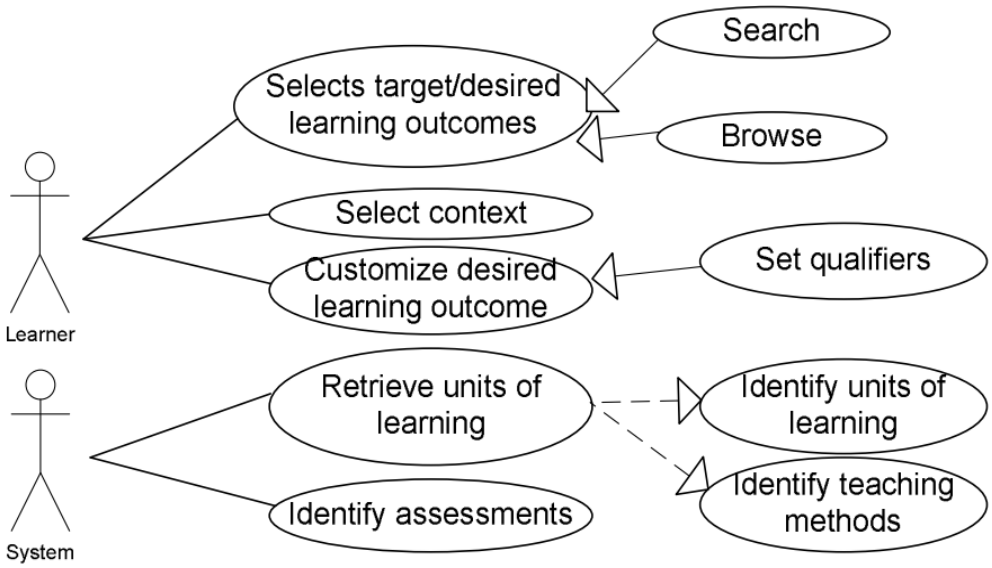


Figure 7- iCoper use-cases (Competency based search) [15]

Summary

Related works are summarized below, we compared projects related to learning objects standards in Table 2 we compared projects related to linking learning objects with learning outcomes, and these projects are very related to our work.

Project Name	**Used Standard for Learning Objects Data Modeling**	**Used Standard for Learning Outcomes Data Modeling**	**Application Profile**
A Data Model for Describing and Exchanging Personal Achieved Learning Outcomes PALO	IEEE LOM	IEEE RCD	Yes

Learning Outcome Based Higher Education: iCoper Use Cases	IEEE LOM	IEEE RCD	Yes
QLearn (Our research project)	IEEE LOM	IEEE RCD	Yes

Table 2 – Compare outcome based learning works

Chapter Four

Case Study

In this chapter, we will introduce QLearn platform components, the responsibility of each component, system architecture, how we linked learning outcomes with learning objects, RESTFul API's, and system implementation.

4.1 Introduction

QLearn recommendations engine was divided into three main parts:

- The first part related to how to store and track user profile.
- The second part is how to process the data from user profile learning outcomes and learning objects repositories.
- The third part related to how to store learning objects and learning outcomes and links the learning objects with learning outcomes, which we will focus on in this work.

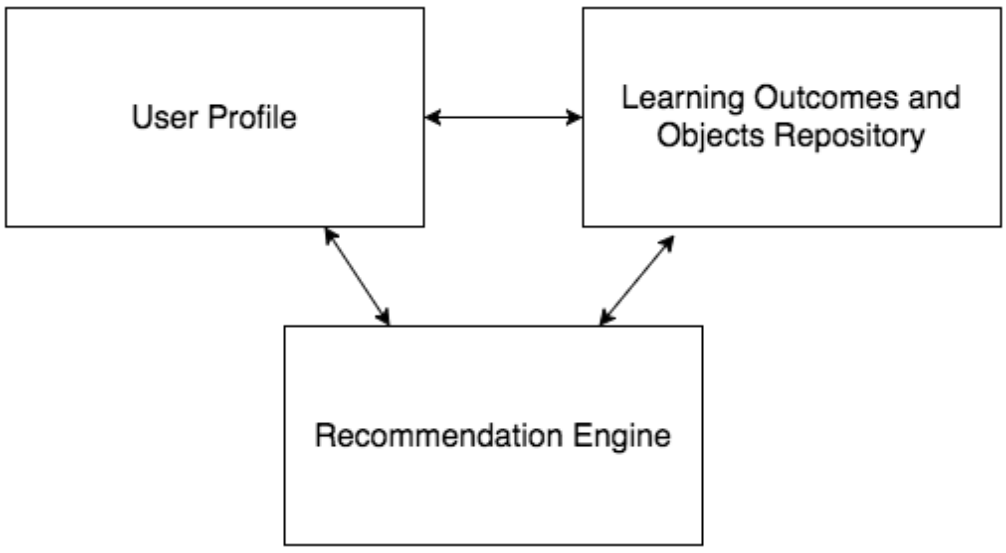


Figure 8 – Qlearn main parts

Linking learning objects with learning outcomes is the core part of QLearn, for example if we want to get the learning objects for learning outcome, the framework should provide an

easy method to get the learning outcomes either via programming method call, or using RESTful API's for external entities.

This framework provides:

- RESTful API's and programming methods to:
  - Get list of learning objects, and related outcomes or vice versa.
  - Add/Update new learning objects and learning outcomes.
  - Auto linking between learning objects and learning outcomes.

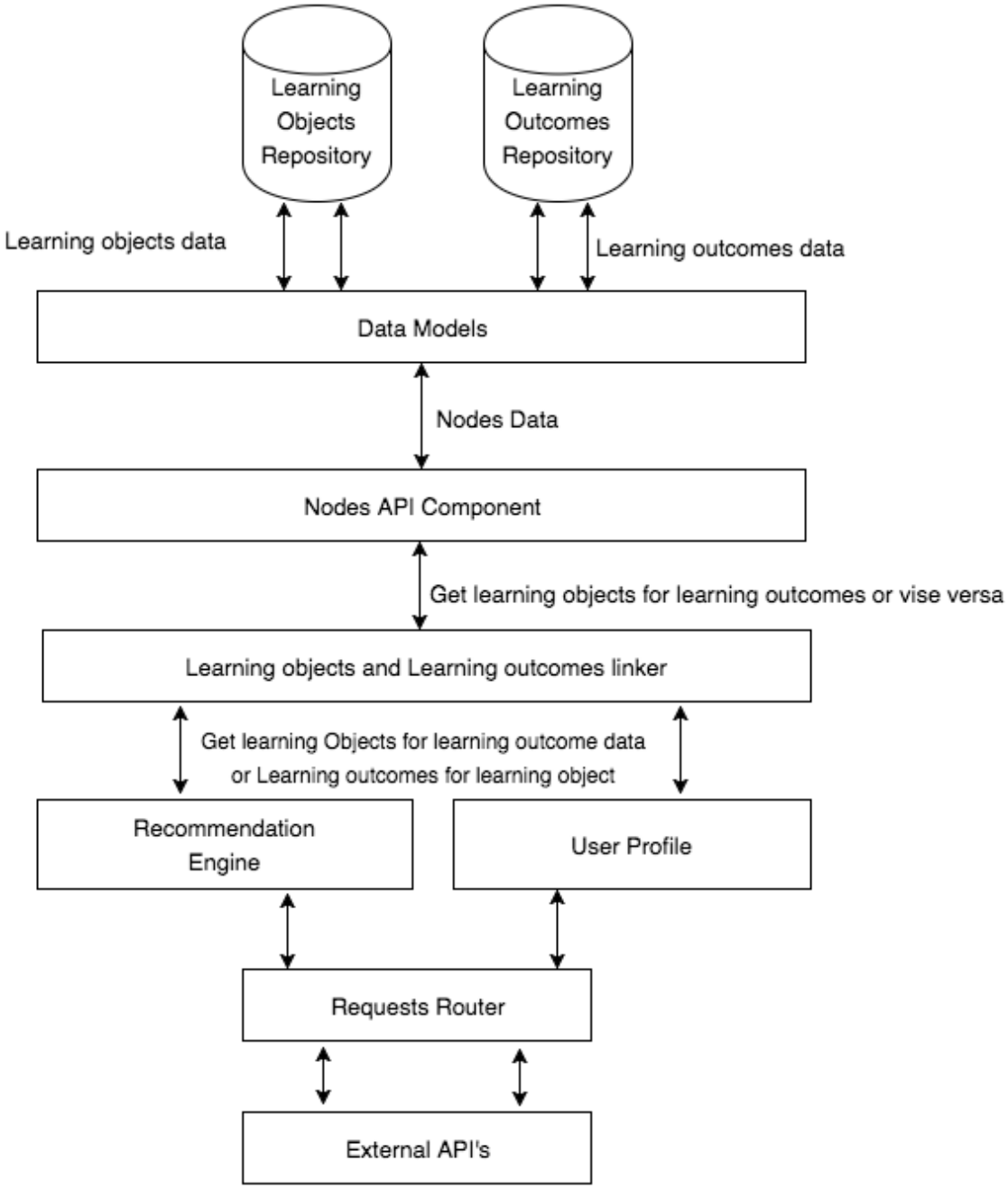


Figure 9 – ILT Architecture

4.2 Learning Object Repository

We have choose IEEE LOM [23], a well-known standards, to represent learning objects in our repository, and it was used in several other projects [3][25][26], basically LOM that contains most of the attributes required for a learning object [23]. In addition, using LOM will make it easier for us to exchange the learning objects with other repositories without

many extra modifications because any repository can implement a web service to parse IEEE LOM learning objects.

The learning object repository is a repository that contains set of learning objects. Learning object can be video files, URL for learning game, or a PDF file, Each learning object should at least have an identifier and other attributes are optionally provided when we insert new object to the repository.

4.3 Learning Outcomes Repository

For learning outcomes, limited numbers of standards were available; we used IEEE RCD a simple standard to store Learning Outcomes, this standard only contain title, description, and type of given learning outcome, and this is what really needed to store learning outcome.

Figure 10 shows the main ILT project parts. The different parts of the architecture are described in the subsections below.

Each learning outcomes is located under a certain classification with Tree-Like structure, examples of learning outcomes statements for "Automata Theory Subject":

- "Generate a regular expression to represent a specified language."
- "Define the classes P and NP."
- "Discuss the concept of finite state machines."
- "Explain the significance of NP-completeness."
- "Design a context-free grammar to represent a specified language."

The learning outcomes repository schema is based on ICOPER Learning Outcome Definitions (LOD) [36] concepts, the major concept we took from iCOPER is representing

data as nodes, and we called it **QLearn-*Outcomes***. But with minor modifications to our project purposes, on how data stored in schema like in Figure 11.

4.4 Data Storage

We have set of nodes, each node have association with other nodes that enable us to build a graph of learning objects and learning outcomes or basically any object we want.

The tree like structure used to store data enable us to find new relationships in any depth, also its easier to map any hierarchy to it, the node could be one of the following values:

- Broad Field
- Narrow Field
- Detailed Field
- Knowledge Area
- Knowledge Unit
- Course
- Topic
- Sub Topic
- Learning Outcome
- Learning Object

The Table diagram goes as the following:

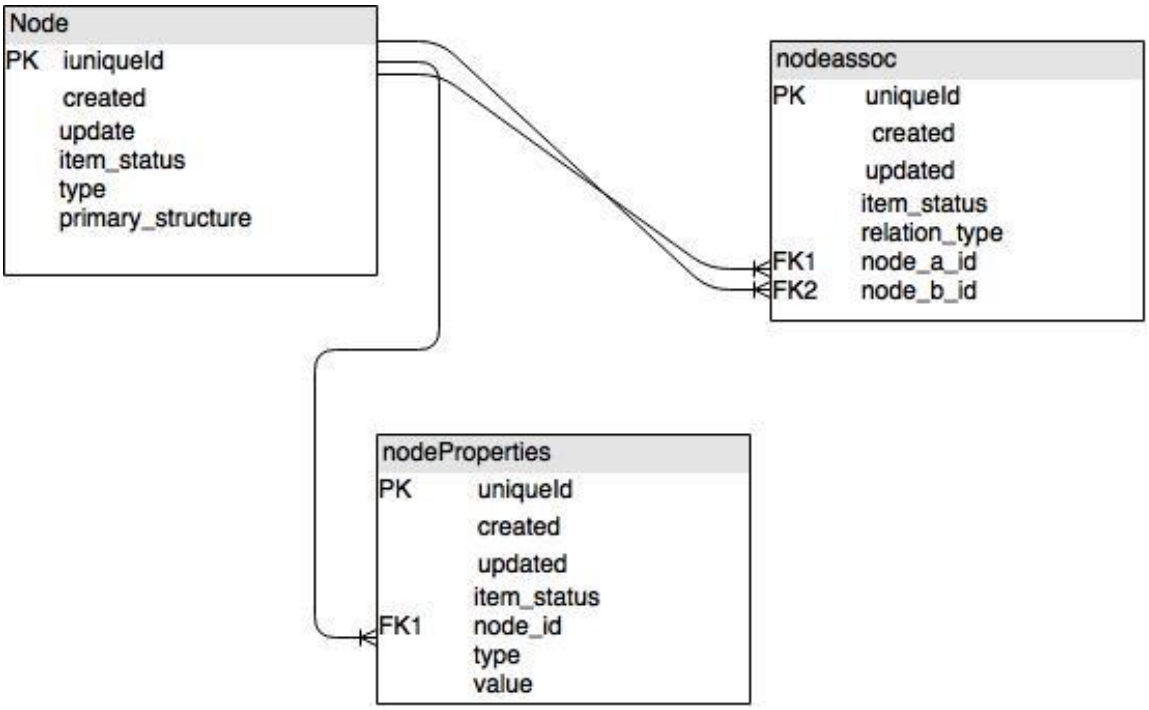


Figure 11 – ILTOutcomes database diagram

The description of the above figure:

- Node table contains a set of nodes.
- Each node connected through set of associations, through Node Association table.
- The node properties table contains set of properties for each node such as LABEL, LEVEL, OUTCOME_LEVEL, TIER_HOURS.

4.5 Linking Learning outcomes to learning objects

The core idea behind QLearn is to link the learning outcomes to learning objects, this will open new way for recommending learning objects based on learning outcomes.

We introduced new relationship between learning outcomes and learning objects, each learning outcome has multiple learning objects and each learning object maybe have multiple learning outcomes, and this is many-to-many type of relationship, the way data stored as shown in the previous section will enable us to implement this type of relation ship as any node can be associate with other nodes.

Then we design and implement algorithm to auto link learning objects and learning outcomes based on learning outcome statement and learning object title keywords.

To compute the distance between each two key words, we will use Levenshtein distance formally defined as "The algorithm finds the cost of the least expensive set of insertions, deletions or substitutions that would be needed to transform one string into the other" [70].

Below we describe in algorithmic language how we actually auto linked each learning object with learning outcomes:

1. Parameters (***X:*** keywords, ***Y*:** Levenshtein distance threshold)
2. *Initialization*:
    a. Clean all stop words in the learning outcomes.
    b. Take the longest X keywords in the learning outcomes.
    c. Set string Levenshtein distance threshold to Y.
3. For each keyword in the learning outcome
    a. Query two web services such as Youtube[41] and Ariadne[42].
    b. Return list of learning objects from the web services
    c. For each returned learning object in the list:
        i. Split the returned learning object string into keywords by space.
        ii. Set distance total to 0.
    d. For each learning object keyword:
        i. Compute the distance between the longest X keywords in the learning outcome and learning object keyword using Levenshtein distance algorithm.
        ii. Add distance value to distance total
    e. Compute average distance by dividing distance total by the total number of learning outcome keywords which is X.
4. If the average distance > threshold (Y):
    a. Link learning object with learning outcome.
    b. Where X is the number of keywords from learning objects, and Y is the Levenshtein distance threshold.

And below is the code snippet for algorithm implementation (written in Python Programming Language [28]):

```
def link_learning_objects_to_learning_outcomes(self):
    learning_outcomes = get_leaf_of_type('LEARNING_OUTCOME')
    for learning_outcome in learning_outcomes:
        learning_outcome_node = lod_app.models.Node.objects.get(id=learning_outcome['key'])
        learning_outcome_keyowrds = learning_outcome['title'].split(' ')
        learning_outcome_keyowrds = sorted(learning_outcome_keyowrds, key=len, reverse=True)

        # generate random number between 1 and 4 for the number of keywords and avg levenshtein distance
        numberY = random.randrange(1, 4)
        max_distance = random.randrange(1, 4)
        query_text = ' '.join(learning_outcome_keyowrds[:numberY])

        learningo_title_prop = lod_app.models.NodeProperties.objects.filter(value=learning_outcome['title']).first()
        # Attach them to title
        learningo_title_prop.value = 'X:'+str(numberY)+'-Y:'+str(max_distance) +' ' + learningo_title_prop.value
        learningo_title_prop.save()

        # query web service
        items = get_by_keywords(query_text)
        for item in items:

            vid = item['id']['videoId']

            learning_object = lod_app.models.Node.createNode('LEARNING_OBJECT')

            if 'default' in item:
                item = item['default']
            elif 'title' in item:
                item = item['title']
            elif 'snippet' in item:
                item=item['snippet']

            # compute levenshtein distance
            avg = lev(item['title'], learning_outcome_keyowrds)
            if avg<max_distance:
                lod_app.models.NodeProperties.prop(learning_object, 'TITLE', item['title'])
                lod_app.models.NodeProperties.prop(learning_object, 'DESC', item['description'])
                lod_app.models.NodeProperties.prop(learning_object, 'LINK', 'https://www.youtube.com/watch?v=' + vid)
                lod_app.models.NodeAssoc.create_assoc(learning_outcome_node, learning_object)
```

Figure 12 – Auto linking learning objects to learning outcome algorithm implementation

4.6 Linking Algorithm and Metadata Standards

Right now the linking algorithm used only the keywords in title attribute from learning objects and learning outcomes for mapping, other attributes from the standards also might be used, more specifically from IEEE LOM:

- The similarity in language attribute.
- Similar classification.
- Contributors, for if they are the same contibuter
- Related taxonomies.

And in IEEE RCD, mainly the type of learning outcome (skill, knowledge, competency) can be used to get related learning object from the same classification (type), for example if the

outcome type is skill then get only learning objects with similar type, all these factors might be used in computing relevancy weight.

4.7 Data Access Layer

In order to separate the actual data representation from the application, we introduce this layer, which enable us to change the RDBMS engine to another, without too many changes on the application level.

We introduced shared component called BaseModel, which is the base model for each entity, this will make it easier for us to add general attribute for each entity, and the model looks like the code snippets.

```
class BaseModel(models.Model):

    created = models.DateTimeField(auto_now=True)
    updated = models.DateTimeField(auto_now=True)
    item_status = models.BooleanField(default=True)

    class Meta:
        abstract = True

    def __unicode__(self):
        return str(self.created)

.....
```

Figure 13 – ILT learning objects repository based model

The code snippets above show the new attribute we introduce for each entity in IEEE LOM, created and updated date for helping us in tracking creation date and update date. In case nothing provided in the creation process, the default value for the dates (updated and created) will be current time stamp in the system.

The item status attribute help us in activating or deactivating the item, which is very helpful in some cases; for example, once we like to hide the learning object from the search engine result, without being deleted it completely from the repository or for “soft deletion”.

4.8 RESTful API

REST is a software architecture for building scalable web Application Programming Interface, (API) to exchange data between client and server, originally its described by Roy Thomas Fielding (in 2000) in his PhD thesis [29], REST is an architectural programming philosophy or paradigm, and not a web standard for design web services [30]. RESTful is the adjective of the REST, RESTful typically used to describe the web services that implement REST architectural.

Fielding described REST as: *"The Representational State Transfer (REST) style is an abstraction of the architectural elements within a distributed hypermedia system REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements. It encompasses the fundamental constraints upon components, connectors, and data that define the basis of the Web architecture, and thus the essence of its behavior as a network-based application."* [29].

4.9 REST architectural constraints

Now a days, most of the REST characteristics implemented using HTTP protocols, the main characteristic of RESTful protocols as listed below [29]:

Client-Server

The protocol should use client-server architectural, client-server architectural apply separation of concern principal, which separate UI concerns and data processing, this allow us to improve the portability of the learner interface code, and the scalability of the backend servers.

Stateless

Which means that each request is treated as separated request and it's unrelated to the previous request, sessions is used to track user status and it's stored completely on the

client, and that's mean that each request should contain all the needed to information understand the request.

Cache

Is the ability to provide the learner with required information without travel the same trip each time, in this context cache means the ability to provide the learner with required information without hitting the server each time the learner request the same information, which is obviously improves the performance of getting response and also improve network traffic and congestions.

Uniform Interface

Each component should have uniform interface to talk with, that's enable the component to evolve without any interference with other system components for example we use URI to identify end-point server services, the URI use DNS protocol [29] to resolve URI to IP, which allow the service to scale without any interruptions on the client side.

Layered System

Separate a system into layers simplify the complexity of the system, and also improves the system scalability, whereas it has some cons such as the overhead causing via intermediate processing layers.

Code-On-Demand

Allow the server to send downloadable code that run on the client side, example of modern usage of this feature: JavaScript code, Flash, HTML..etc.

QLearn Web Services Architecture

We take the advantages of HTTP protocol to implement general REST web services, basically the learner of the web services could do all the Create Read Update Delete (CRUD)

operations via the web services, the figure below show the basic structure of QLearn RESTFul web services.

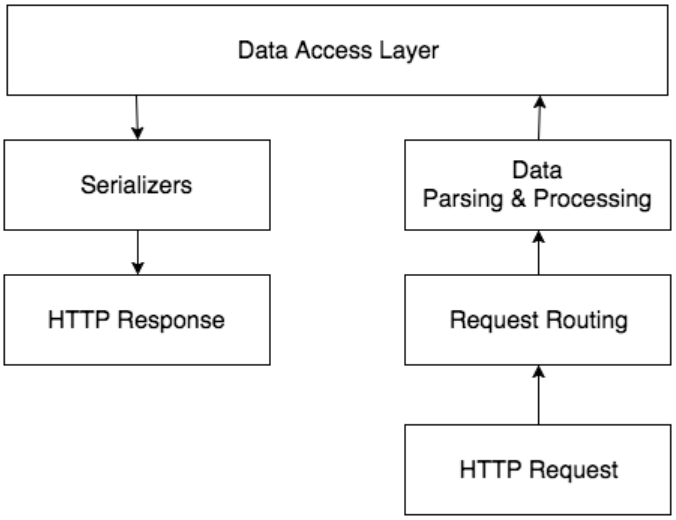


Figure 14 – ILT basic web services structure

The data received via HTTP request call below, is an example of real HTTP request made using real ILT web service:

PUT /api/v1/learning_objects/ HTTP/1.0

Host: localhost:8000

Accept: */*

Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.8,ar;q=0.6,de;q=0.4

Cache-Control: no-cache

content-type: application/json

...

payload data here..

…

Response:

HTTP/1.0 200 OK

Access-Control-Allow-Origin: *

Allow: GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS

Content-Type: application/json

Date: Fri, 20 Feb 2015 08:43:09 GMT

Server: WSGIServer/0.1 Python/2.7.6

Vary: Accept

X-Frame-Options: SAMEORIGIN

Once the HTTP request is received on the web service, the server parses the request, analyze, and rout it to the appropriate handler, for example the code snippets below is the handler for adding new learning outcomes:

```
def put(self, request):
    data = json_decode(request.body)
    validation = LODValidation()
    data_validation_result = validation.lod_validation(data)
    if not data_validation_result['done']:
        return error(data_validation_result['error'])
    else:
        result = add_update(data)
        if result['done']:
            return success("the learning outcome '"+data['identifier']+"' added successfully")
        else:
            return error(result['error'])
```

Figure 15 – Learning outcomes add handler

Basically, each handler should do the following steps when it receiving the data:

Parse data: For example, parse JSON data into language data structure, in our case Python programming language, the steps below show how we parse it:

- Validate the data before insert it to the database.
- If the data passed the validation step:

    - Go data layer step.
- Return success message to the learner or error in case of any exception happened during the data access layer step.
- If not return error to the learner.

On the other way if user request data from the web service to get the latest learning objects, he should perform HTTP call get the data from the web service:

- Route the request to the appropriate web handler.
- Get the required data from data access layer.
- Serialize data in the appropriate format.
- Return HTTP response with the requested payload.

In the coming section, the implementation of the architecture is described in detail.

4.10 Architecture Implementation

As an implementation for QLearn project, we implemented a tool called **COURSE BUILDER,** which allows the learner to easily create a course based on Topics and Learning Outcomes, and allow him to link learning objects to learning outcomes.

After that, we implement a proof of concept for User Profile, User Interface, Learning objects and Learning outcomes repositories all integrated together.

In this section, we present the implementation of the QLearn architecture that shows how the web services and user interface are related, behave and interchange data.

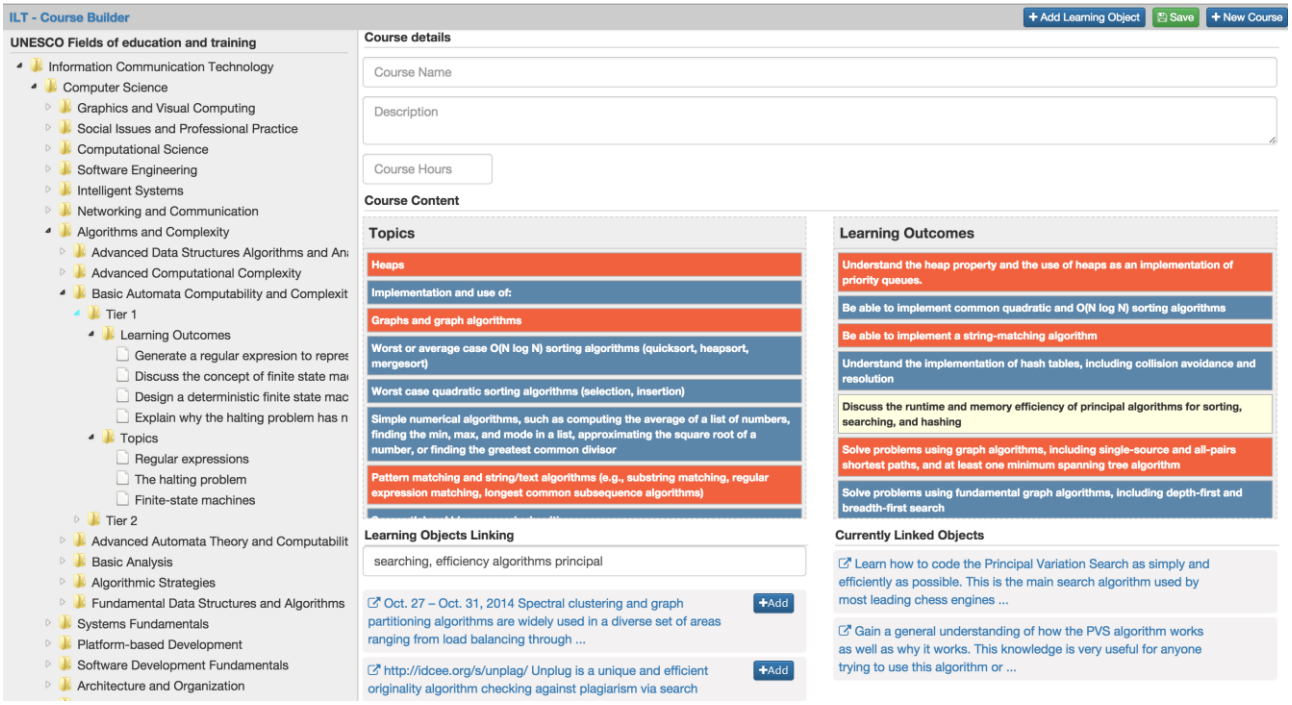


Figure 16 – Course builder tool

4.11 UNESCO Fields of education and training

We have implemented UNESCO Fields of education and training, shown in Figure 16 – Course builder tool, the tree represents from top to bottom:

- Broad Field: such as Information Communication Technology.
- Narrow Field: such as Computer Science.
- Detailed Field: such as Computer Science.
- Knowledge Area: such as Algorithms and complexity
- Knowledge Unit: such as Basic Automata Computational complexity
- Tiers: from 1-3 (based on IEEE/ACM CS Curricula scheme).
- Learning Outcomes: such as “Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing”
- Topics “Generate a regular expression to represent a specified language.”

Through the tree, (see figure 16) you can easily navigate through the Tree by expanding the nodes and visualize how it's structured.

4.12 Implementation

The tree is divided into two components:

The backend: RESTFul API to get the children of node id, code example:

```python
def get_node_properties(request, node_id):

    try:
        node_properties = NodeComponent.get_node_properties(node_id=node_id)
    except Exception as e:
        return JSONResponse({"error": str(e.message)}, status=400)

    result =[]
    for node_property in node_properties:
        result.append({
            "id": node_property.id,
            "type": node_property.type,
            "value": node_property.value
        })

    return JSONResponse({"properties": result}, status=200)
```

Figure 17 – Code snippet of get node properties written in python

The code above (Figure 17) get the primary structures (ROOT) in case we have primary key in the HTTP GET Request and append the structure to data array, later the array will convert to JSON format and then it will be send back to the client side (Front End).

Request Example:

```
▾General
    Remote Address: 62.90.25.202:9000
    Request URL: http://ilt.science.alquds.edu:9000/get_lod?key=2499&_=1435579680880
    Request Method: GET
    Status Code: 200 OK
▾Response Headers      view source
    Connection: Keep-Alive
    Content-Type: application/json
    Date: Mon, 29 Jun 2015 12:06:19 GMT
    Keep-Alive: timeout=5, max=96
    Server: Apache/2.4.7 (Ubuntu)
    Transfer-Encoding: chunked
    X-Frame-Options: SAMEORIGIN
▾Request Headers      view source
    Accept: application/json, text/javascript, */*; q=0.01
    Accept-Encoding: gzip, deflate, sdch
    Accept-Language: en-US,en;q=0.8,ar;q=0.6
    Cache-Control: no-cache
    Connection: keep-alive
    Cookie: sessionid=l0br7mjsy1kv60osx21346wi5p3fvvxi; csrftoken=XHQ2Wd9qpjfiTDRMsyyZfd2Az8aL1HKc
    Host: ilt.science.alquds.edu:9000
    Pragma: no-cache
    Referer: http://ilt.science.alquds.edu:9000/course_builder/2734
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2443.0 Safari/537.36
    X-Requested-With: XMLHttpRequest
```

Figure 18 - Example of tree data request

Response Example:

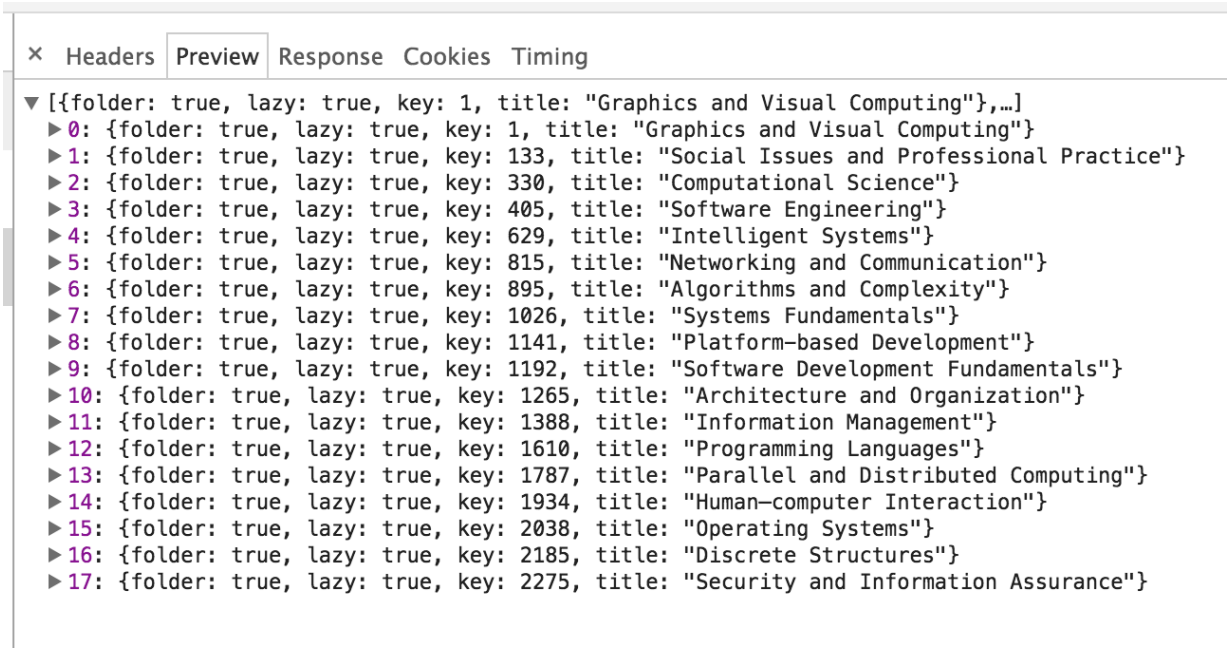


Figure 19 – Example of the formatted JSON data

4.12 The Front End

The Front End (Web Browser) sends HTTP GET request to get nodes status from the API, initialize the Tree widget and render the nodes, in case the learner clicks on the parent child an *Asynchronous JavaScript and XML (AJAX) a* Request is sent to the server to get the required data; as shown in Figure 20.

```
$("#tree").fancytree({
    autoScroll: true,
    source: {
        url: "/get_lod?primary=1",
        cache: false
    },
    lazyLoad: function (event, data) {
        var node = data.node;
        // Issue an ajax request to load child nodes
        data.result = {
            url: "/get_lod",
            data: {key: node.key}
        }
    },
    filter: {
        autoApply: true,
        // autoExpand: true,
        mode: "hide"
    },
    extensions: ["dnd"],
    dnd: {
        preventVoidMoves: true,
        preventRecursiveMoves: true,
        autoExpandMS: 400,
        dragStart: function (node, data) {
            if (node.folder) {
                return false;
            }
            draggedId = data.node.key;
            return true;
        },
        dragEnter: function (node, data) {
            // return ["before", "after"];
            return true;
        },
        dragDrop: function (node, data) {
            //   data.otherNode.moveTo(node, data.hitMode);
        }
    }
});
```

Figure 20 – Example of initialization code for the tree widget

4.13 Node data storage

So far, we did show backend API's, the request made on server, server response then user interface and front end part, now we will show how the data actually represented inside the RDBMS engine.

Example of how nodes data stored inside table is shown in Figure 21.

id	created	updated	item_status	type	primary_structure
1	2015-06-16 23:04:52	2015-06-16 23:04:52	1	KA	0
2	2015-06-16 23:04:52	2015-06-16 23:04:52	1	KU	0
17	2015-06-16 23:04:52	2015-06-16 23:04:52	1	KU	0
35	2015-06-16 23:04:52	2015-06-16 23:04:52	1	KU	0
57	2015-06-16 23:04:52	2015-06-16 23:04:52	1	KU	0
87	2015-06-16 23:04:52	2015-06-16 23:04:52	1	KU	0
108	2015-06-16 23:04:52	2015-06-16 23:04:52	1	KU	0
128	2015-06-16 23:04:52	2015-06-16 23:04:52	1	KA	0
129	2015-06-16 23:04:52	2015-06-16 23:04:52	1	KU	0

Figure 21 – Nodes storage

Example of how nodes properties are stored inside table as in Figure 22:

id	created	updated	item_status	node_id	type	value
6064	2015-06-17 06:27:05	2015-06-17 06:27:05	1	2732	desc	Automata 1
6067	2015-06-28 16:43:12	2015-06-28 16:43:12	1	2734	desc	123
6070	2015-06-28 16:45:56	2015-06-28 16:45:56	1	2735	desc	desc
6076	2015-06-28 16:50:46	2015-06-28 16:50:46	1	2736	desc	test
6065	2015-06-17 06:27:05	2015-06-17 06:27:05	1	2732	hours	2
6068	2015-06-28 16:43:12	2015-06-28 16:43:12	1	2734	hours	123
6071	2015-06-28 16:45:56	2015-06-28 16:45:56	1	2735	hours	123
6077	2015-06-28 16:50:46	2015-06-28 16:50:46	1	2736	hours	123
512	2015-06-16 23:04:53	2015-06-16 23:04:53	1	225	LABEL	Elective
1792	2015-06-16 23:04:54	2015-06-16 23:04:54	1	758	LABEL	Explain the distinction between temporal and spati...
2048	2015-06-16 23:04:55	2015-06-16 23:04:55	1	868	LABEL	Evaluate a given interpretation of a social networ...
3584	2015-06-16 23:04:57	2015-06-16 23:04:57	1	1520	LABEL	Describe what is a multi-valued dependency and wha...
4352	2015-06-16 23:04:58	2015-06-16 23:04:58	1	1864	LABEL	Describe how data distribution/layout can affect a...

Figure 22 – Nodes PROPERTIES storage.

Example of how nodes association stored inside table is shown in Figure 23.

id	created	updated	item_status	relation_type	node_a_id	node_b_id
26	2015-06-16 23:04:52	2015-06-16 23:04:52	1	ASSOC	17	27
15	2015-06-16 23:04:52	2015-06-16 23:04:52	1	ASSOC	2	16
16	2015-06-16 23:04:52	2015-06-16 23:04:52	1	ASSOC	1	17
17	2015-06-16 23:04:52	2015-06-16 23:04:52	1	ASSOC	17	18
18	2015-06-16 23:04:52	2015-06-16 23:04:52	1	ASSOC	17	19
19	2015-06-16 23:04:52	2015-06-16 23:04:52	1	ASSOC	17	20
20	2015-06-16 23:04:52	2015-06-16 23:04:52	1	ASSOC	17	21
21	2015-06-16 23:04:52	2015-06-16 23:04:52	1	ASSOC	17	22
22	2015-06-16 23:04:52	2015-06-16 23:04:52	1	ASSOC	17	23

Figure 23 – Node association storage

4.14 QLearn

QLearn is a learning management system that is built to put different parts of ILT project together, the learning outcomes and learning object repository, student profile, and the recommendation engine; below are screenshots from QLearn system.

In the main screen shot of QLearn, the learner can enter keywords of the required learning object to learn in the box below to get relevant results.

Qlearn Main Page

The screenshot in Figure 24 show the QLearn System main page:

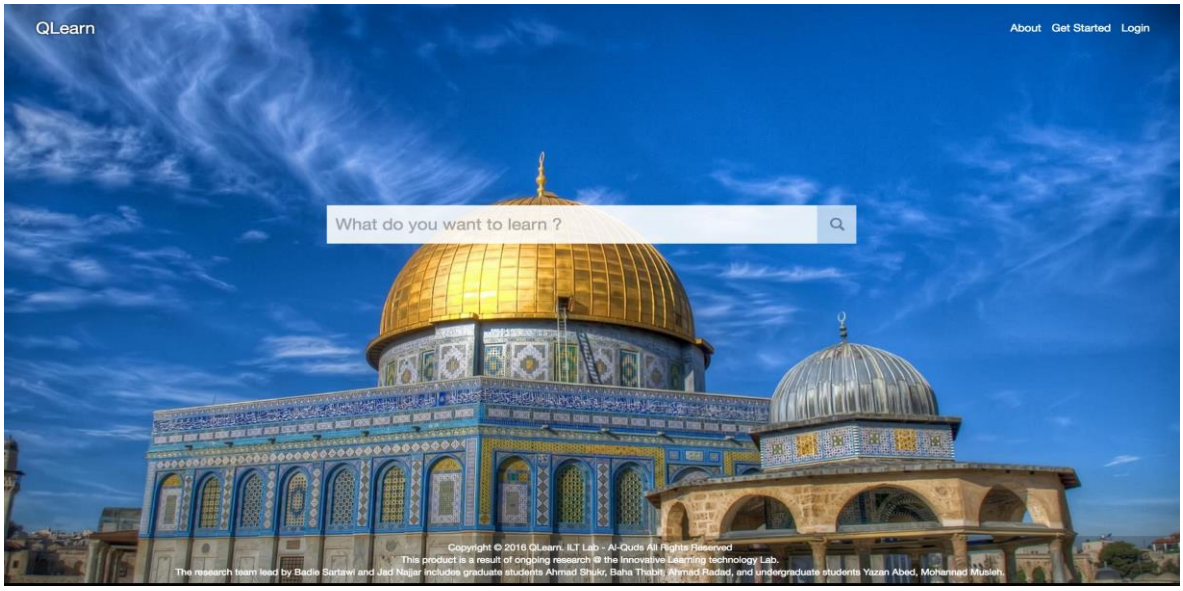


Figure 24 – QLEARN main page

The login screen, where the learner can login to QLearn using his email and password as in Figure 26

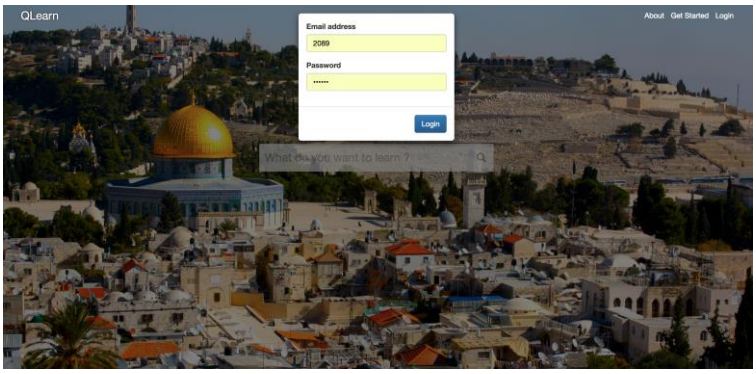


Figure 25 – Login page

The registration page, where the learner can register for new account is given in Figure 26.

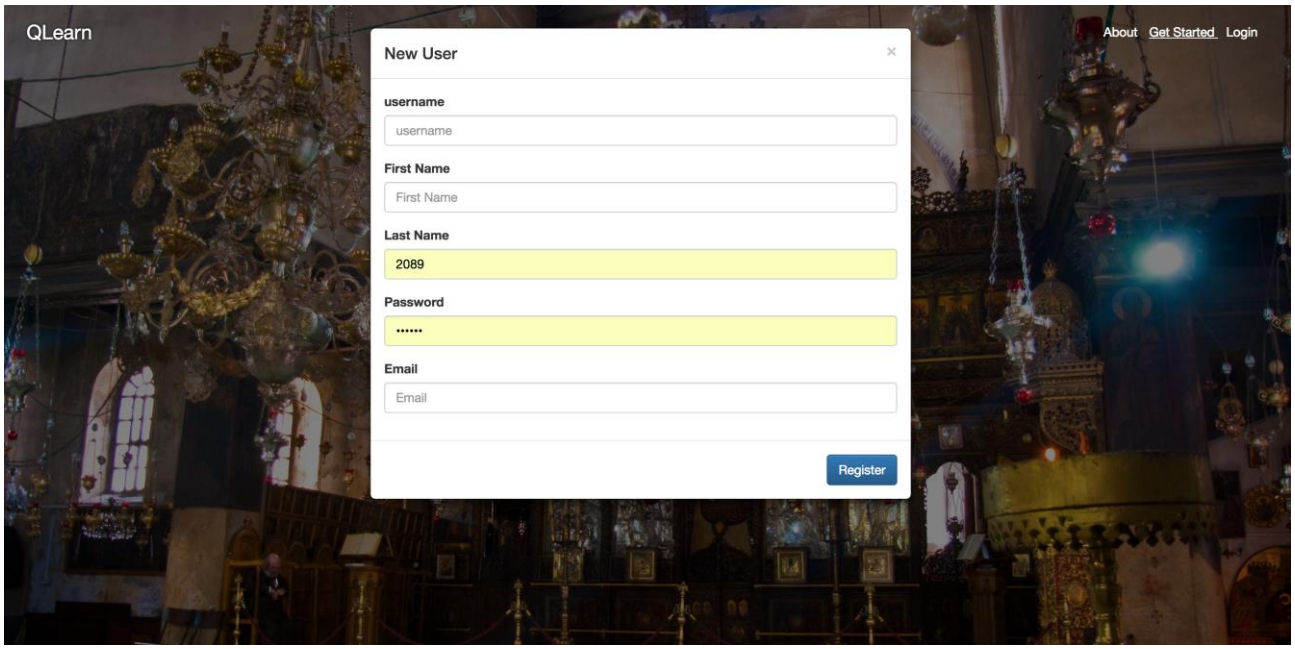


Figure 26 – Registration page

Add new course

The screens in Figure 27 – 30 show how the teacher can add new course, please note that teacher must have special permission in order to enter teacher pages.

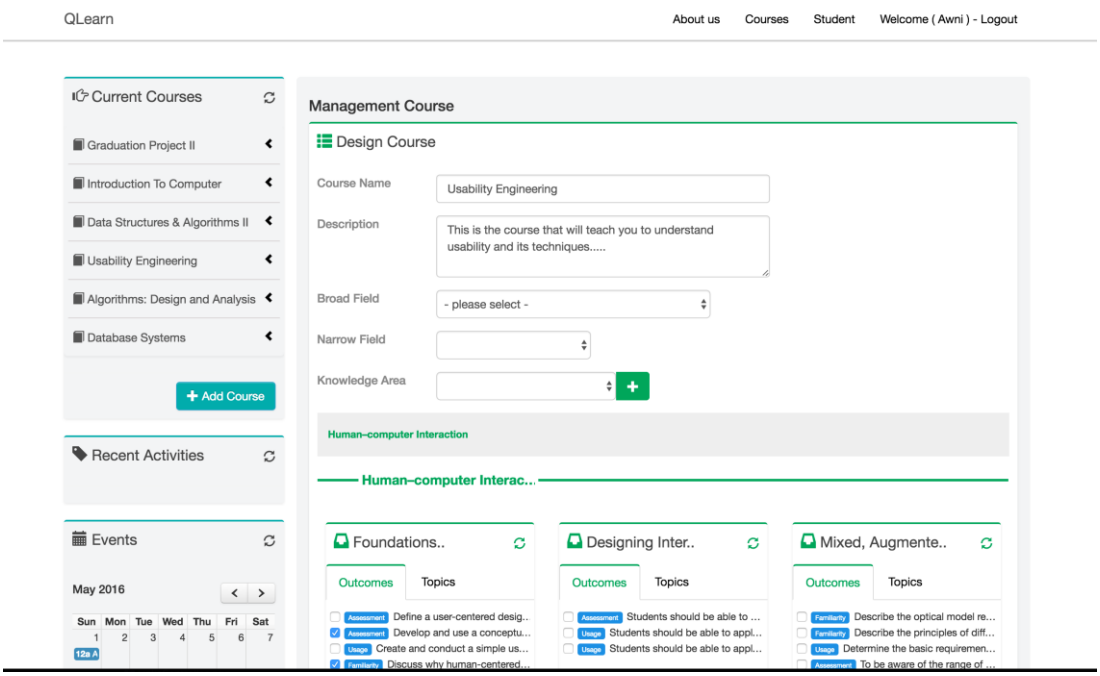


Figure 27 – Course management page, course properties and outcomes selection

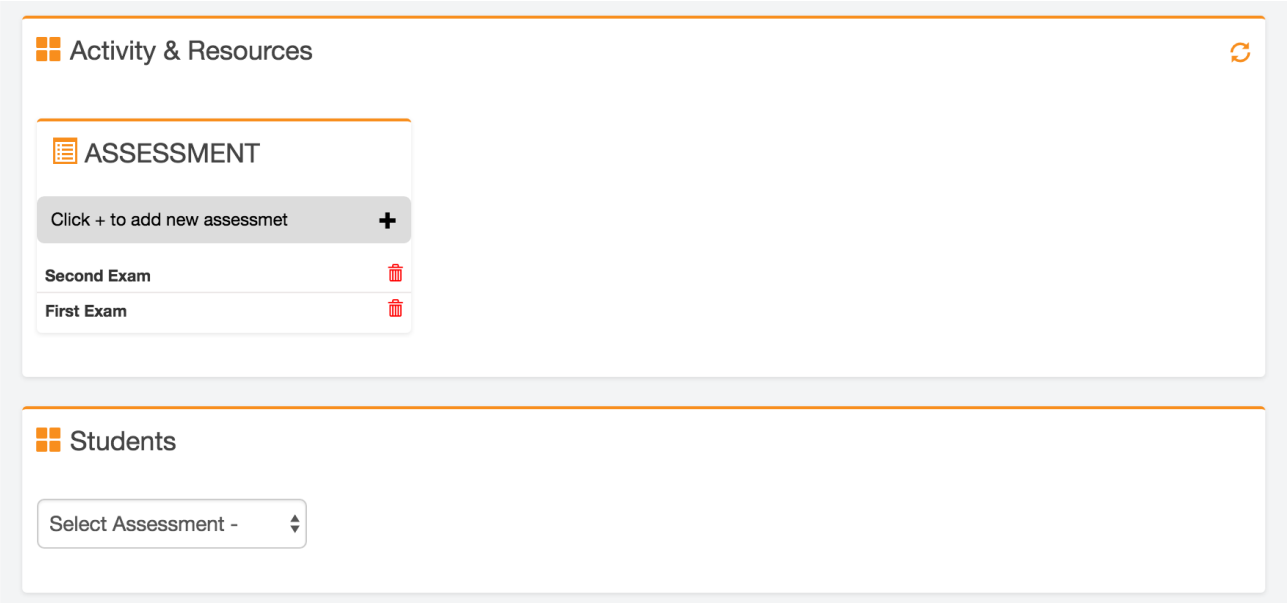


Figure 28 – Assessment and student management

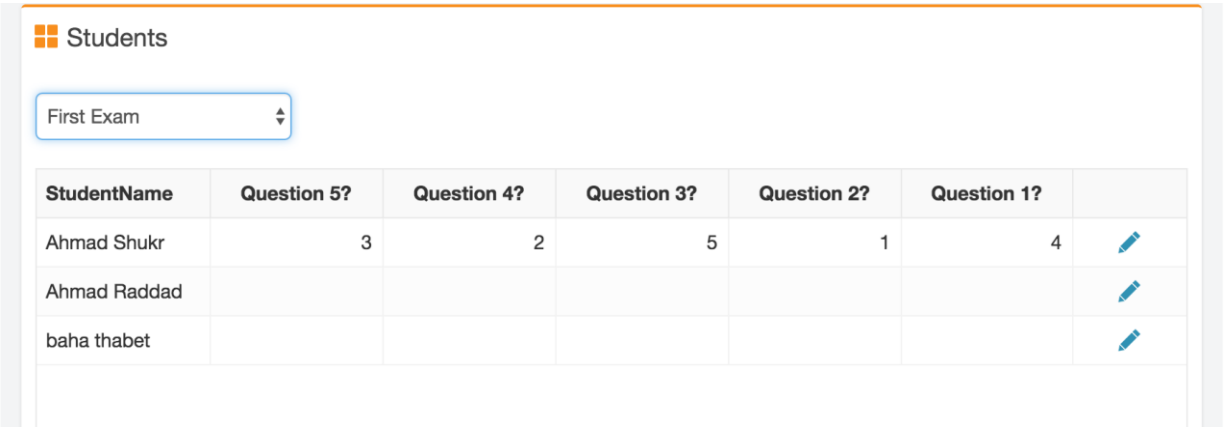


Figure 29 – where teacher can update students marks for evaluation

Add New Learning Object for An Outcome

To link learning outcome with a learning object, the user first clicks on the learning outcome where he want to add learning objects to, then the system show modal window with list of the current linked learning objects. When the teacher clicks on one of these learning object the system redirect him to the learning object source such as image or book or video etc., then if he want to add new learning object the system show him simple form where he can add learning object attributes, see the screenshots below, see Figure 30 and Figure 31.

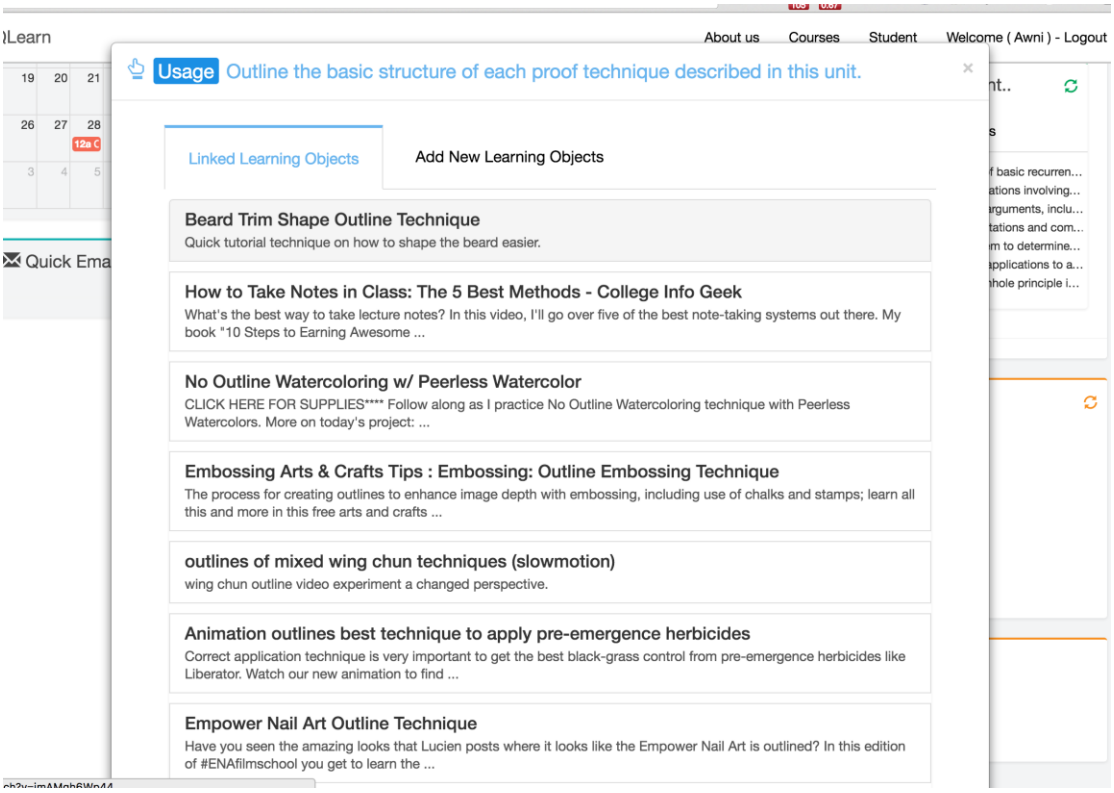


Figure 30 - currently linked learning objects to learning outcome

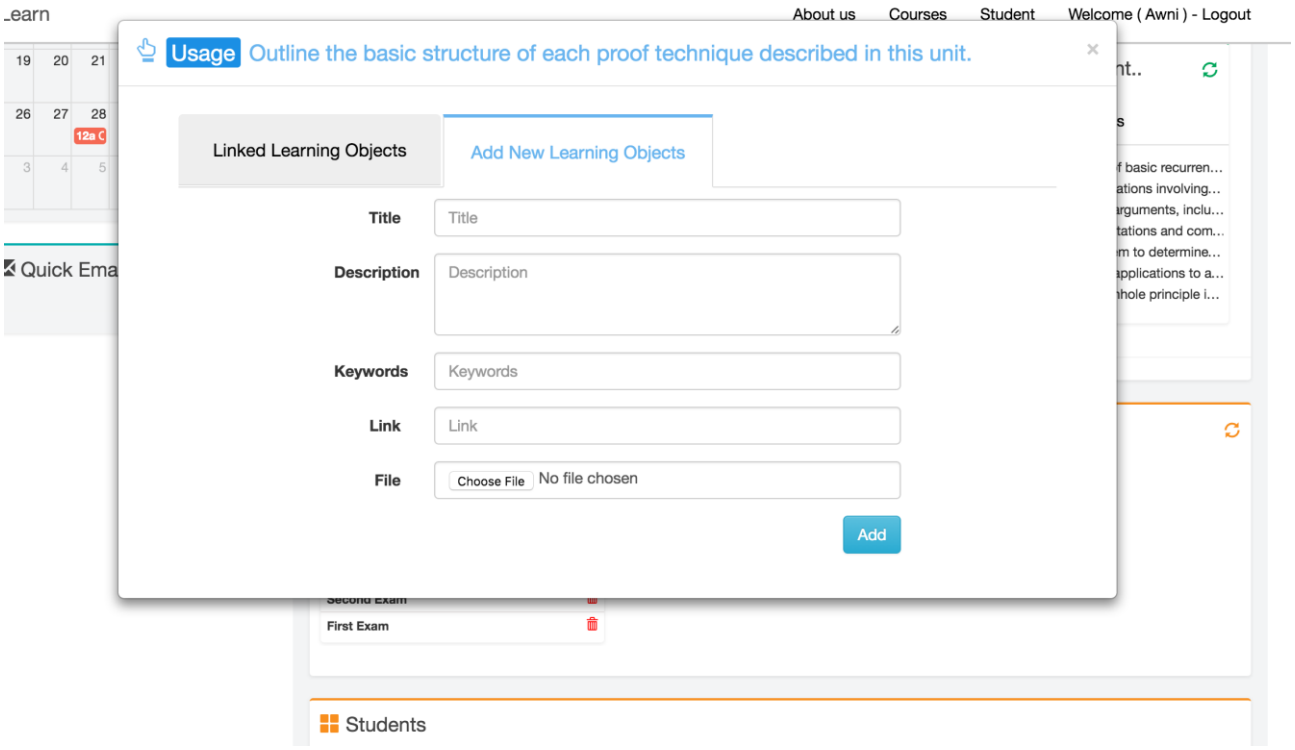


Figure 31 - add new learning object for an outcome

Assessments Management

Assessments are major part of how we evaluate the student in order to recommend new learning objects, in Figure 32 – 35 shows where the teacher can add new assessment policy e.g. exam, then he chooses which outcomes this assessment cover, then the teacher can assign questions for each outcome.

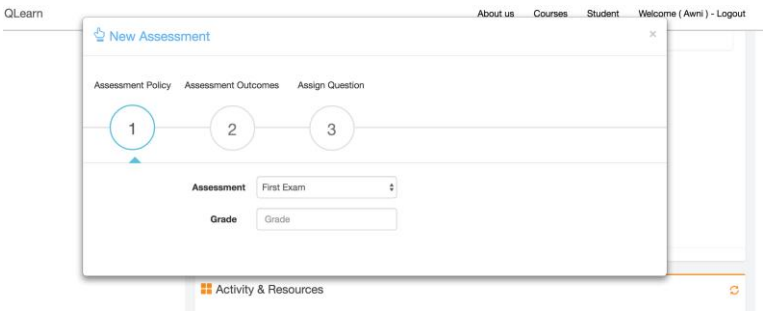


Figure 32 – Add new assessment page

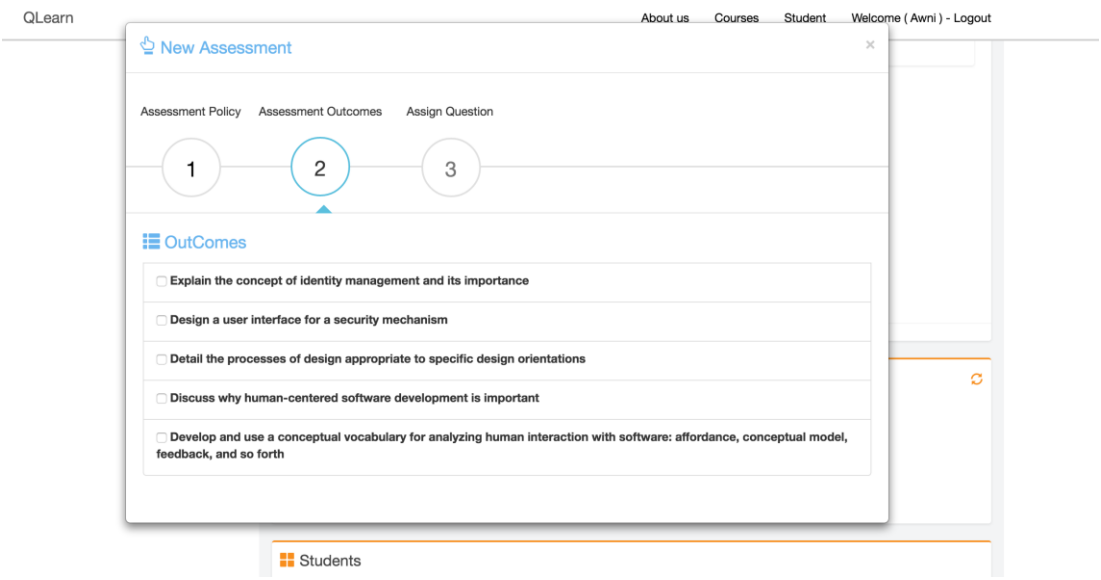


Figure 33 – Assessment outcome selection

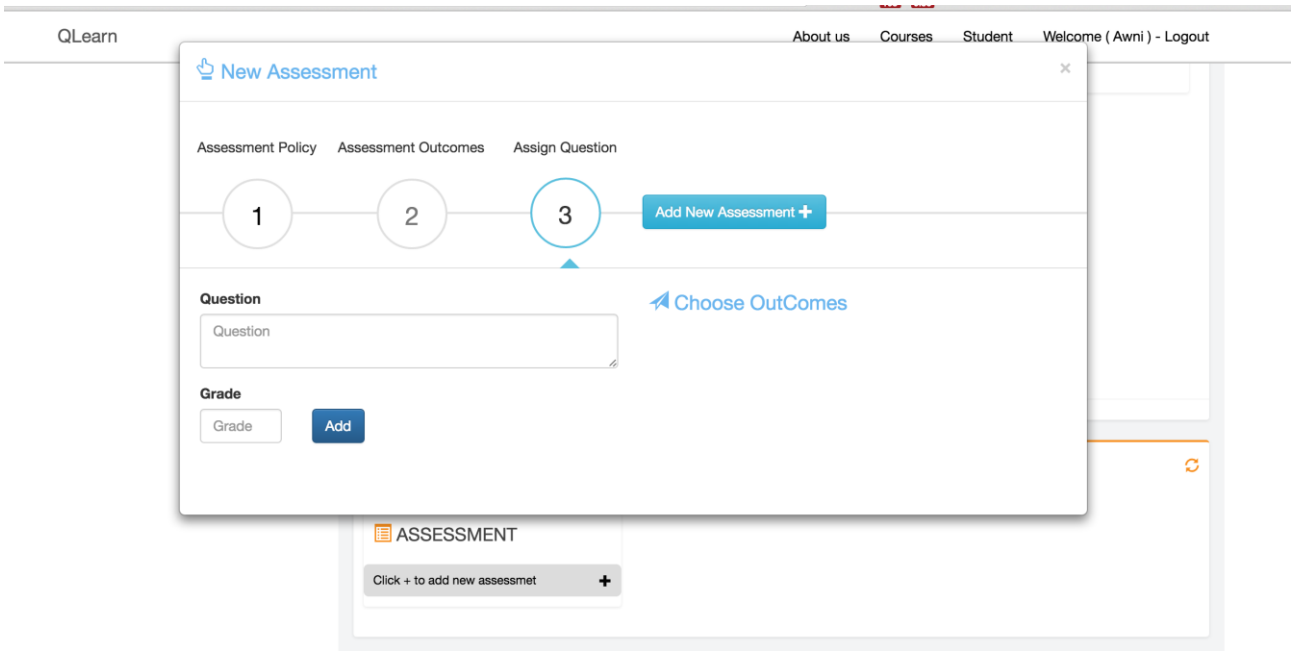


Figure 34 –assessment questions

Student Page

The screenshots Figure 35 - 38 show the student page, where we recommend to the learner what to learn based on his pervious history using QLearn recommendation algorithm, Also we show Learning outcomes progress and how he perform in different courses.

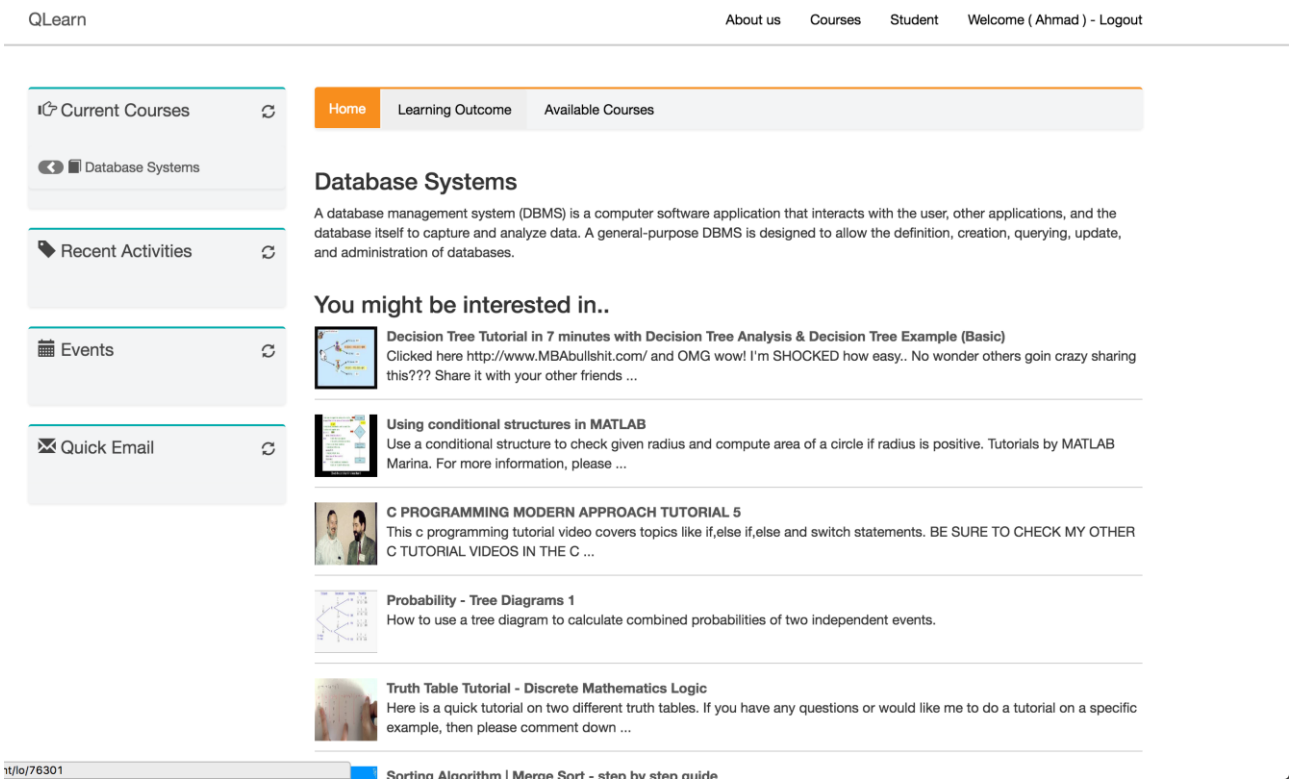


Figure 35 – Student recommended learning objects sample

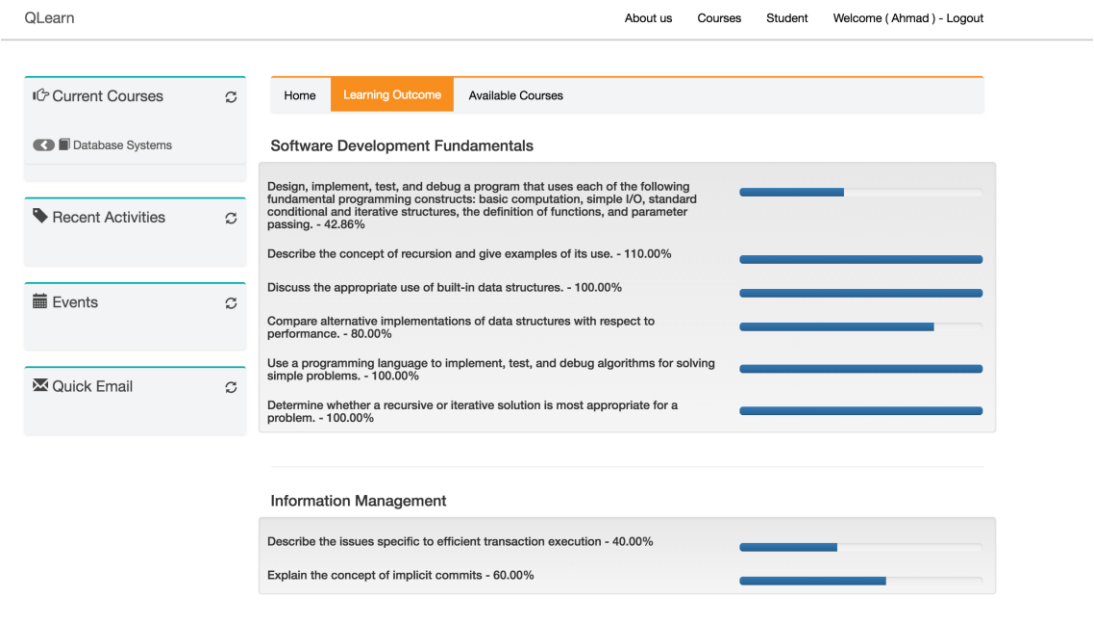


Figure 36 – Student learning outcomes progress

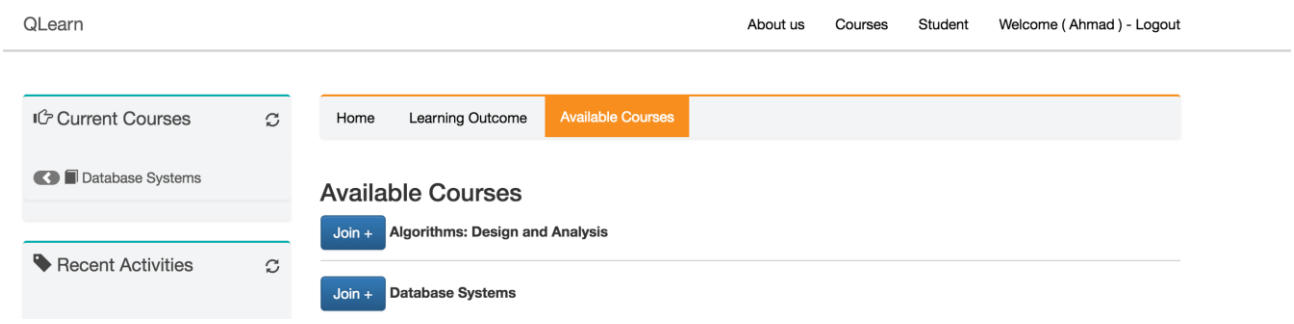


Figure 37 – join available courses

Chapter Five

Evaluation

In this chapter, we will show how we validate our work:

- Validate the linking between learning objects and learning outcomes.
- Validate the interface by conducting user test on small set of users (following discount usability technique [39]).

5.1 Approach

To evaluate our assumption and our software design and architecture:

- We wrote script to import data into our database with UNESCO tree of knowledge and full ACM computer science program [40].
- Then we linked ACM computer science curriculum outcomes automatically with learning objects, these objects were fetched from external web services using the algorithm we described before.

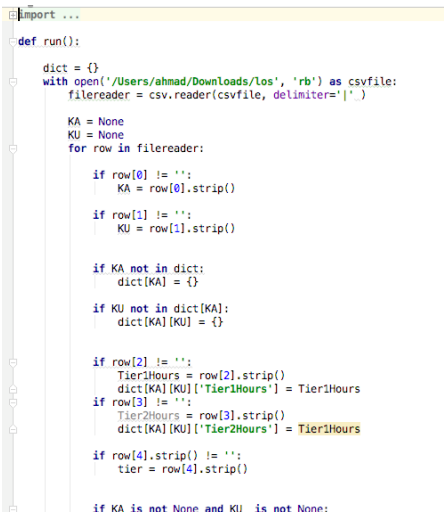


Figure 38 – Importing script code snippet

Figure 39-44 shows learning outcome and linked learning objects.

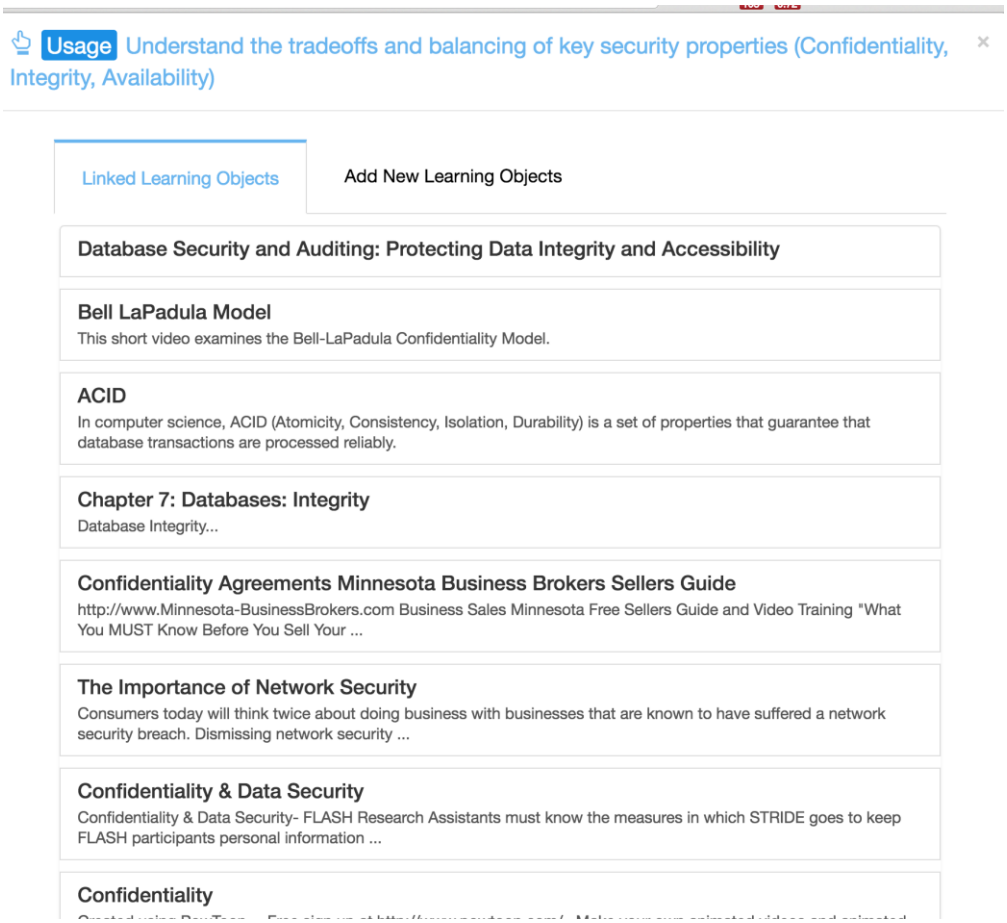


Figure 39 – Auto linked learning objects and learning outcome as part of evaluation process.

```
    for ka in dict:

        node_ka = Node.createNode('KA')
        if ka:
            NodeProperties.prop(node_ka, 'LABEL', ka)
        for ku in dict[ka]:

            node_ku = Node.createNode('KU')
            if ku:
                NodeProperties.prop(node_ku, 'LABEL', ku)
            NodeAssoc.create_assoc(node_ka, node_ku)

            if 'topics' in dict[ka][ku]:
                topics = dict[ka][ku]['topics']

                for topic_title in topics:
                    tier = topics[topic_title]['tier']
                    title = topics[topic_title]['title']

                    sub_topics = topics[topic_title]['sub_topic']
                    node_topic = Node.createNode('TOPIC')

                    NodeProperties.prop(node_topic, 'LABEL', title)
                    NodeProperties.prop(node_topic, 'TIER', tier)

                    for sub_topic in sub_topics:
                        sub_topic_node = Node.createNode('SUB_TOPIC')
                        NodeProperties.prop(sub_topic_node, 'LABEL', sub_topic)
                        NodeAssoc.create_assoc(node_topic, sub_topic_node)

                    NodeAssoc.create_assoc(node_ku, node_topic)
```

Figure 40 – Create associations and categorized the data.

```
for learning_outcome in learning_outcomes:
    learning_outcome_node = lod_app.models.Node.objects.get(id=learning_outcome['key'])
    keywords = learning_outcome['title'].split(' ')
    top4_keywords = sorted(keywords, key=len)[-2:-1]
    query_text = ' '.join(top4_keywords)

    # items = get_by_keywords(query_text) # THIS for videos
    print "querying"
    items = search_ariadne(query_text, size=100)
    for item in items:

        """
        print "Import video: ", item['snippet']['title']
        learning_object = lod_app.models.Node.createNode('LEARNING_OBJECT')
        lod_app.models.NodeProperties.prop(learning_object, 'TITLE', item['snippet']['title'])
        lod_app.models.NodeProperties.prop(learning_object, 'DESC', item['snippet']['description'])
        lod_app.models.NodeProperties.prop(learning_object, 'IMAGE', item['snippet']['thumbnails']['medium']['url'])
        lod_app.models.NodeProperties.prop(learning_object, 'LINK', 'https://www.youtube.com/watch?v='+item['id']['videoId'])
        lod_app.models.NodeAssoc.create_assoc(learning_outcome_node, learning_object)
        """

        print "Import Ariadne object: ", item['title']

        learning_object = lod_app.models.Node.createNode('LEARNING_OBJECT')
        lod_app.models.NodeProperties.prop(learning_object, 'TITLE', item['title'])
        lod_app.models.NodeProperties.prop(learning_object, 'DESC', item['description'])
        lod_app.models.NodeProperties.prop(learning_object, 'KEYWORDS', ', '.join(item['keywords']))
        lod_app.models.NodeProperties.prop(learning_object, 'LINK', item['link'])
        lod_app.models.NodeAssoc.create_assoc(learning_outcome_node, learning_object)

        # time.sleep(100/1000000.0)
```

Figure 41 – Auto linking learning object and learning outcome code snippet

```
def search_ariadne(q, size=100):

    ret = []
    size = str(size)
    r = requests.get('http://ariadne.grnet.gr/globe_search/v1/akif?q='+q+'&facets=set,language,contexts,learningResourceTypes,endUse
    response_dict = r.json()
    results = response_dict['results']
    print 'Ariadne returned:'+ str(results.__len__())+' results.'
    for result in results:
        if 'en' in result['languageBlocks']:
            try:
                ret.append({
                    'title': result['languageBlocks']['en']['title'],
                    'keywords': result['languageBlocks']['en']['keywords'],
                    'description': result['languageBlocks']['en']['description'],
                    'link': result['expressions'][0]['manifestations'][0]['items'][0]['url'],
                })
            except:
                pass


    return ret
```

Figure 42 – search Ariadne [42] web service for learning objects by keywords

```
def get_by_keywords(keywords):
    DEVELOPER_KEY = settings.YOUTUBE_DEVELOPER_KEY
    YOUTUBE_API_SERVICE_NAME = settings.YOUTUBE_API_SERVICE_NAME
    YOUTUBE_API_VERSION = settings.YOUTUBE_API_VERSION
    YOUTUBE_API_MAX_RESULTS = settings.YOUTUBE_API_MAX_RESULTS

    youtube = build(
        YOUTUBE_API_SERVICE_NAME,
        YOUTUBE_API_VERSION,
        developerKey=DEVELOPER_KEY)

    search_response = youtube.search().list(
        q=keywords,
        part="id,snippet",
        maxResults=YOUTUBE_API_MAX_RESULTS
    ).execute()

    videos = []

    for search_result in search_response.get("items", []):
        if search_result["id"]["kind"] == "youtube#video":
            videos.append(search_result)

    return videos
```

Figure 43 – search YouTube[41] web service for learning objects by keywods

The imported data appear in the figure below:

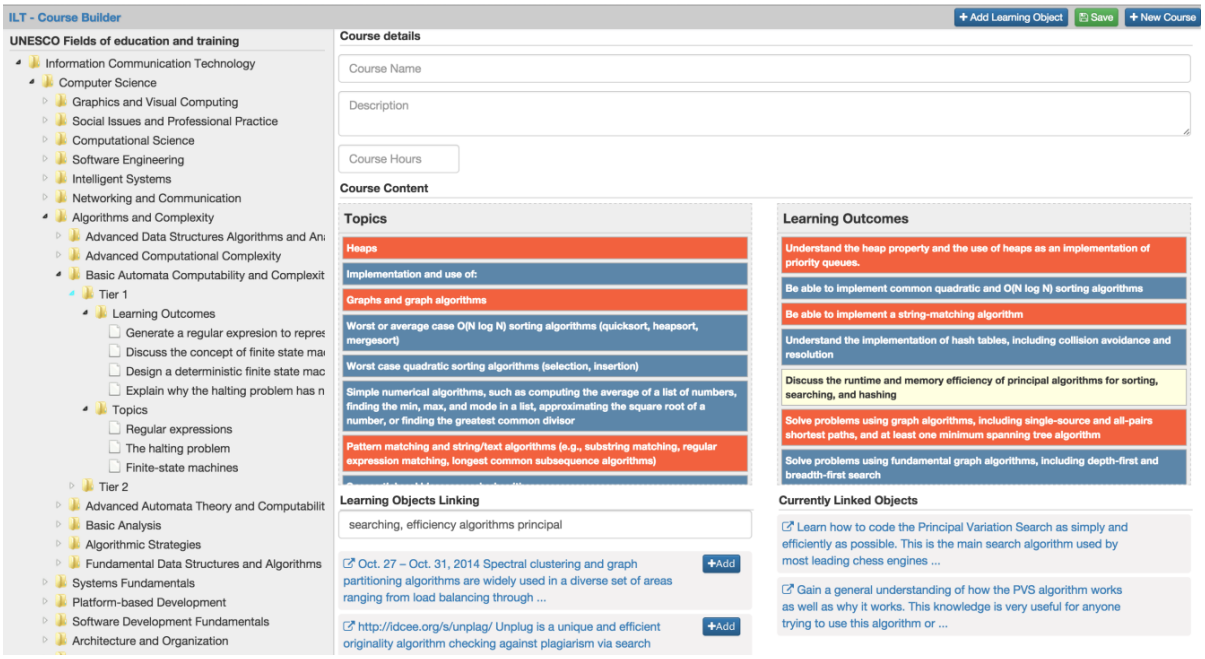


Figure 44 – UNESCO Tree of knowledge and ACM computer science program

For each entry in the data there is a linked topics and learning outcomes, simply if you click on any leaf item in the tree, the topics and learning outcomes will display in the boxes as in the figure 27.

To link learning outcome to a learning object, simply you click on the learning outcome, and the system will suggest list of learning objects, then when you click on add button the object will linked to that learning object as in figure below.

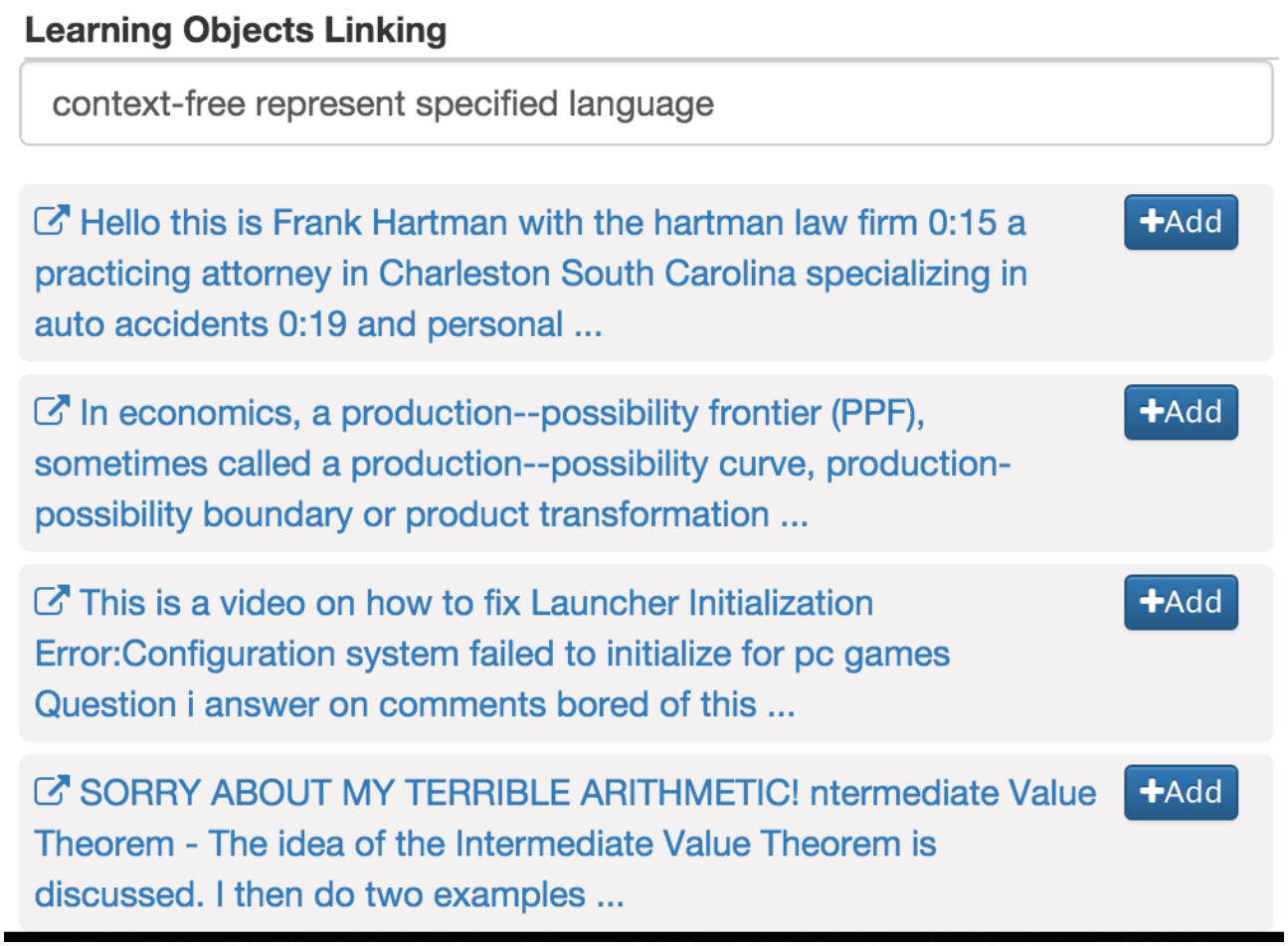


Figure 45 – Learning object search and linking to learning outcomes

When you click on add, the selected learning object will be linked to learning outcome and appeared as in the figure 29.

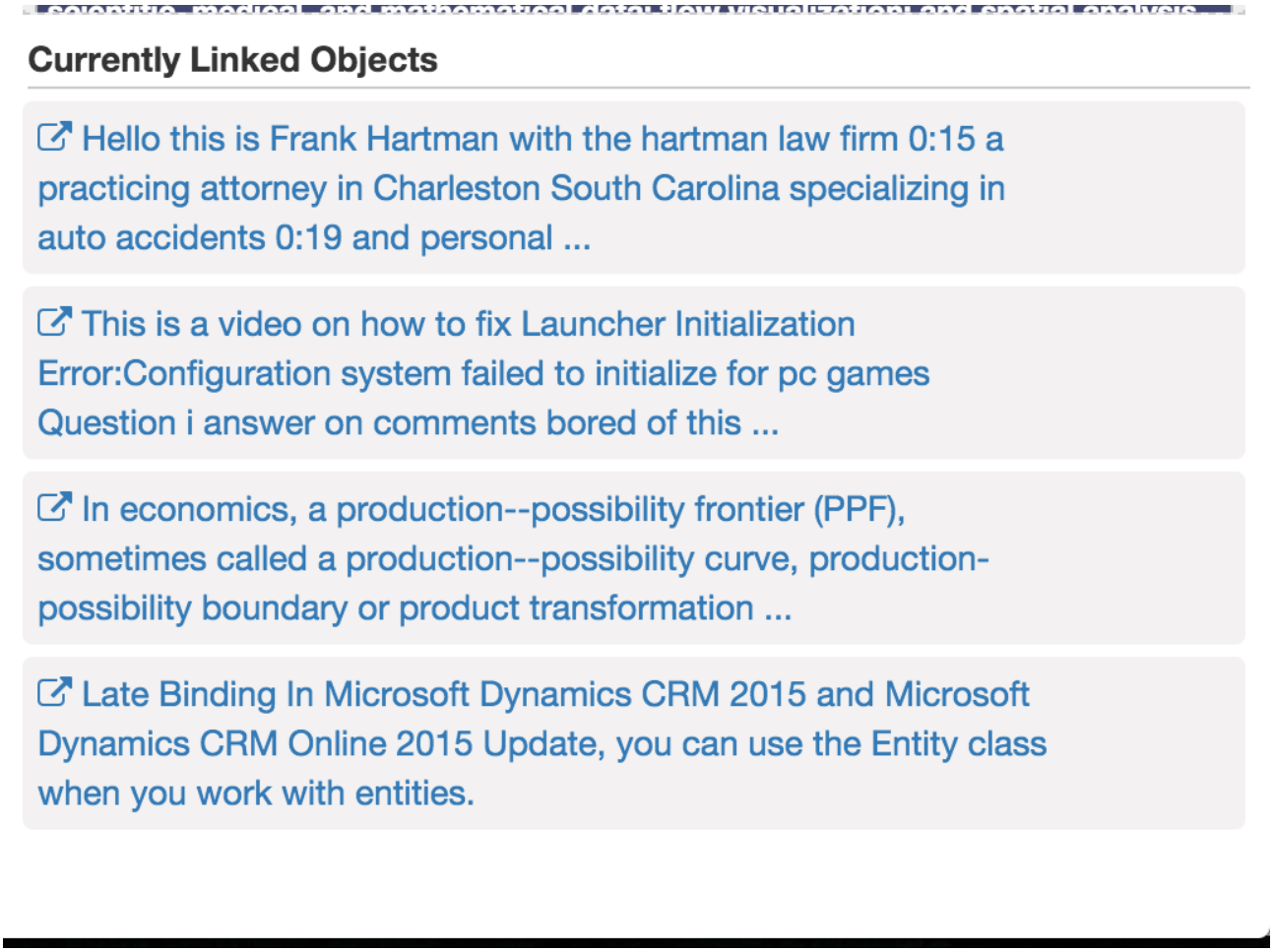


Figure 46 – currently linked learning objects

To preview a learning object, just click on the link and it will open in a separate tab in the browser like the figure below:

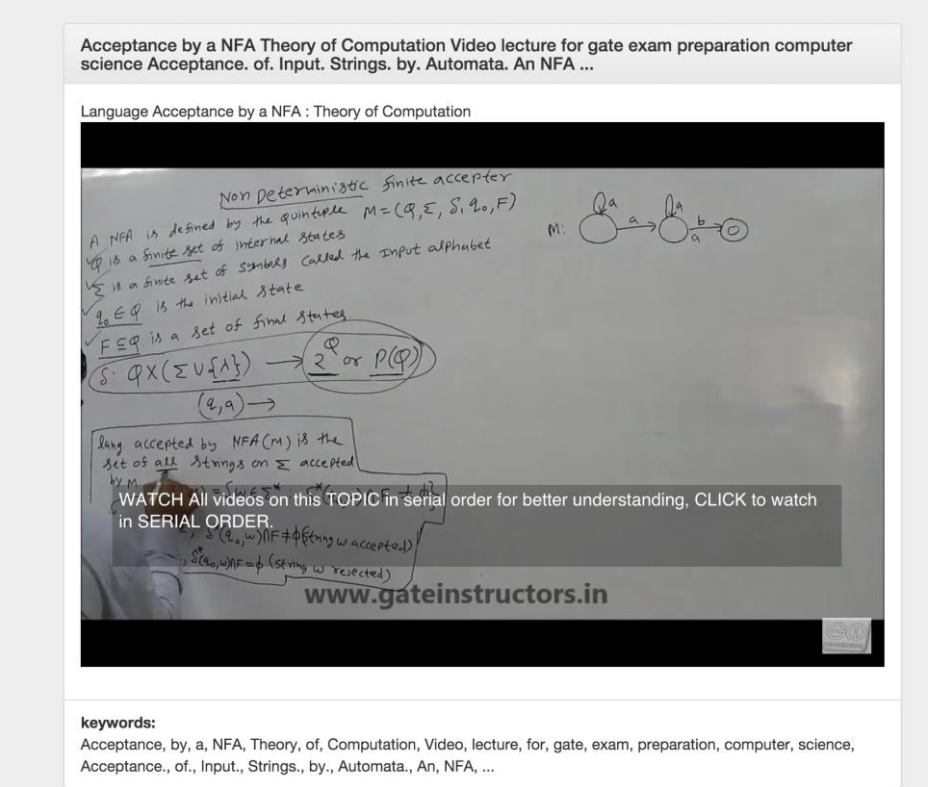


Figure 47 – Learning object preview

5.2 User Evaluation

The purpose of the evaluation conducted with target users of the QLearn system is to ensure that data linkage between learning outcomes and learning objects is working, and stands as proof of concept for our approach. Also to make sure that our user interface is working as expected by users and in the same way they expect it to work. However, thorough and systematical evaluation of the system usability is out of scope of this work.

The evaluation is designed as follows:

- 11 user evaluation sessions are conducted, with about 30 minutes for each participant; six teachers and five students.
- The evaluation sessions were conducted either at the working place of the participants or online; in all sessions screen recording was used, and notes were taken.

- After the usability sessions, working with QLearn, the participants were asked to fill in a survey with predefined questions; shown in Table 5 and Table 6 in the Appendix.
- All participants are not related to QLearn or ILT research project, and they volunteer in the experiment without any kind of pressure.

In our experiment we followed the "Discount usability", where we conduct the experiment under fewer resources and time than formal usability testing technique. This technique is the most effective method to measure the validity and usefulness of our system and approach with very cost-effective. The tools we used are:

1) Think-aloud: simply we asked the participant to verbalizing their thoughts as they move through our system so we can understand how they think.
2) Heuristic evaluation [43]: we covered these points in our heuristic evaluation
    a. Visibility of system status: such as asynchronous calls indicators.
    b. Error prevention: for example in registration form when you entered wrong data we will show you how to correct it.
    c. Minimalist design: only we displayed the relevant information in modals, forms, and other user interface components.
3) Facilitator notes: All sessions were recorded with user voice, this allow us to revise the session and take extra notes and analyze user behavior on the system.
4) Questionnaire: After the session we asked users to answer a survey with questions designed to cover all session parts, we followed answers for each question was strongly disagreed, disagree, neutral, agree, and strongly agree.

5.3 Teachers

The teachers participated were with different academic levels; holding PHD, Master, and BA degree. Each participant was asked to walk through the system and accomplish 12 tasks (shown in figure below).

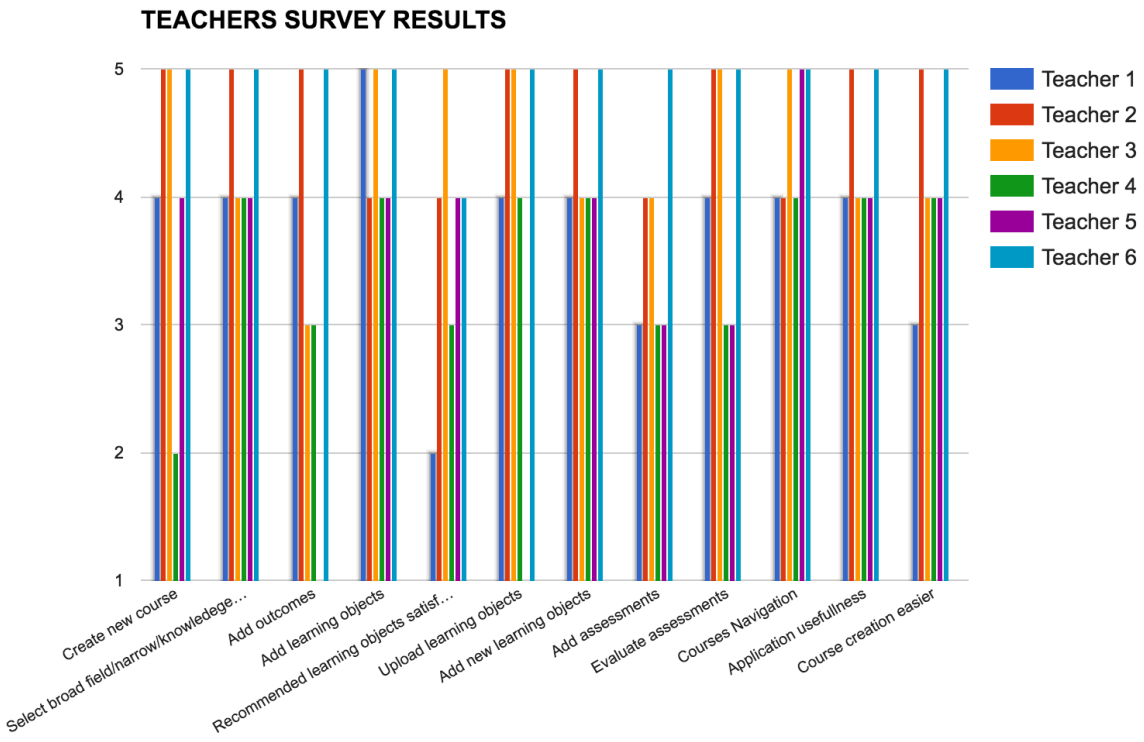


Figure 48 – User Evaluation for teachers survey answers, The scale we used from 1-5, where 1 is Strongly disagree, 2 is disagree, 3 neutral, 4 is agree, 5 is strongly agree

About 84% of teachers were able to create new course, while they had some comments such as we should add another level under outcomes called objectives like how to achieve a learning outcome. We also noticed that they had some difficulties in identifying the role of each field in the first section of the page. All users were able to select (Broad Field/Narrow Field/Knowledge Area), they had some concerns such as if this is only for CS/IT fields or for general purposes, also, they prefer to order the select fields apathetically, in addition some of them suggested choosing to better name than “Broad field, Narrow Field,

Knowledge area", also all of them had confusion what to do next after choosing the fields, and the last comment is really serious and it should be fixed in the next version.

About 50% were able to choose learning outcomes from checkboxes, but majority had obvious difficulties to open learning objects modal, but once they open it, all of them were able to add learning objects from their machines. All of the users were satisfied with auto recommended learning objects; while we believe it's still need more work to improve the quality of the recommended learning objects.

Most of the teachers were able to add assessments with some with some difficulties, but they all agreed that multiple choices exams are not enough to evaluate a student. In addition to this, about 66% were able to evaluate students, and all of them were able to navigate through courses easily.

To summarize the experiment, The UI should be improved based on teachers feedback to enhance the usability of the system, but the good point that all of them said the system will is useful for them and will make the process of creating new course much easier.

5.3 Students

We asked 5 students to use the system from student's side; the following questions:

And the results were as the following:

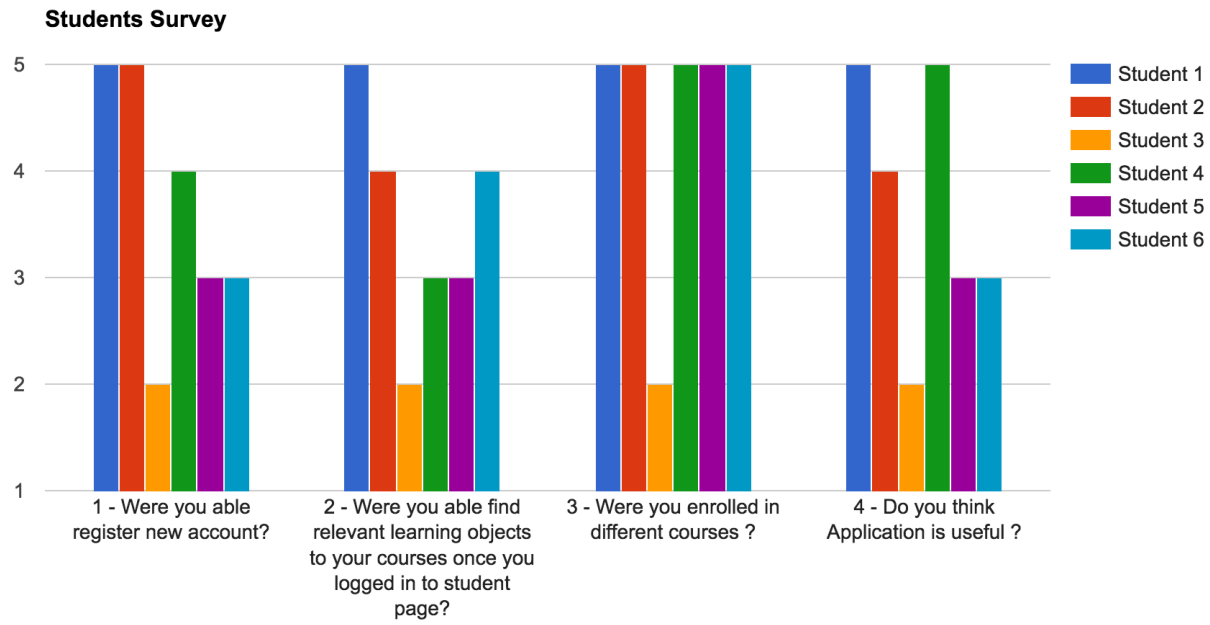


Figure 49 - Students survey results, we used the same scale from 1-5 as we used in teachers survey.

Half of the students were able to register new account, the next half were able to register with few difficulties such as there is no activity indicator when making asynchronous requests to web server, enter key is working in some fields and not in other fields. The same results were found for finding relevant learning objects once they requested the page, other said there the application is slow in some areas, this is mainly due to the underlying server is small plus we have to implement some cache techniques to improve server response speed. In addition to this, all of the students were able to join new courses. Moreover, 50% of students said this application would be useful for them while, 33% were neutral, and 16% disagreed.

5.4 Algorithm Evaluation

In this section, we will evaluate our algorithm for auto linking between learning objects and learning outcomes.

We parsed ACM Compute Science curriculum, which contain about 1052 learning outcome, And to fine tuning algorithm parameters (Number of keywords and Levenshtein distance), we randomly took 1 to 4 longest keywords from these learning outcomes and set the maximum allowed Levenshtein distance randomly between 1 and 4, then we show sample of results to expert in computer science, to evaluate the relevance between learning objects and learning outcomes.

Several studies [44] suggest that sample size should be around 30 for qualitative research, so we randomly picked 30 learning outcomes with its auto linked learning objects for each Levenshtein distance and number of keywords as in Figure 50, full results details found in the appendix.

Figure 50 show the relation between number of keywords and Levensitien distance and the results by expert on 1 to 5 scale where 1 (strongly disagree) to 5 (strongly agreed).

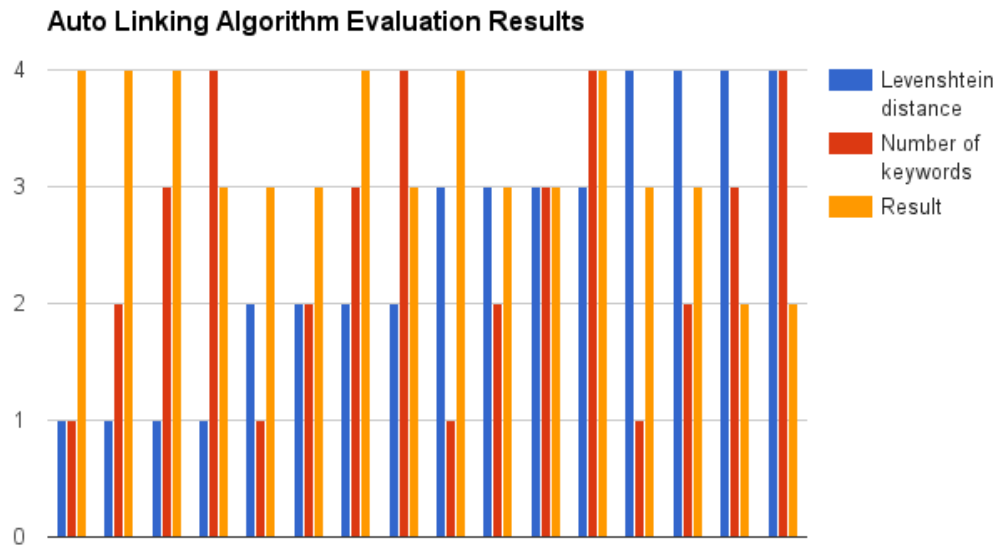


Figure 50 – Expert results and its relation with number of keywords and Levenshtein distance

In the figure above the results for given number of keywords taken from a learning outcome string compared with learning object title at given allowed Levenshtein distance, for example in the first group, the result is “agree” by the expert when the number of keywords is one and Levenshtein distance threshold is one, while the result is neutral when Levenshtein distance threshold is four and number of keywords is four.

Chapter Six

Conclusion

6.1 Introduction

In this research, we designed learning objects repository based on IEEE LOM, and we designed a learning outcomes repository based on IEEE RCD, each of the learning outcome is linked internally in the system to one or more learning objects.

After that we implemented the design with a web tool that enable learners to add learning outcomes and topics for the created course, and from the same page the learner can search and link learning outcomes to existing learning objects or creating a new one. Then we validated our assumption by importing real learning objects and auto link them with real learning outcomes from different sources. After that, we conduct an experiment on group of teachers and students to measure the usefulness of the system for the actual users, the experiment show how users interacted with our system and the areas that need to be improved in order to enhance our user interaction with our system, also we evaluated the auto linking algorithm with sufficient samples for each algorithm parameter space.

6.2 Main Results

1. We designed new QLearn-LOM Application Profile based on IEEE LOM to use it in storing and exchange learning objects
2. We designed new QLearn-LearningOurcomes based on IEEE RCD standard, and then we used it in storing and exchanging learning outcomes.
3. We linked both QLearn-LOM and QLearn-LearningOurcomes, so you can get the learning outcomes for specific learning objects and vice versa.
4. We implemented the new standards then we evaluate the designs using real case data.

5. We designed and implemented and evaluated an auto-linking algorithm between learning objects and learning outcomes.

6.3 Recommendation Further Research

The work presented is a first step towards retrieving learning objects based on relevant and linked learning outcomes. Linked metadata about learning outcomes to the descriptive metadata of the object, like subject and title, would enhance the find ability of the learning objects. Moreover, our proposed solution enabled teachers and students to have a learning system that is outcome based focused, for the design of the course components, adding learning resources, assessment and students' achievements. The recommender engine of this solutions is still be enhanced with more complex criteria and parameters, to be able to filter the relevance of the retrieved objects.

Our implementation of QLearn -LOM and QLearn -LO is based on Relational Database Management System (RDBMS); we think both standards can also work very well with NO-SQL or Graph databases, Also we could use this work to process CV's, so the job matching system can analyze or determine the CV's of applicants based on the content and recommend set of CVs for specific job based on the CV achieved outcomes or what the candidate can do.

References

[1] N. Smith, M. Van Coillie, and E. Duval, “Guidelines and Support for Building Application Profiles in E-Learning,” Proc. CEN/ISSS WS/LT Learning Technologies Workshop CWA, pp. 1-26, 2006.

[2] IMS Global Learning Consortium (GLC), “IMS Application Profile Guidelines Overview”, http://www.imsglobal.org/ap/apv1p0/imsap_oviewv1p0.html.

[3] D. G. Sampson, P. Zervas, G. Chloros, "Supporting the Process of Developing and Managing LOM Application Profiles: The ASK-LOM-AP Tool," IEEE Transactions on Learning Technologies, vol. 5, no. 3, pp. 238-250, Third Quarter, 2012.

[5] Learning Outcomes Definitions http://wiki.teria.no/display/inloc/LOD retrieved on 5/Mar/2015.

[6] IEEE Learning Technology Standards Committee (LTSC) http://ieeeltsc.org/wg12LOM/ retrieved on 5/Mar/2015.

[7] Dublin Core http://dublincore.org/ retrieved on 5/Mar/2015.

[8] Luciana A. M. Zaina, Jose F. Rodrigues, Jr, and Gra a Bressan. 2010. An approach to design the student interaction based on the recommendation of e-learning objects. In Proceedings of the 28th ACM International Conference on Design of Communication (SIGDOC '10). ACM, New York, NY, USA, 223-228. DOI=10.1145/1878450.1878488 http://doi.acm.org/10.1145/1878450.1878488

[9] CEN/the European Committee for Standardization https://www.cen.eu/Pages/default.aspx retrieved on 20/May/2015.

[10] IMS Global Learning Consortium http://www.imsglobal.org/ retrieved on 20/May/2015.

[11] IGI global publisher http://www.igi-global.com/dictionary/learning-objects/16918 retrieved on 20/May/2015.

[13] Tuning Educational Structures, Guide to the Process v1.0, ©2010 The Institute for Evidence-Based Change.

[14] Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society, December 20, 2013.

[15] Najjar, J.; Simon, B., "Learning Outcome Based Higher Education: iCoper Use Cases," Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on , vol., no., pp.718,719, 15-17 July 2009

[16] Jad Najjar, Michael Derntl, Tomaž Klobucar, Bernd Simon, Michael Totschnig, Simon Grant, and Jan Pawlowski. 2010. A Data Model for Describing and Exchanging Personal Achieved Learning Outcomes PALO. Int. J. IT Stand. Stand. Res. 8, 2 (July 2010), 87-104.

[17] Filip Neven and Erik Duval. 2002. Reusable learning objects: a survey of LOM-based repositories. In Proceedings of the tenth ACM international conference on Multimedia (MULTIMEDIA '02). ACM, New York, NY, USA, 291-294.

[18] Thomas Zschocke, Jan Beniest, Courtney Paisley, Jehad Najjar, and Erik Duval. 2009. The LOM application profile for agricultural learning resources of the CGIAR. Int. J. Metadata Semant. Ontologies 4, 1/2 (May 2009), 13-23.

[19] S. Chatzinotas and D. Sampson, "eMAP: design and implementation of educational metadata application profiles," Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on, 2004, pp. 876-877.

[20] Tim Berner Lee http://inventors.about.com/od/bstartinventors/p/TimBernersLee.htm retrieved on 15/Jan/2016.

[21] IEEE. (2002). Draft standard for Learning Object Metadata (LOM) (IEEE 1484.12.1-2002). IEEE Learning Technology Standards Committee (LTSC). Retrieved July 7, 2007, from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

[22] Usability Evaluation of Learning Object Indexation: The ARIADNE Experience. Najjar, J. / Klerkx, J. / Ternier, S. et

[23] IEEE LOM final draft http://grouper.ieee.org/groups/ltsc/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf retrieved on 31/Mar/2016.

[24] Dublin Core elements http://dublincore.org/documents/usageguide/elements.shtml retrieved on 31/Mar/2016.

[25] UK LOM CORE http://zope.cetis.ac.uk/profiles/uklomcore/ retrieved on 31/Mar/2016.

[26] CanCore http://cancore.tru.ca/en/guidelines.html retrieved on 31/Mar/2016.

[27] https://en.wikipedia.org/wiki/Learning_object_metadata retrieved on 31/Mar/2016.

[28] Python Programming Language https://www.python.org/ retrieved on 1/Apr/2016.

[29] Roy Thomas Fielding. 2000. Architectural Styles and the Design of Network-Based Software Architectures. Ph.D. Dissertation. University of California, Irvine. AAI9980887.

[30] Guidelines for Implementation of REST, National Security Agency(NSA), https://www.nsa.gov/ia/_files/support/guidelines_implementation_rest.pdf retrieved on 1/Apr/2016.

[31] CERN http://home.cern/topics/birth-web

[32] S. Chatzinotas, P. Karampiperis, D. Sampson, "Enhancing Educational Metadata with Digital Rights Information: the EMDiR Web-based Environment", in Proc. of the IADIS WWW/Internet 2004 Conference, ISBN: 9729935300, pp. 779-782, Madrid, Spain, IADIS Press, October 6-9, 2004

[33] https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/ retrieved on 4/June/2016.

[34] Michael Totschnig, Michael Derntl, Israel Gutiérrez Rojas, Jad Najjar, Roland Klemke, Joris Klerkx, Erik Duval, Franz Müller (2010): Repository Services for Outcome-based Learning. In: Search and Exchange of e-learning Materials 2010. Proceedings of the Fourth International Workshop on Search and Exchange of e-learning Materials 2010 (SEAM'10). Barcelona, Spain, September 27-28, 2010.

[35] Moodle https://moodle.org/ retrieved on 4/June/2016.

[36] Elgg https://elgg.org/ retrieved on 4/June/2016.

[37] CLIX https://clix.tiss.edu/ retrieved on 4/June/2016.

[38] LOD http://www.cetis.org.uk/inloc/LOD retrieved on 4/June/2016.

[39] Discount Usability: 20 Years https://www.nngroup.com/articles/discount-usability-20-years/ retrieved on 5/June/2016.

[40] Computer Science Curricula 2013 https://www.acm.org/education/CS2013-final-report.pdf

[41] Youtube https://www.youtube.com/ retrieved on 28/June/2016.

[42] Ariadne http://ariadne.grnet.gr/ retrieved on 28/June/2016.

[43] Heeringa, Wilbert Jan. Measuring dialect pronunciation differences using Levenshtein distance. Diss. University Library Groningen][Host], 2004.

[44] Sample Size and Saturation in PhD Studies Using Qualitative Interviews http://www.qualitative-research.net/index.php/fqs/article/viewArticle/1428/3027 retrieved on 24/Dec/2016.

[45] Filip Neven and Erik Duval. 2002. Reusable learning objects: a survey of LOM-based repositories. In Proceedings of the tenth ACM international conference on Multimedia (MULTIMEDIA '02). ACM, New York, NY, USA, 291-294.

[46] Jehad Najjar and Erick Duval. 2004. INTEROPERABILITY OF LEARNING OBJECT

REPOSITORIES: COMPLICATIONS AND GUIDELINES.

APPENDIX

IEEE LOM

Category	Description
General	This category groups the general information that describes this learning object as a whole.
Life Cycle	This category describes the history and current state of this learning object and those entities that have affected this learning object during its evolution.
Meta-Metadata	This category describes this metadata record itself (rather than the learning object that this record describes). This category describes how the metadata instance can be identified, who created this metadata instance, how, when, and with what references.
Technical	This category describes the technical requirements and characteristics of this learning object.
Requirement	The technical capabilities necessary for usingthis learning object. If there are multiple requirements, then all are required, i.e., the logical connector is AND.
Educational	This category describes the key educational or pedagogic characteristics of this learning object.
Rights	This category describes the intellectual property rights and conditions of use for this learning object.
Relation	This category defines the relationship between this learning object and other learning objects, if any.

Annotation	This category provides comments on the educational use of this learning object, and information on when and by whom the comments were created.
Classification	This category describes where this learning object falls within a particular classification system.

Table 3 - Main categories of IEEE LOM [23]

Dublin Core

Attribute	Description
Title	The name given to the resource. Typically, a Title will be a name by which the resource is formally known.
Subject	The topic of the content of the resource. Typically, a Subject will be expressed as keywords, key phrases, or classification codes that describe the topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.
Description	An account of the content of the resource. Description may include but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.
Type	The nature or genre of the content of the resource. Type includes terms describing general categories, functions, genres, or aggregation levels for content.
Source	A Reference to a resource from which the present resource is derived. The present resource was derived from the Source resource in whole or part. Recommended best

	practice is to reference the resource by means of a string or number conforming to a formal identification system.
Relation	A reference to a related resource. Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system.
Coverage	He extent or scope of the content of the resource. Coverage will typically include spatial location (a place name or geographic co-ordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity).
Creator	An entity primarily responsible for making the content of the resource. Examples of a Creator include a person, an organization, or a service. Typically, the name of the Creator should be used to indicate the entity.
Publisher	The entity responsible for making the resource available. Examples of a Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.
Contributor	An entity responsible for contributing to the content of the resource. Examples of a Contributor include a person, an organization or a service. Typically, the name of a Contributor should be used to indicate the entity.
Rights	Information about rights held in and over the resource. Typically, a Rights element will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the rights element is absent, no assumptions can be made about the status of these and other rights with respect to the resource.

Date	A date associated with an event in the life cycle of the resource. Typically, Date will be associated with the creation or availability of the resource.
Format	The physical or digital manifestation of the resource. Typically, Format may include the media-type or dimensions of the resource.
Identifier	An unambiguous reference to the resource within a given context. Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system. Examples of formal identification systems include the Uniform Resource Identifier (URI) (including the Uniform Resource Locator (URL), the Digital Object Identifier (DOI) and the International Standard Book Number (ISBN).
Language	A language of the intellectual content of the resource.
Audience	A class of entity for whom the resource is intended or useful. A class of entity may be determined by the creator or the publisher or by a third party.
Provenance	A statement of any changes in ownership and custody of the resource since its creation that is significant for its authenticity, integrity and interpretation.
RightsHolder	A person or organization owning or managing rights over the resource.
InstructionalMethod	A process, used to engender knowledge, attitudes and skills, that the resource is designed to support. Instructional Method will typically include ways of presenting instructional materials or conducting instructional activities, patterns of learner-to-learner and learner-to-instructor interactions, and mechanisms by which group and individual levels of learning are measured. Instructional methods include all aspects of the instruction

	and learning processes from planning and implementation through evaluation and feedback.
AccrualMethod	The method by which items are added to a collection.
AccrualPeriodicity	The frequency with which items are added to a collection. Recommended best practice is to use a value from a controlled vocabulary.
AccrualPolicy	The policy governing the addition of items to a collection. Recommended best practice is to use a value from a controlled vocabulary.

Table 4 - Dubline core attributes [24]

Evaluation Questions

Question
1 - Were you able to create new course?
2 - Were you able to select Broad Field/Narrow Field/Knowledge Area?
3 - Were you able to add outcomes to your courses?
4 - Were you able to add learning object to your courses outcomes from the recommended learning objects?
5 - How satisfactory is the recommended and selected object?
6 - Were you able to add learning object to your courses outcomes from your machine?
7 - Were you able to add new learning objects? Were you able to add assessments add to course?
Were you able to evaluate student assessments (add grades)?
Were you able to navigate through courses?
Do you think Application is useful?
Will it make course creation easier?

Table 5 – Teachers evaluation questions

Question
1 - Were you able register new account?
2 - Were you able find relevant learning objects to your courses once you logged in to student page?
3 - Were you enrolled in different courses?
4 - Do you think Application is useful?

Table 6– Students evaluation questions.

Auto Link Algorithm Result

Levenshtein distance	Number of keywords	Expert Evaluation Result
1	1	Agree
1	2	Agree
1	3	Agree
1	4	Neutral
2	1	Neutral
2	2	Neutral
2	3	Agree
2	4	Neutral
3	1	Agree
3	2	Neutral
3	3	Neutral
3	4	Agree
4	1	Neutral
4	2	Neutral
4	3	Disagree
4	4	Disagree

Table 7 – Algorithm Evaluation result.