A CROSS PLATFORM APPLICATION PROGRAMMING INTERFACE FOR LEARNING MANAGEMENT SYSTEMS

by Mabonga Hesbourne

School of Computing and Informatics

University of Nairobi

A CROSS PLATFORM APPLICATION PROGRAMMING INTERFACE FOR LEARNING MANAGEMENT SYSTEMS

MABONGA HESBOURNE

P58/9169/2006

SUPERVISOR

DR. ROBERT OBOKO

A project report submitted in partial fulfillment for the requirement of Master of Science Degree in Computer Science of the University of Nairobi.

2016

DECLARATION

This report is my original work. It has not been presented for a degree in any other university. No part of this research may be reproduced without the prior permission of the author or the University of Nairobi.

Student Name:

Hesbourne Mabonga Signature ____________________________ Date __________________

This report has been submitted for examination with my approval as the University Supervisor.

Dr. Robert Oboko Signature ____________________________ Date __________________

EXECUTIVE SUMMARY

Technological ubiquity has proliferated in our midst to the extend it is part of our daily lives; academics not exempted. The needs for connectivity, convergence of technological infrastructure and cross platform interoperability have necessitated the development of cross platform application interfaces to aggregate information and manage tasks in our environments. This project examined the design features of two learning management systems Claroline and ATutor, designed, developed, implemented, and evaluated the functionality of a cross platform application programming interface in those two learning environments. The successful evaluation of the functionality of the project is of benefit to other platforms such as e-commerce. The development approach was agile with strict adherence to extreme programming software development life cycle. The study environment was the two learning management systems configured at Kibabii University, in such a way that a team of Information Technology students utilized the Application Programming Interface with Claroline while the Computer Science team worked with the ATutor platform. Questionnaires were distributed to both groups of students to evaluate and ascertain the system's functionality. In conclusion the system worked well with the need to incorporate more features from the two LMSs, such as notifications from chats and forums on topical issues trending and are of interest to the user according to his/her defined profile. I recommend that the functionality be extrapolated into the mobile bound applications such as WhatsApp, since the environment today is moving towards the mobile environment.

Table of Contents.

List of figures

List of Tables

ABBREVIATIONS

LMS - Learning Management System

API – Application Programming Interface

URL – Uniform Resource Locator

PHP - Hypertext Preprocessor

xAPI - Experience API

XP – Extreme Programming

SDLC – Software Development Life Cycle

SCORM – Sharable Content Object Reference Model

AICC – Aviation Industry CBT Committee

DOD – Department of Defense

ADL – Advanced Distributed Learning

IEEE – Institute of Electrical and Electronics Engineers

ARIADNE – Alliance of Remote Instructional Authoring & Distribution Networks for Europe

AGRs – AICC Guidelines and Recommendations

CDS – Courseware Delivery Stations

CMI – Computer-Managed Instruction

CAM – Content Aggregation Model

SCOs – Sharable Content Objects

CHAPTER ONE
INTRODUCTION

This chapter introduces the background against which the research shall be undertaken, the statement of the problem, the purpose and objectives of the researcher and the significance of the research. It is therefore a summary of what the entire research entails.

1.0 Background

In the current continuum, technological interconnectivity is a reality that defines every aspect of our lives. Organizations, institutions, societies, government, sundry and all have gone connected. The need for functionality, portability and connectivity of these technological gadgets has defined new ways in which participants in the environment behave and act, at times enhancing certain aspects, modifying some or even defining totally new phenomena. These trends have seriously reengineered existing systems. Such technological trends call for concerted efforts in understanding and exploring new opportunities that arise such as cloud computing. Organizations have invested heavily to harness or capitalize on any arising opportunities; the society has positively transformed their mindset into embracing services brought forth by these trends, academic institutions have established or tailored courses towards the arising challenges while government has enacted policy guidelines and laws to govern matters arising.

The rate at which technology has percolated and been embraced by the society has simply turned every aspect of our lives upside down. In the commercial circles, there is a paradigm shift in the attention from the producer to the customer. The producer no longer dictates the market trends. The customer dictates what he/she wants, in which way he/she wants it and when he/she wants it. In the academic field things are not any different. There has been a paradigm shift on attention from the teacher to the student in the learning process. Technology has introduced learning management systems that have enhanced corroborative learning through discussion forums, wikis, file sharing, etc. The student no longer depends on the instructor to inform or convince them on any concept. They get information elsewhere in the systems with ease and convenience.

Among the many benefits brought about by technological learning management systems in the learning environment is the ability of the student to access academic content conveniently, repetitively and cost effectively. The LMSs have quite ably put the student at the center of the learning process. This phenomenon has not only enabled the student access enormous amounts of information, but as well gone further in trying to convince them about certain concepts without the intervention of the instructor.

Even though these enhanced collaborative learning features have enumerable benefits to the student, they have not done the student any justice either. They have vehemently bombarded the student with colossal amounts of information from every aspect of the academic environment that the student loses track midway on what they desire. A student joining a certain forum would be forced to participate in these forums actively or passively depending on the trending topic. It is with this view that results into the need to introduce an algorithm in a learning environment to sift through the vast amounts of data in the discussion forums, wikis and files, and inform the student on articles, topical issues and information pertaining to what they are seeking.

1.2 Statement of the problem

A Learning Management System (LMS) is an application that comprises of tools that manage the learning process (Educause, 2010). As defined by the K-12 BluePrint (2014), it is a platform that is meant to deliver content to students in and out of the classroom environment. Watson & Watson (2014) say that an LMS is a systemic infrastructure that manages the learning process of an entire organization which is currently being used to describe a number of different educational computer applications. In this context therefore, there are severally defined applications that are geared towards the management of the learning process in various organizations. These applications have distinct structural demarcations within which tools such as chats and forums have been created for the management of the learning process. Some are vendor oriented while others are open source platforms with similar features and functionalities.

In learner centered learning which is motivated by most of the LMSs in the learning process, is the ability to place the student at the center of the learning process. It is a pedagogy that has been variously proposed and supported as the most suitable learning method for higher education e-learning environments. The reason being that at this point, the student need to have an in-depth knowledge of concepts introduced earlier. Here, instructors do not pursue one teaching method but a variety of different types that shifts the role of the instructors from givers of information to facilitators (Blumberg P, 2013). However the distinct boundaries that creates the independence of these LMSs from each other tethers the student within their respective confines. That is, they enable the student to only access the tools and content within the confines of that application and not elsewhere.

The lack of structured interoperability amongst these educative computer applications poses a challenge. Any deficiencies or limitations in an approach espoused in one of the applications or lack of awareness of other approaches engaged in other applications is heavily felt in these educative computer applications. For instance, a student pursuing a neuro networks unit in a computer science course could be highly disadvantaged if such an application only tethers the student in the confines of computer science discussion forums, chats and wikis whereas there are better approaches, articles and discussions that could be of interest and beneficial in the medical and psychological course units.

1.3 The proposed solution

In view of the above challenges, there is therefore need to converge content of these variously available educative computer applications operating in different environments, in such a way that they notify a student whenever there is availability of discussions or articles of interest elsewhere according to the students quest for information, search patterns or availed bio data. That need can be fulfilled by having a cross platform API to provide a dashboard onto which students shall be notified on topical issues arising and could be of importance or interest to the student.

The API shall enable the student to search for forums, wikis and articles that are of importance across various LMS applications from a centralized dashboard, suggest previous discussions, articles and documents that relate to the student's area of interest, notify the student of an article that has just been submitted in the student's area of interest that the student previously searched, provide an interface to upload documents and classify them, and give materials suggested for learning based on the student's bio information.

1.4 Purpose of the project

The purpose of this project is to develop a cross platform Application Programming Interface that can aggregate information from variously available Learning Management Systems.

1.5 Objectives

1. To examine the design of the features of at least two LMSs i.e. Claroline and ATutor.

2. To implement a cross platform API that interconnects at least two LMSs in a learning environment

3. To evaluate the API in a learning scenario

4. To extrapolate the functionality of the API to mobile platforms.

1.6 Significance of the project

This project shall be of great benefits to the student fraternity who shall be in need to explore various sources of information in vain in spite of the time the information was sought and across various available platforms. The application shall sieve and avail to the student only materials that are relevant to their interests.

If the extrapolated functionality works well with various other platforms such as the ecommerce systems, then all others including the hospitality industry, the health sector, the entertainment industry, the sporting fraternity, etc., shall all be of keen interest in acclimatizing into this technological development.

With the world moving towards converging mobile data communications, this functionality would highly be of importance when trying to accrue the benefits that comes with the converging technological infrastructure.

CHAPTER TWO

LITERATURE REVIEW

In this chapter, the researcher explorers work done in the area of research under study. Here, we review and quote extensively work done by others in the area of study and bring out the gaps that inform the genesis of the problem under study.

2.0 Introduction

When the World Wide Web came into existence in the 1990s, the core functionalities that sustained its existence and drove its growth were web applications. Content management applications such as those for news publishing, discussion forums and web-based chats were introduced. These applications later became quite popular because they were not accustomed to any other client tools but the then popular web browsers such as Mosaic, Netscape or Internet Explorer (elacd.carnet.hr, 2006). Academicians found these applications quite useful for teaching purposes. These applications had varied functionalities such that they made user management problematic. For instance, users needed to create and sustain different user accounts for every applications they used and the need to redo repetitive tasks such as adding, updating and deleting student and course information from one application to the other. These challenges necessitated the development of dedicated LMSs to integrate the scattered functionalities into a wholesome intergrated learning process.

Learning Management Systems (LMS) are applications that comprises of tool that manages the learning process (Educause, 2010). These tools have different designs and implementation techniques but achieve the same thing; that of managing the learning process. The integrated functionalities that have wholesomely described the LMSs have been variously named (Watson & Watson, 2014), i.e. Claroline, ATutor, Moodle, WebCT, ILIAS, etc.

For these LMSs to conform to the prevailing technologies, they have been inclined to adhere to various set standards such as the Sharable Content Object Reference Model (SCORM), the Aviation Industry CBT Committee (AICC) and Section 508 Standard.

2.1 Standards for Learning Management Systems (LMSs)

In 1997, the American government wanted to standardize and modernize education management. Through the Department of Defense (DOD), it established the Advanced Distributed Learning (ADL) Initiative that created an international community to collaboratively develop a cost-effective distributed learning model that was to be consistent across national and organizational borders (CourseAvenue LLC, 2011). The initiative worked with the Institute of Electrical and Electronics Engineers (IEEE), the Aviation Industry CBT (Computer-based Training) Committee (AICC), the IMS Global Learning Consortium, Inc., and the Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE) to develop guidelines and specifications to make learning software accessible, interoperable, portable, durable, and reusable. These guidelines have since become the standards for learning management systems.

2.1.1 Aviation Industry CBT Committee (AICC)

Founded in 1988, the AICC is made up of a cluster of industry experts from various professions that provides and promotes information, guidelines, and standards that result in the cost-effective implementation of computer-based training (CBT) for the aviation industry and the worldwide training community(ADLnet.gov, 2014). It is arguably the first eLearning standard to be developed and is fairly well entrenched. However, not many organizations and institutional products support the standard. Though gone through several revisions to keep pace with the developing technological needs, the standard has not been consistently updated. Consequent revisions have resulted in the development of several functionalities to help contain rising challenges. The main artifacts of AICC are subsumed under the AICC Guidelines and Recommendations (AGRs). Relevant e-learning AGRs issued by the AICC include:

1. AGR-002 (Courseware Delivery Stations): Includes technical recommendations for the acquisition of CBT stations.
2. AGR-006 (Computer-Managed Instruction – CMI): Recommends guidelines for the interoperability of CMI systems, enabling them to use CBTs from different origins.
3. AGR-007 (Courseware Interchange): Includes guidelines for interchange of CBT courseware elements such as text, graphic, audio, etc.

4. AGR-010 (Web-Based Computer Managed Instruction): Adapts the AGR-006 interoperability guidelines particularly for Web-based CMI systems.(Adina Uta, 2007).

In tandem with the development of fragmented AGRs, AICC is redirecting its effort on the latest version of its Computer Managed Instruction (CMI 5) specification, a leveraged development from the Experience API, as the run-time communication protocol for the next generation of AICC CMI systems (ADLnet.gov, 2012).

2.1.2 Sharable Content Object Reference Model (SCORM)

This standard puts together sets of technical features that corroborates learning experiences to all online and on demand through attributes that include interoperability, portability, reusability and the instructional sequencing of self-paced e-learning content (ADLnet.gov, 2015). The content distributed through SCORM can be delivered to the learners via any SCROM – conformant LMS.

2.1.2.1 Interoperability

This attribute defines a common data model and application program interface (API) for e-learning content that allows standardized communications between client-side content and a system component (called "the run-time environment"), which is commonly provided by a Learning Management System (LMS) (ADLnet.gov, 2015).

2.1.2.2 Portability

This attribute defines how content can be integrated within systems that are SCORM – conformant. For example, the SCORM Content Aggregation Model (CAM) defines how to package content for exchange from system to system, in a transferable ZIP file called the Package Interchange Format (PIF). The packaging enables a standardized portability mechanism between various SCORM – conformant learning environment applications (ADLnet.gov, 2015).

2.1.2.3 Reusability

The SCORM Content Aggregation Model (CAM) describes the components used in a learning experience, and defines how to describe those components to enable search and discovery. The model therefore promotes reusability of learning content across Learning Management Systems (LMSs) and repositories, describes responsibilities and requirements for building content and content organizations such as course, lessons, modules, etc., and contains instructions for applying metadata to the all of the content organization components in the content package. On the server side, the CAM details the format an LMS must be able to "import" for the purpose of providing content to users (ADLnet.gov, 2015).

2.1.2.4 Sequencing

This attribute describes how SCORM-conformant content is delivered to learners through a set of learner or system-initiated navigation events. The branching and flow of that content may be described by a predefined set of activities. SCORM sequencing rules allow instructional designers and content developers to specify the order in which sharable content objects (SCOs), the smallest piece of content that tracks progress, are delivered to learners and what navigation controls are present in that SCORM – conformant LMS (ADLnet.gov, 2015).

2.1.3 Experience Application Programming Interface (xAPI)

The Experience API is an electronic learning software tool that learning systems and their consequent content to interoperate as it tracks all learning experiences. The experiences are written into a Learning Record Store (LRS) that is encompassed in the traditional Learning Management Systems (LMSs) or are independent (elearningchef.com, 2014).

Managed by the ADL, the xAPI is a young but promising specification that is hastily gathering steam. It is different from SCORM in that it releases the content from the confines of a specific application to a directly web accessed application with defined sets of tracking parameters.

2.1.4 Common Cartridge and Learning Tool Interoperability standards (CC/LTI).

Put in place by the ‘IMS Global Learning Consortium’ (IMS), the CC/LTI standard is purely applied in the academic arena. The Common Cartridge content packages have similar features to to SCORM-compliant packages but may slightly differ based on the inclusion of features common to LMS products such as discussion forums. In the actual sense, CC widens the scope beyond the actual lessons, at times covering the entire range of digital course materials such as a course, lesson plan, or assessment or a combination of all three (elearningchef.com, 2014). In general, LTI-compliant tools are plugged into learning management systems, allowing the application to track user actions such as eBooks uploaded, grading tools availed, etc., to the LMS directly.

2.1.5 Questions and Test Interoperability (QTI)

This is a standard that is a subset of CC but maintained by IMS. If an LMS conforms to the QTI environment, then one can import a QTI document and the LMS works out the rest of the details. This standard has the widest reach in academics, corporate circles, and public development. However, compared to others in the environment, it is in a nascent stage. The standard has an interesting approach with the credentials of being the only standard that has ventured out of the academics. Examples include Blackboard, Moodle, and Canvas (elearningchef.com, 2014).

2.1.6 CMI-5

This is an AICC organization initiative that is building on the success of the xAPI to leap into the next generations’ technological specifications management. The AICC’s CMI-5 professionals are constantly monitoring progress.

The underlying advantage of the CMI-5 is the ability to aggregate the best features and functionalities of the other standards such as SCORM, AICC and xAPI into a more robust standard they are individually. However, its success heavily depends on the type of data required for the successful implementation of the xAPI (elearningchef.com, 2014).

Research Gaps

In view of the above discussed standards, the all the specification standards are struggling to keep pace with the developing technological trends especially in the Mobile computing areas. Though trying to re-invent the specifications through fragmented AGRs such as CMI-5, and xAPI specifications, it is still quite clear that they have not fully addressed the concerns of the Mobile computing area. That is, there is no core standard that has been developed to specifically address the interoperability concerns between web based LMSs and Mobile based LMSs or Mobile based LMSs and other Mobile based LMSs.

Secondly, the above discussed standards address portability of the content rather than interoperability of the content. For example, SCORM discusses interoperability attribute as one that defines a common data model and an API that allows standardized communications between client-side content and a server side component. This happens within the SCORM – conformant application. Now, this interoperability attribute has limitedly tethered the functionality within one common standard. That that is SCORM – conformant. When it comes to integrating content across platform, the standard elaborately addresses these challenges quite ably. The xAPI specification has tried to address these concerns but has not fully exploited the challenges involved.

Thirdly, the learning process in endless, hence the need to expand those standards that are purely academic through cross platform interoperable Application Programming Interface by linking the applications confined in the standards with those that have been developed in other standards.

2.2 Web based Learning Management Systems

Learning Management Systems (LMS) are currently most popular e-learning tools. Just like in other technological fields, there are no standardized terminologies that refer to various LMS related software. They are web-based software, designed to support teaching and learning activities. They typically provide a combination of functions that can be carried out online, such as evaluation, communication, content submission, gathering students' works, management of student groups, questionnaires, monitoring tools, wiki, blog, chat, and discussion forums (M.

Yorulmaz, et al, 2012). Examples of web based LMSs include but are not limited to Claroline, Moodle, WebCT, atutor, etc. They provide interaction from plugged on the network larger computing devices such as desktop computers and laptops.

2.3 Mobile based Learning Management Systems

Also referred to as “anytime-anywhere learning”, Mobile learning is refers to two things - one, the mobile devices on which learning is conducted, and two, the term mobility, which is the ability to move around (P. Nimritta, 2015). Mobile learning has all the benefits accrued from e-Learning platforms which include collaboration, fostering and facilitating communication, and increased mobility. When dealing with Mobile learning, one needs to put in mind quite a number of factors such as the location of users, i.e. at home, in transit, or at school, how much time they have, and what work they are trying to accomplish i.e. are they trying to read course material, or take a quiz? Most learners as well switch between different devices when learning so the ability to save work across platforms is an essential component of mobile learning. Learning materials should be easily accessible, as well as easy to read, use, and engage, or else it isn’t likely that learners will use them. Examples of Mobile learning systems include among others, Moodle, Litmos, etc., which comprise features such as Mobile learning apps, Mobile access, compatible w*ith mobile based browsers, etc.*

Research Gaps

In view of the above LMS technologies, there is lack of interoperability attributes between the web based and mobile based LMSs. Though the underlying technological infrastructure between the two LMSs are slightly different, they can still be aggregated so that the goings on in Mobile LMS can be reflected in the web based LMS and vice versa.

2.4 Application Programming Interface (API)

An application program interface (API) is a set of routines, protocols, and tools for building software applications (V. Beal, 2016). An API usually specifies how software components should interact as they work together towards a common end. A good API makes it easier to develop a program by providing all the building blocks. K. Stanton (2015) further breaks it down as follows: “API is a precise specification written by providers of a service that programmers must follow when using that service,” he says. “It describes what functionality is available, how it must be used and what formats it will accept as input or return as output.”

2.4.1 Importance of APIs

1. Businesses Create Apps with APIs to pull data from open APIs and help streamline business processes in some new way. (M. Patterson, 2015)

2. Business People Use Apps with APIs to allow programmers build amazing tools that help us do our jobs more effectively. A good example is this keyword tool that accesses Google’s search API to suggest keywords your business should target. (M. Patterson, 2015)

3. Businesses Rely on Open APIs because third-party developers build out applications that further the use of the company’s core product, in turn saving both time and money. (M. Patterson, 2015)

In e-Learning, we can summarize that APIs are important in pulling together data from various sources and aggregating them together for the benefits of the students learning process.

Good examples of e-Learning APIs are the becoming famously known Experience API also known in development terms as Tin Can API and Litmos APIs. The Experience API is a powerful aggregator between an LMS and other systems in an organization (Rustici Software LLC, 2016). It uses a Learning Record Store as functionality to aggregate data within the various systems that the API gathers information from. The Litmos API enables developers to connect their applications to the Litmos training engine, allowing or greater integration of training data generated in Litmos with any other systems that are currently used within the organization (J. Barnes, 2012).

Research Gaps

The two APIs can interface between web-based LMSs and Mobile based LMSs quite ably but are still under development and testing hence the need to reinforce the growth of such APIs from various perspectives so as to expose any failures that could arise from any of the approaches. S.

Boller (2013) observes that the Experience API is still at the hypothetical stage and still cannot be relied upon in interpreting all of the new data points it can now collect. He avers that LMS developers shall need to build robust new analytical, reporting and data visualization capabilities if they are to take advantage of the abilities of the Experience APIs to collect data from informal learning activities and detailed results from games and mobile app usage. This observation holds water because the data collected can only be as good as the means for processing and interpreting that data.

CHAPTER THREE

THE SPECIFICATION AND DESIGN

This chapter is informed by the characteristics espoused in the proposed solution. Those characteristics describe the features that the proposed system should comprise. For that reason, this chapter goes deep in to analyzing those characteristics and designing a solution that fits those characteristics.

3.0 The System Specifications

The system provides a dashboard unto which users' uses to search for academic materials of their interest. If the material(s) are found in any of the two LMSs, Claroline and/or ATutor, the system displays the materials on the dashboard, together with the platform unto which the material were found in, i.e. Claroline or ATutor. If no material(s) is found to be relevant to the searched item, the activity ends there.

To track and notify users based on user activities such as search trends and the respective user bio data, the user is supposed to login. If the user does not have the login credentials, then they are required to sign up from a provision on the API dashboard. During the sign up procedure, the user is required to provide information about his/her interest, profession or specialty. It is this information that the system uses to notify the user about the trending activities from the two LMSs.

When a user logs in successfully, the user has liberty to upload their own materials from the dashboard without necessarily being required to belong to either of the two LMSs. However, the user can still belong to either or both LMSs and use the login credentials of any of the two LMSs to login into the API from the dashboard. If the user searches an article with login credentials from either of the two LMSs and the article is not found, the searched article name is saved in the respective LMSs database. However, the when an article shall be uploaded at a later time that has some corresponding name features similar to the saved search name, the API captures that and displays it as a notification. If several are availed, whether from the same LMS or other, the number of those found is indicated on the dashboard notification area.

Equally so, if there is a highly statistical trending topical issue that is of interest to the user in accordance to the logged in user's profession or specialty from either of the three platforms, then the algorithm captures that, and notifies the user on the dashboard that there is a trending topic that is of interest as per your profile definitions, and suggest that the user joins the forum. If several are available from either of the platforms, the number of those trending are indicated in the notification area.

3.1 System Design

The system was broken down into user stories as per the requirements of the agile software development approach's Extreme Programming software development life cycle, with each user story designed as an independent module, then integrated.

The dashboard is designed in such a way that there is a field for the user to enter item names to search for in the system. If the article sought is found, the article is displayed and the statistical access resource center is updated.

For a user to login so that they get personalized notifications to search patterns, then they are required to sign up. During the process, the user creates an account that includes his/her areas of interest, profession or specialty. He/she then is required to login. After successful login, the user gets notifications on high statistical trending topical issues, items or articles as per the user's profile definitions. Additionally, if the user previously requested a resource and the resource was not available but has become available, the successfully logged in user is notified on the same. Finally, if there is a resource that matches the interests, profession or specialty of the logged in user, then the user in notified accordingly.

The system then updates the resource access center every time there is a notification that has been acknowledged for statistical functions. The system was designed such that a user could be utilized whether the user is logged in or not. If a user is logged in, the system could customize search pattern to the user's preferences and notify him/her accordingly. The following diagram summarizes the system design.

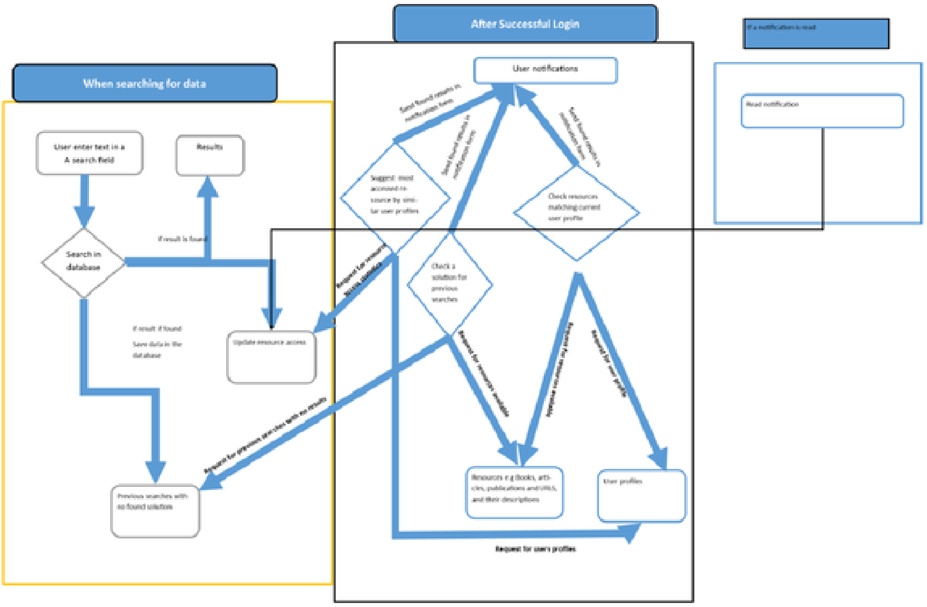

When searching for data
User enter text in a A search field
Results
Search in database
If result is found
If result if found
Save data in the dat abase
Update resource access
Request for previous searches with no results
Previous searches with no found solution
After Successful Login
User notifications
Send found results in notification form
Send found results in notification form
Send found results in notification form
Suggest most accessed re-source by simi-lar user profiles
Check resources matching current user profile
Check a solution for previous searches
Request for resource access statistics
Request for resources available
Request for resources available
Request for user profile
Resources e.g Books, arti-cles, publications and URLS, and their descriptions
User profiles
Request for users profiles
If a notification is read
Read notification

Fig 1 : API design architecture (Source: Author)

CHAPTER FOUR

IMPLEMENTATION

This chapter describes the methodology that the features designed in chapter three are to be put to use. The chapter describes the approach and the method used in actualizing the stated system design.

4.0 Methodology

Extreme Programming (XP) is a disciplined methodology that focuses primarily on customer satisfaction. The system was therefore implemented in phases as described in the XP Life cycle. The XP life cycle can diagrammatically be summarized as follows:

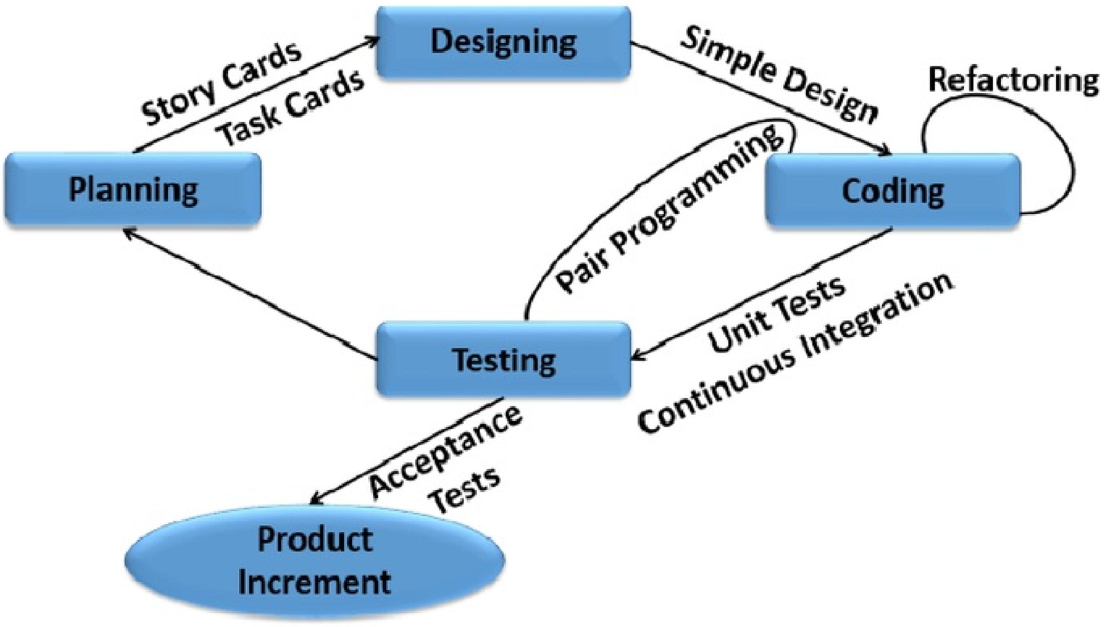


Fig 2: XP SDLC (Source: Tutorials Point)

4.1 Planning Phase

While adhering to the XP values of communication, simplicity, feedback, courage and respect, the Claroline platform was installed, configured and introduced to the Information technology (I.T) team while the ATutor platform was installed, configured and introduced to the Computer Science team. Both teams were differently taken through the platforms features and informed to utilize the system as the course outlines and course materials for two courses, i.e. Principles of Operating Systems and Data Communications I for the I.T. teams and Operating Systems Design and Data Communications for Computer Science teams, were to be uploaded to the platforms for their references. In a week's time they were to be given assignments on various topics. In three weeks' time both teams were to be subjected to a continuous assessment test basing on the various course materials that were to be available. However, both teams were advised to read widely and upload relevant academic materials within their respective accounts.

True to the developer's wishes, the teams were given the assignments and continuous assessment tests based on a variety of academic materials uploaded from either of the system. The two teams were given questionnaires as illustrated in Appendix 1.

After the students had sat for the assignments and continuous assessment tests, they were both introduced to a dashboard that would search for materials from both platforms and given second assignments on various topics. They were to submit that in a week's time. As they submitted their work, they were given another questionnaire as illustrated in Appendix 2.

Responses from some of the team members were that the API enable users be given the capability to log into the system and be able to create their own articles and/or summaries of work they have gone through as they research. There were also requests to have forums and chat trends included in the notifications on the dashboard. This was found to be quite challenging given the colossal number of lines that were to be sifted through, captured and included in the API. It is worth noting that the LMS platforms took teams great lengths of time to develop and test before release, of which Claroline was version 1.11.10 while ATutor was version 2.2.2, and therefore the forum and chat functionalities could not be achieved within the given stipulated period of time.

4.2 Design Phase

The design of the system was broken into modular incremental user story preferences. The first user story was the search algorithm prototype that was meant to enable the teams understand what the aim of the system was. Here, the search was designed such that the algorithm could fetch information from the two LMS platforms and avail the details on the dashboard. Once a result had been found, the article would be displayed and the statistical resource access would be updated. The following diagram illustrates the design of the prototype user story.

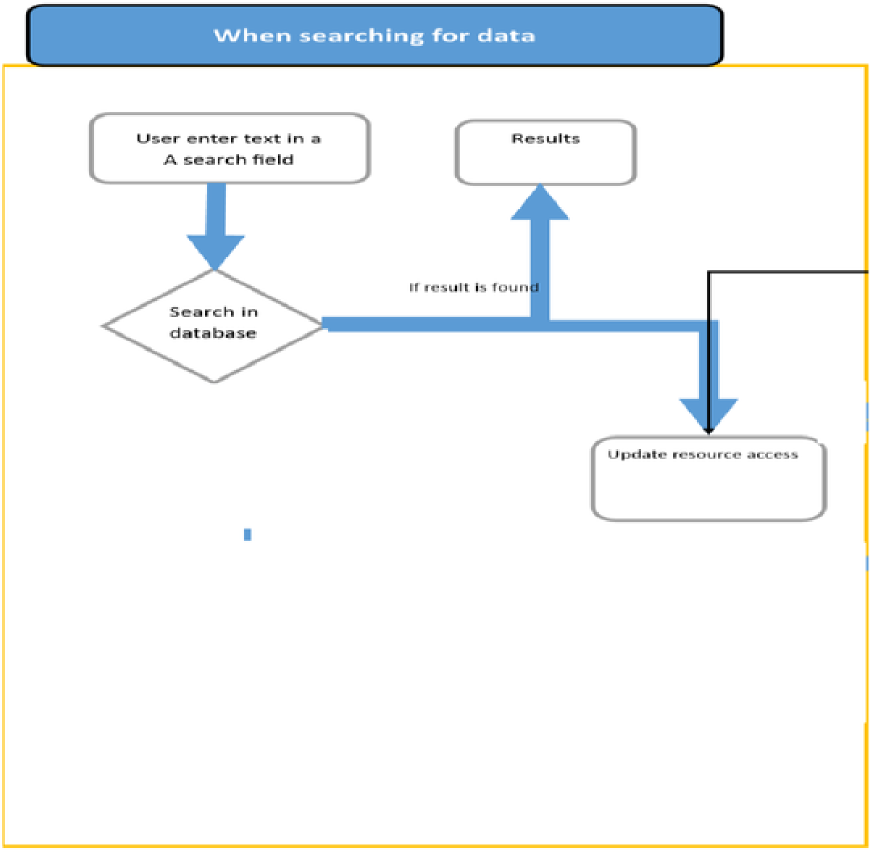


Fig 3: Part of the API design Architecture (Source: Author)

Responses from the second questionnaire had an overwhelming requirement that user's be given an opportunity to have their own summarized articles that they could save and refer to from time to time. This necessitated the need to provide a feature to create and save articles. The API also provided an opportunity to save URL links to necessary reference books the user may have come across and/or had gone through. To capture search patterns and customize user info as per every user's preferences, the user was henceforth required to sign up. In the sign up requirements, there was need to capture the user preferences. These preferences were include in the customized user notification requirements. The following is a diagrammatic summarized design of the user story that required user preference customization.

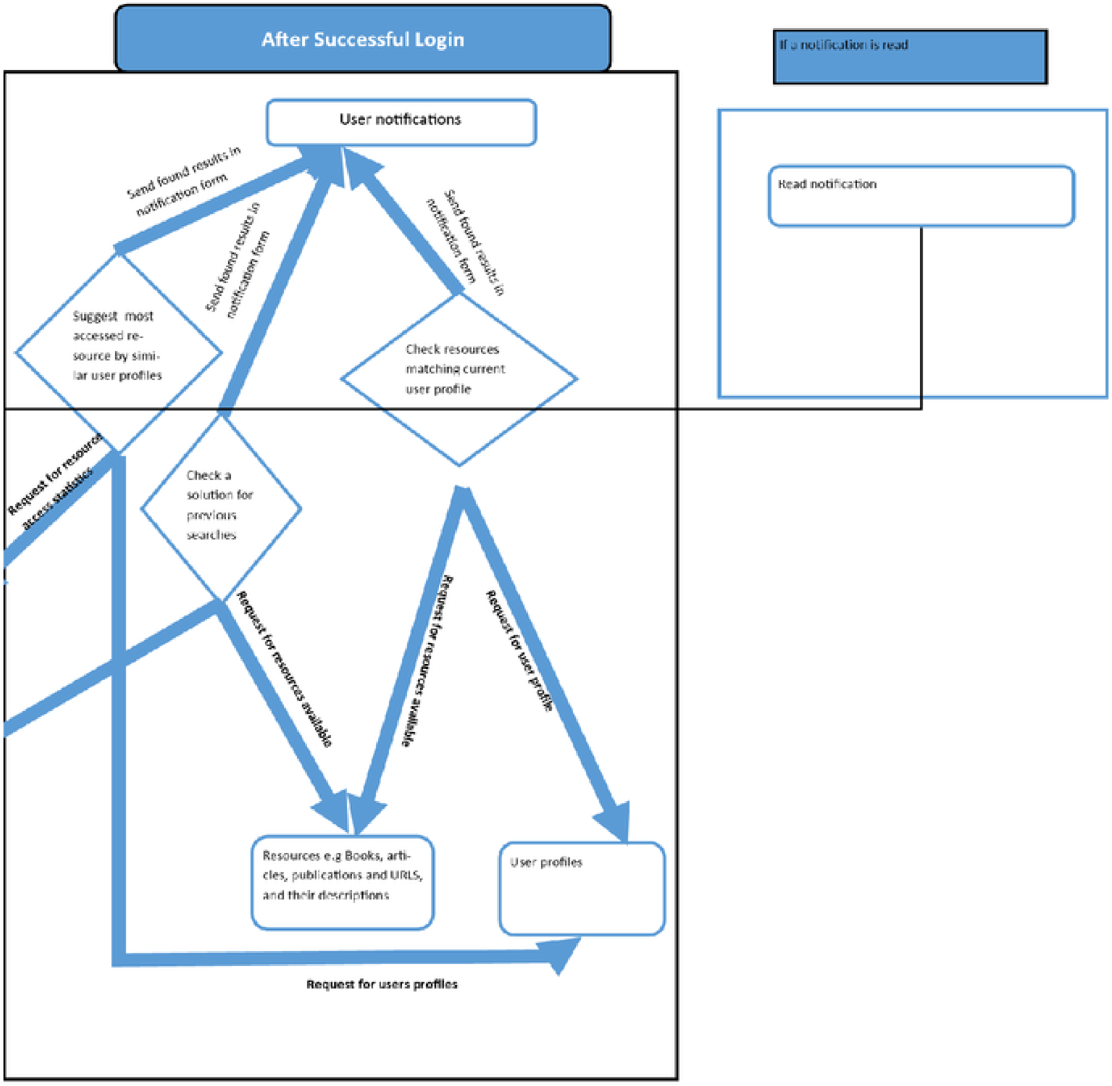

After Successful Login
If a notification is read
User notifications
Read notification
Send found results in notification form
Send found results in notification form
Send found results in notification form
Suggest most accessed re-source by simi-lar user profiles
Check resources matching current user profile
Request for resource access statistics
Check a solution for previous searches
Request for resources available
Request for resources available
Request for user profile
Resources e.g Books, arti-cles, publications and URLS, and their descriptions
User profiles
Request for users profiles

Fig 4: Part of the API design Architecture (Source: Author)

The API database that captures the user's profile and other details is designed as follows:

Conceptual design

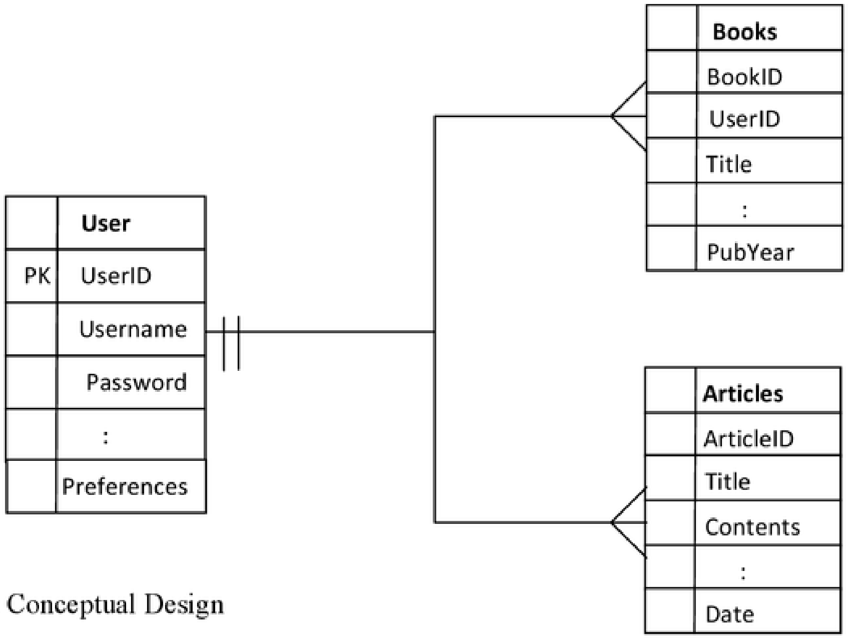


Fig 5: Conceptual Design

Logical Design

The logical designs for the database is as follows.

Table for users

```
CREATE TABLE User (Salutation VARCHAR (20),
                   Surname VARCHAR (20),
                   OtherNames VARCHAR (50),
                   EmailAddress VARCHAR (20),
                   Contacts VARCHAR (20),
                   Interests VARCHAR (20),
                   Username VARCHAR (10)
```

```
                    Passwd VARCHAR (15),

                    DateAdded DATE);
```

The logical design for the Articles table

```
CREATE TABLE Articles     (Title VARCHAR (20),

                          Description VARCHAR (20),

                          MainContent VARCHAR (250),

                          DateAdded DATE);
```

The logical design for the Books table is as follows:

```
CREATE TABLE Books        (Title VARCHAR (20),

                          Author VARCHAR (20),

                          Publisher VARCHAR (20),

                          YearofPub VARCHAR (20),

                          Vol VARCHAR (20),

                          DownloadUrl VARCHAR (150),

                          Description VARCHAR (250),

                          DateAdded DATE);
```

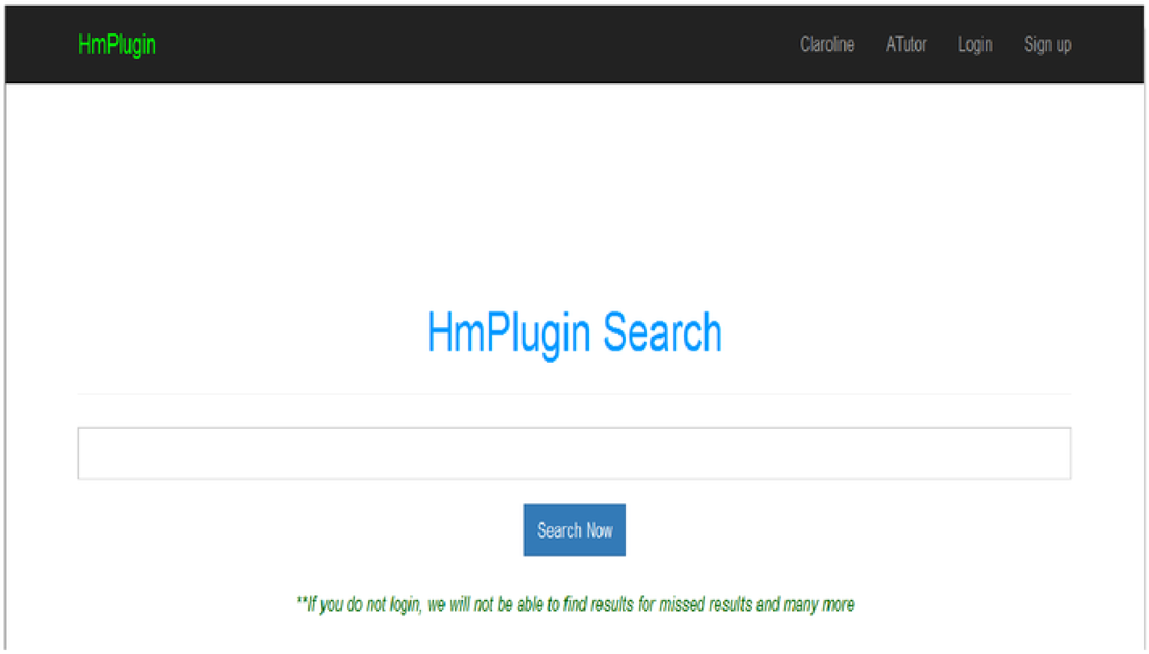
4.3 Coding Phase

Each user module was coded, validated independently and implemented. The pieces of code that provided the functionality of the user stories is as illustrated in Appendix 3.

The interface design was made so simplistic so as to simplify navigation between the LMS platforms in use. Therefore, the buttons that would lead to Claroline and ATutor were introduced and conspicuously displayed together and close to the login and sign up buttons.

There is a warning on the dashboard stating that if the user does not sign up then login, their search results shall not be monitored. This is meant to inform the user to login so that they can enjoy more functionalities, otherwise, the system is equally usable even without logging in.

The following is the description of the user interface the user could use to interact with users.



**If you do not login, we will not be able to find results for missed results and many more

Fig 6: API dashboard

The following pieces of code are important to mention for they provide vital functionalities. This statement <link href="css/bootstrap.min.css" rel="stylesheet"> enables the interface fit on a mobile device. It enables the interface condense controls into one that fits onto a mobile device display.

The statement <?php require_once ("_includes/menu.inc");?> requires that functionality from menu.inc file be imported into the interface. The menu.inc file is the one that contains functionality to notify the user. But this only happens if the user has logged into the system.

For users to log in, they must have signed up on the system. The sign up button is quite visible on the dashboard, at the top right hand corner of the interface. Once clicked, another interface as shown below is availed to take the particulars of the user. Of great interest is the field that captures the user's interests. This is the field that the API uses to notify the user whenever there are materials that are of interest to the user or trending topical issues. They are then saved in the database for future reference purposes.

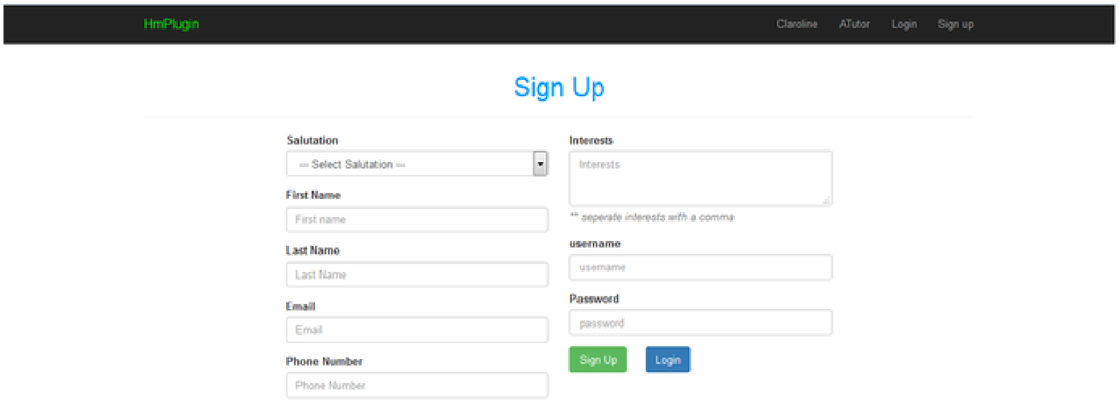
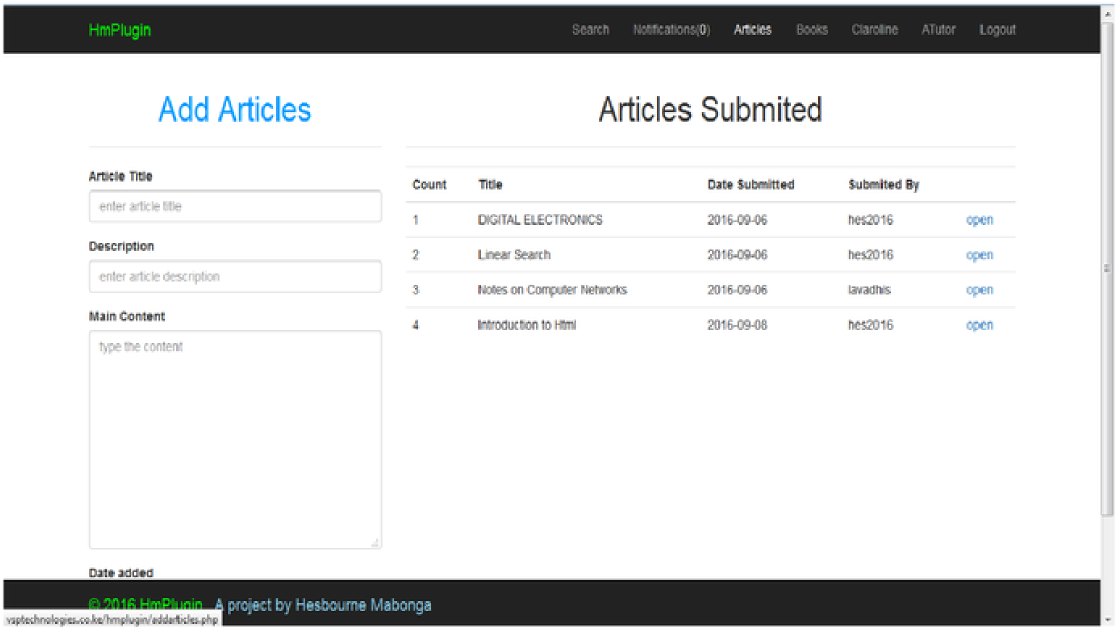


Fig 7 API sign up interface

This piece of code <form method="POST" action="processing/signupprocessing.php" novalidate> calls for a functionality from the signupprocessing.php file, a file that is responsible for saving the data in the database.

When the user has successfully logged in, they are able to create their own articles and save them for future reference. Other users as well who log into the system can search and have access to the articles. The interface that provides for this functionality is as shown in the diagram below.



Add Articles

Article Title

enter article title

Description

enter article description

Main Content

type the content

Date added

Articles Submited

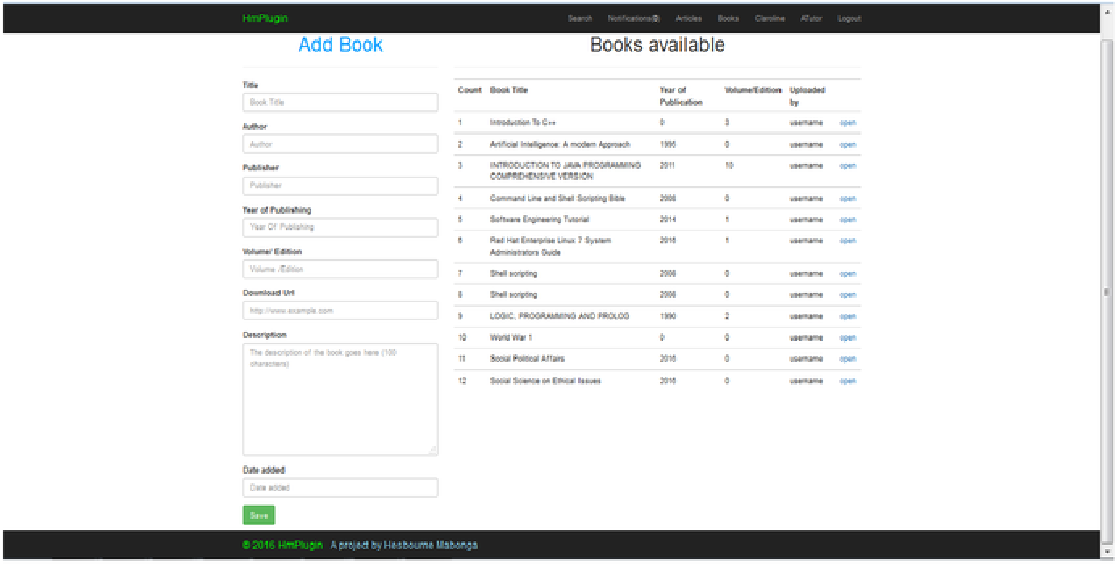
Count	Title	Date Submitted	Submited By	
1	DIGITAL ELECTRONICS	2016-09-06	hes2016	open
2	Linear Search	2016-09-06	hes2016	open
3	Notes on Computer Networks	2016-09-06	lavadhis	open
4	Introduction to Html	2016-09-08	hes2016	open

© 2016 HmPlugin A project by Hesbourne Mabonga

vaptechnologies.co.ke/hmplugin/addarticles.php

Fig 8 API Saving articles interface

The users who are logged in as well can upload reference materials for the API. These reference materials need not be uploaded wholly but the urls to their respective locations are saved. The following interface provides the feature for the reference materials update.



Books available

Count	Book Title	Year of Publication	Volume/Edition	Uploaded by	
1	Introduction To C++	0	3	username	open
2	Artificial Intelligence: A modern Approach	1995	0	username	open
3	INTRODUCTION TO JAVA PROGRAMMING COMPREHENSIVE VERSION	2011	10	username	open
4	Command Line and Shell Scripting Bible	2008	0	username	open
5	Software Engineering Tutorial	2014	1	username	open
6	Red Hat Enterprise Linux 7 System Administrators Guide	2016	1	username	open
7	Shell scripting	2008	0	username	open
8	Shell scripting	2008	0	username	open
9	LOGIC, PROGRAMMING AND PROLOG	1990	2	username	open
10	World War 1	0	0	username	open
11	Social Political Affairs	2016	0	username	open
12	Social Science on Ethical Issues	2016	0	username	open

Fig 9 API reference book saving interface

4.4 Testing Phase

The teams tested every user stories that were implemented, giving feedback to the developer continuously in tandem with XP principles of rapid feedback, assumption of simplicity, incremental change, embrace of change and quality work. Blackbox type of testing was vastly used by the user teams because of their limited knowledge of the architectural design and scripting language used. The teams selected were second year students from the I.T and Computer Science courses, with a target of using course units that had similar concepts to enable the teams share items on the system.

4.4.1 Test Cases

The following are the test cases that the teams tried on the first user story that was presented as a pilot for the teams to understand the system.

Test Case	Outcome	Verdict
Search for lecture Computer	The search din't find any	Not sure if the system works

Graphics notes.	materials related to the Computer Graphics unit.	as expected
Search for Data Communications lecture notes.	The search captured books for either of the system, i.e. one book on Claroline while the other on Atutor uploaded by the respective course lecturers of the Data Communications units.	Worked as expected.
Search for items on Analogue and Digital Signals.	Articles and url links to books were found on the system related to the item title searched. The articles were from both Digital Logic and Data Communications materials.	By virtue that it captured more than the areas of concern, i.e. Data Communications, the system had performed beyond expectations at the moment.

Table 1 First Test results

The following test cases were done after the second user story was implemented and the teams were aware of the expectations of the system.

Test Case	**Outcome**	**Verdict**
Team members tried to log into the system, and searched for Web Programming Tutorials.	There was no item found that was basically addressed for Web Programming. But users were informed that their quest had been saved for notification when available.	The ability to save searched items for future notifications was once more a feature that was beyond team expectations.

Team Members tried to search for the File Transfer Protocol (FTP) tutorial.	Materials were available that talked about FTP from Data Communications and Computer Networks course materials.	Worked as expected.
Team members logged into the system created articles based on what they were reading and were able to save them for future reference.	The articles were successfully saved and availed whenever there was a search or team members logged into the system.	Worked as expected.
Team members logged into the system and saved urls that linked to book materials of their interest.	The book materials were availed whenever there was a search on titles that related to the books and were as well availed whenever the teams logged back into the system.	Worked as expected.
Some team members uploaded materials and articles relating to some of the articles that were search and not found.	Notifications were immediately raised and the respective team members notified on the date they made the searches and from which platforms those articles were.	Worked beyond expectations are the teams had not intended to receive such notifications

Table 2 Second table results

CHAPTER FIVE

THE RESULTS AND EVALUATION

This chapter evaluates the suitability of the developed system in the environment in which the system is going to function. Such questions suffice. Has the goals of its development met? To what extend? These are some of the questions this chapter seeks to address.

5.0 Evaluation

The essence of this section is to determine the suitability for use of the system in a particular environment. To achieve the objectives, the system's results were captured and evaluated based on four basic activities, namely, Plan, Establish, Collect and Analyze (PECA) process, an ISO 14598 based standard (J Dean et al, 2004).

5.1 Planning the evaluation

During planning for the evaluation, the teams were formulated with careful credence to expertise, specialty and experiences. Team members with interest and expertise in systems development in various fields were called upon to test the API functionality and point out weaknesses. For instance, was the API successfully searching and aggregating the necessary information from the targeted platforms? Was the API notifying the user correctly on the number of items found based on the user's search pattern and history?, etc. Team members that had specialized in Web development capabilities were required to evaluate the various features of the functionalities such as the capability to notify the user, the color selection and the interface format with which the items found were to be presented on the interface, etc. On experiences, lecturers were included in the teams to bring in versatility as they had on one occasion or the other interacted with various LMSs and therefore could easily bring in a third perspective that the team members may have overlooked.

5.2 Establishing the criteria

To establish the criteria, there had to be a measurable statement that defines the capability necessary to satisfy a goal, and secondly, a means for assessing and assigning a value to the

API's level of compliance with the capability. In this project, four goals were identified, a number of capability inquiries made and a performance metrics sought to justify those capabilities. The following table shows the criterion used in this case.

Goal

- The API should aggregate information from more than one LMS
- The API should track a user's search patterns and notify him/her accordingly whenever searched items become available on any of the platforms
- The API should enable the user's to create their own articles and book references that can be availed to other users with interest in them.
- The API should track activities and trends that relates to the user's bio data from forums and chat activities and notify him/her accordingly.

Capability statement

- The API shall be able to search for articles, books and associated learning materials from
    - Claroline LMS
    - ATutor LMS
- The API should save search patterns so that items sought can be availed whenever they become available
- The API should save other articles and reference materials for the user so that they can be availed to others interested in the articles and materials
- The API should capture the user's bio data by letting him/her define a profile to use in tracking his/her preferences

Measurement method

- A dashboard shall be provided unto which the aggregated information for the LMSs shall be placed.
- The API shall provide feature that should enable users save articles and reference materials that can be availed to the users themselves as well as others.
- The API shall provide features that shall enable users sign up, hence providing the necessary bio data required to use in tracking him/her.

Table 3 Criterion used to evaluate the API

5.3 Collecting the data

Data collection is the third phase of the PECA process and it forms the foundation for analysis. In this phase, data was collected in simple repeatable measures with intent to measure and capture information in a form suitable for analysis. However it is worth to note that there were some similarities with software testing. In software testing, the goal was to discover whether the software behaves as expected while data collection was how the API behaves against the selected criteria.

Product Probe lead hands-on data collection technique was used to investigate features of the API such as its interoperability capabilities with other LMSs. The features examined under this technique were quite close in similarity to the features under testing the phase. The questionnaire that was used to collect data is as illustrated in Appendix 2.

5.4 Analyzing Data

During the fourth phase of the PECA process, the facts, checklists and other relevant types of data are consolidated and converted from raw data into meaningful information. In this project, the weighted aggregation technique was used to analyze data collected. A weighted scoring method was completed by defining the criteria weights. The weighted scoring method criteria weights were assigned using a scoring method, i.e. assigning a value between one and five to each criterion. The evaluation teams repetitively but independently discussed the scores until they reached a consensus on each criteria, with the professional evaluators ensuring that all the evaluators opinions carried equal weights. The results are as presented in the Table below which includes the criteria, their weights, weight in percentages and the associated goals.

		Evaluation Teams			
		A		**B**	
Criteria/Tests	**Weighted Score**	**Raw**	**Weighted Value**	**Raw**	**Weighted**
The API should aggregate information from more than one LMS	**20%**		**18%**		**14%**
Does the API have a provision to search for materials from the system?	5%	1.0	5%	1.0	5%
Does the API aggregate information from any LMS?	15%	1.0	15%	1.0	15%
Does it aggregate information from both LMS?	60%	1.0	60%	1.0	60%
Does it indicate from which platform that information was retrieved from?	20%	0.5	10%	-0.5	-10
Sub total			90%		70%
The API should track a user's search patterns and notify him/her accordingly whenever searched items become available on any of the platforms	**20%**		**20%**		**18%**
Does the API provide a feature for users to sign up?	10%	1.0	10%	1.0	10%
Are users allowed to login into the system?	10%	1.0	10%	1.0	10%
Does API save search patterns for the users on items not found in the LMSs?	40%	1.0	40%	1.0	40%
Are notifications raised whenever there is availability of items searched before?	20%	1.0	20%	1.0	20%
Does it inform the user how many items have become available and from which LMS platform?	20%	1.0	20%	0.5	10%
Sub Total			100%		90%
The API should enable the user's to create their own articles and book references that can be availed to	**30%**		**15%**		**6%**

other users with interest in them.					
Does the API provide a facility to create and save articles and/or notes for materials of interest?	10%	1.0	10%	1.0	10%
Does the feature advise the user to indicate the date or year of creation?	5%	1.0	5%	1.0	5%
Does it provide a facility to enable users save or create links to articles and/or books for reference purposes?	10%	1.0	10%	1.0	10%
During searching by any user, are these articles captured and availed for other users?	15%	1.0	15%	1.0	15%
Does the search indicate the plaform from which they have been captured from?	20%	-0.5	-10%	-1.0	-20%
Does the search incorporate the found articles and other book references with those from other platforms?	40%	0.5	20%	0.5	20%
Sub Total			50%		20%
The API should track activities and trends that relates to the user's bio data from forums and chat activities and notify him/her accordingly.	**30%**		**6%**		**6%**
During signing up on the API by the users, does the API capture any information that can enable the user be informed about trending topical issues?	10%	1.0	10%	1.0	10%
Does the API raise notifications on trending topical issues in forums of any of the LMSs?	20%	-1.0	-20%	-1.0	-20%
Does it capture any trends and raise a notication on activities that trends in available chats on any of the LMSs?	20%	-1.0	-20%	-1.0	-20%
Does it capture and raise notifications on materials availabe based on the users bio data?	50%	1.0	50%	1.0	50%
Sub Total			20%		20%

Table 4 A table of results from the evaluation of the API

The score legend is as follows:

The Score Value	**Definition**
1.0	The feature fully satisfies the decision criterion.
0.5	The feature partially satisfies the decision criterion.
0.0	The feature neither satisfies nor dissatisfies the decision criterion.
-0.5	The feature partially dissatisfies the decision criterion
-1.0	The feature fully dissatisfies the decision criterion

Table 5 The scores legend

The evaluation team A was majorly composed of students from the Computer Science class with their respective lecturer as a professional member while team B were students from the Information Technology class with their respective lecturer as the professional member.

The score for this evaluation differs slightly in that the final score for the goals is calculated by multiplying the total weighted score for each of the goals by the total weighted value for the goal.

Total Score (Goal) = ∑ (Weighted Score (Goal) * Weighted Value (Goal)

For example, in the Table shown, the goal Weighted Score is 20%, and the Weighted Value for Team A is 18% i.e. (90% of 20%).

CHAPTER SIX

RECOMMENDATIONS AND CONCLUSION

This chapter discusses the prospects for the future based on what the researcher has experienced as well as the conclusion of the research work.

6.0 Future Work

The current technological trends require that technological systems converge. In the convergence of these systems, there is a need to implement algorithms that communicates across various platforms, especially those platforms that have similar objectives, so that users are not tethered into environments that have limitations, whereas there are better alternatives. Technology has reengineered the society such that the producer no longer dictates. It is the consumer who dictates what they want, when they want and how they want goods and services. For these reasons, there is demand for cross platform algorithms to satisfy the consumer's changing preferences and accrue benefits that comes with technology.

Equally so, the paradigm shift by the societies from cabled communication media to wireless media has seen an upsurge in mobile computing. The proliferation of mobile devices for convenience and class has seen a niche develop in the society that needs apps to satisfy various needs in the cyber world. Quite a number of these apps have commercial objectives. It can be vital if cross platform algorithms could be employed as well to avail alternative perspectives and sources to users.

6.1 Conclusion

The average of the total scores for each evaluation team when summed up is above 50%. For example Team A (Computer Science team) is 59% while Team B (Information Technology team) is 44%. The average is 51.5%, i.e. (59%+44%)/2. This means that the project has achieved its objectives to a larger extend. Given enough resources in terms of time and effort, the algorithm is a worthwhile pursuit in the current environment. From the literature reviews, efforts are underway to develop more and more algorithms to suite variously available converging environments. This is evidenced by the reviews happening in the existing development standards. Further, the shift in the infrastructural architectures from cabled to wireless infrastructures emphasizes the need for these kinds of algorithms.

REFERENCES

1. Abts, Christopher M. & Boehm, Barry. COCOTS (Constructive COTS) Software Integration Cost Model: An Overview (USC-98-520). Los Angeles, CA: USC Center for Software Engineering, University of Southern California.

   http://sunset.usc.edu/publications/TECHRPTS/1998/1998_main.html.

2. International Organization for Standardization (ISO). ISO/IEC 9126:1991 – Information Technology – Software Product Evaluation – Quality Characteristics and Guidelines for Their Use. Geneva, Switzerland: ISO/IEC, 1991.

3. International Organization for Standardization (ISO). ISO/IEC 14598 1:1999 – Information Technology – Software Product Evaluation. Geneva, Switzerland: ISO/IEC, 1999.

4. The PHP GROUP 2001 – 2016 https://secure.php.net/releases/

5. Oracle Corporation et al, 2016 https://www.mysql.com/products/

6. ADLnet.gov, 2015 https://www.adlnet.gov/adl-research/performance-tracking-analysis/experience-api/

7. Rustici Software LLC, 2016 http://experienceapi.com/

APPENDIX 1

QUESTIONNAIRE ONE

You have been included in a select team of evaluators that has been using the Claroline/ ATutor Learning Management System (LMS) for a while now. You have even submitted an Assignment and a Continuous Assessment Test (CAT) based on various materials uploaded on either of the platforms. Kindly respond to the following questions.

Which type of LMS have you been using?

How did you find the functionality of platform?

○ Easy to use ○ Average ○ Difficult to use ○ Very Difficult to use

Did you upload any academic materials on to the platform?

If yes, approximately how many?

○ Less than 5 ○ Less than 10 ○ More than 10

Did you submit your assignment?

In your opinion, how do you rate the level of difficulty of the Assignment?

○ Easy ○ Average ○ Difficult ○ Very Difficult

Why?

Did you find the materials in the LMS platform relevant to the assignment you were given?

If yes, how relevant?

○ At least less than half of the questions were based on materials from the platform
○ Half of the questions were based on materials from the platform
○ At least more than half of the questions were based on materials from the platform

Did you submit your CAT? ______________________________

In your opinion, how do you rate the level of difficulty of the CAT?

○ Easy ○ Average ○ Difficult ○ Very Difficult

Why?

____

____

____

Did you find the materials in the LMS platform relevant to the CAT you were given?

____

If yes, how relevant?

○ At least less than half of the questions were based on materials from the platform
○ Half of the questions were based on materials from the platform
○ At least more than half of the questions were based on materials from the platform

APPENDIX 2

QUESTIONNAIRE TWO

You are a select team member of evaluators that has been using a dashboard that has included features of Claroline or ATutor Learning Management System (LMS). You have submitted your second Assignment based on various materials uploaded on either of the platforms. Kindly respond to the following questions.

How frequently did you use the dashboard?

○ Never ○ Rarely ○ Sometimes ○ Often ○ Always

How did you find the functionality of the dashboard?

○ Easy to use ○ Average ○ Difficult to use ○ Very Difficult to use

Why?

____

____

____

Did you submit your assignment? ________________________________

If yes, did you find the materials availed on the dashboard relevant to the assignment you were given?

____

If yes, how relevant?

○ At least less than half of the questions were based on materials from the platform
○ Half of the questions were based on materials from the platform
○ At least more than half of the questions were based on materials from the platform

A CROSS PLATFORM APPLICATION PROGRAMMING INTERFACE FOR LEARNING MANAGEMENT SYSTEMS

ORIGINALITY REPORT

%15	%13	%5	%6
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.adlnet.gov Internet Source	%4
2	sproutsocial.com Internet Source	%1
3	elearn.pri.univie.ac.at Internet Source	%1
4	elearningchef.com Internet Source	%1
5	repair-car-manuals.com Internet Source	%1
6	www.sei.cmu.edu Internet Source	%1
7	www.coursehero.com Internet Source	%1
8	wjeis.org Internet Source	%1
9	www.webopedia.com	

	Internet Source	<%1
10	www.bottomlineperformance.com Internet Source	<%1
11	Submitted to Kenyatta University Student Paper	<%1
12	Submitted to Colorado Technical University Online Student Paper	<%1
13	Submitted to Leeds Metropolitan University Student Paper	<%1
14	www.soberit.hut.fi Internet Source	<%1
15	help.litmos.com Internet Source	<%1
16	Submitted to The Robert Gordon University Student Paper	<%1
17	web.ics.purdue.edu Internet Source	<%1
18	Submitted to Blake Hall College Student Paper	<%1
19	Lecture Notes in Computer Science, 2016. Publication	<%1
	www.cen-wslt.din.de	

20	Internet Source	<%1
21	www.ojp.usdoj.gov Internet Source	<%1
22	revistaie.ase.ro Internet Source	<%1
23	www.iit-iti.nrc-cnrc.gc.ca Internet Source	<%1
24	Submitted to Strathmore University Student Paper	<%1
25	Submitted to Informatics Education Limited Student Paper	<%1
26	Submitted to Curtin University of Technology Student Paper	<%1
27	pnc.org.nz Internet Source	<%1
28	B. Slof. "Successfully carrying out complex learning-tasks through guiding teams' qualitative and quantitative reasoning", Instructional Science, 09/04/2011 Publication	<%1
29	www.scoop.it Internet Source	<%1
30	Submitted to South Bank University	

	Student Paper	<%1
31	Brown, . "The Evolution of Standards for E-Learning", e-Learning Standards A Guide to Purchasing Developing and Deploying Standards-Conformant E-Learning, 2002. Publication	<%1
32	Submitted to University of Northumbria at Newcastle Student Paper	<%1
33	gr.xjtu.edu.cn Internet Source	<%1
34	Innovations in E-learning Instruction Technology Assessment and Engineering Education, 2007. Publication	<%1
35	www.dcs.bbk.ac.uk Internet Source	<%1
36	www.eotty.com Internet Source	<%1
37	www.annedawson.net Internet Source	<%1
38	scholarspace.jccc.edu Internet Source	<%1
	www.afdb.org	

39 Internet Source <%1

40 J. Kontio. "A case study in applying a systematic method for COTS selection", Proceedings of IEEE 18th International Conference on Software Engineering ICSE-96, 1996 <%1

Publication

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	OFF		