



UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

A WEB CONTENT-BASED RECOMMENDER SYSTEM TO PROMOTE AUTOMATIC
DISCOVERY OF LEARNING CONTENT FOR HIGH SCHOOL STUDENTS.

BY

JOHN NJIU CHEGE

SUPERVISOR

MR. JOSEPH OGUTU

PROJECT REPORT SUBMISSION IN THE PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER
SCIENCE.

NOVEMBER 2014.

DECLARATION

This project as presented in this report is my original work and has not been presented for any other institutional award.

John Njiu Chege

P58/64862/2011

Sign: _____ Date: _____

This project has been submitted in partial fulfilment of the requirements for the degree of Masters of Science in Computer Science at the University of Nairobi with my approval as the university supervisor.

Sign: _____ Date: _____

Mr. Joseph Ogutu

School of Computing and Informatics

University of Nairobi

ABSTRACT

World Wide Web has become a sea of information with billions of published documents and millions of changes happening on a daily basis. For web users, getting lost in this sea of information hyperspace has become a common thing in addition to facing information overload when browsing the web due to the enormous amount of information they get from the web and for high school students using the World Wide Web as a source of learning content, this might pose a great challenge in their effort to retrieve relevant information from the internet. This study has employed a web content based approach to develop a prototype that learns what High School Students are expected to learn in school and personalizes the way they surf the web in search of relevant learning materials from the World Wide Web. The prototype was evaluated by online learners through an inbuilt review module in the prototype, in addition to conducting a survey on learners who used the prototype. The feedback received from the surveys was positive and showed that the prototype had a positive impact on the learners who used it as a way of personalized online learning.

ACKNOWLEDGEMENT

First I would like to acknowledge the Almighty God for this far I have reached is because of his mercies. Second I would like to appreciate the Guidance and positive criticism from my Supervisor Mr. Joseph Ogutu and also the entire panel for the corrections and insights they provided to help improve my project. Lastly My Dear Wife “Gladys” and Daughter “Shii” who have been with me through this learning journey, May God bless you abundantly.

Table of Contents

Chapter 1: INTRODUCTION.....	1
1.1 Background.....	1
1.2 Problem Statement.....	2
1.3 System Objectives	2
1.4 Functional Objectives	3
1.5 Significance.....	3
1.6 Assumptions and limitations	4
2 Chapter 2: LITERATURE REVIEW	5
2.1 Web personalization	5
2.2 Recommender systems	5
2.2.1 Challenges with Recommender systems	7
2.2.2 Types of Recommender Systems	7
2.3 Related Work	11
2.4 The Gap.....	14
3 Chapter 3: METHODOLOGY	15
3.1 Introduction.....	15
3.2 Data sources	15
3.3 Data collection methods.....	15
3.3.1 Forms.....	15
3.3.2 Focus group.....	15
3.3.3 Observation	16
3.3.4 Text books review.....	16
3.4 System design	16
3.4.1 Stage 2: Query stage.....	16
3.4.2 Stage 1: Learn stage	18
3.4.3 Stage 3: Classification Stage	19
3.4.4 Stage 4: Recommendation	19
3.5 Proposed System Architecture	20
3.6 Development tools	20
4 Chapter 4: SYSTEM ANALYSIS AND DESIGN	21
4.1 System specification.....	21
4.1.1 Overview	21
4.1.1 Inputs and outputs.....	21
4.1.2 Data management	21
4.2 System analysis	22
4.2.1 Identifying goal.....	22
4.2.2 Applying use cases	24
4.2.3 Refining roles	25
4.3 System design	27
4.3.1 Overall System Architecture	27
4.3.2 Flow design.....	29
4.3.3 Creating of Sequence Diagram	32
4.3.4 Database Design.....	33

5	Chapter 5: SYSTEM IMPLEMENTATION	35
5.1	System development	35
5.1.1	User Profile module	35
5.1.2	The Searcher module.....	35
5.1.3	The Recommender module	35
5.1.4	The Review module	35
5.2	System Configuration	35
5.2.1	System experimentation/interaction.....	36
6	Chapter 6: RESULTS	39
6.1.1	Overview	39
6.1.2	Results and discussion	39
6.1.3	Summary	43
7	Chapter 7: CONCLUSION AND RECOMMENDATIONS	44
7.1.1	Conclusion	44
7.1.2	Limitations of the study	44
7.1.3	Further work.....	45
7.1.4	Recommendations.....	45
	REFERENCES.....	46
	APPENDIX	49
	Appendix 1: Questionnaire	49
	Appendix 2: Sample Code	51
	The user profile module.....	51
	The search module	52
	Recommendation	55
	Return form	56
	Appendix 3: Data collected during survey	58
	Appendix 4: how to run the system	58

TABLE OF FIGURES

Figure 1: Initial User Registration form	18
Figure 2: Proposed System Architecture Design	20
Figure 3 : Goal Hierarchy Diagram.....	24
Figure 4 : Use Case Diagram.....	24
Figure 5 : Role Model Diagram	25
Figure 6: Concurrency Model Diagram for receiving feedback and providing feedback	26
Figure 7 : Concurrency Model Diagram for searching the Web and Reviewing Feedback ...	26
Figure 8 : System Architecture.....	27
Figure 9: System Flow Diagram.....	29
Figure 10 : learner profile and recommendations process flow	30
Figure 11 : Feedback review process flow.....	31
Figure 12 : Sequence Diagram.....	32
Figure 13 : the Profile module	36
Figure 14: Recommendations Muodules	37
Figure 15 : Rating module	38
Figure 16: User General Perception about the system.....	40
Figure 17: Learner's perception about individual subject recommendation.....	41
Figure 18: online Use Reviews	42
Figure 19: Survey data	58

Chapter 1: INTRODUCTION

1.1 Background

The internet or the World Wide Web as it is commonly referred to has become a sea of information with billions of web pages already published and millions more being published or modified each passing day and for internet users, getting lost in this sea of information is often a common occurrence because of the information overload posed by the excessive information found in the internet. Furthermore, with the presence of so many web crawlers and search engines it has become increasingly difficult for internet users to decide on the best search engine to use for a specific search query and this has resulted in most web users using only one search engine for all their search queries something that might not guarantee them the best search results for some of their queries. Also the difficulty in formulating the right search queries for most of the internet users has led to not so satisfactory search results being obtained from the search queries performed.

For High school students using the World Wide Web as a source of educational material the above underlined challenges may limit the benefits they obtain by using the World Wide Web because the way and reason that they explore the World Wide Web can affect the success of the information retrieval process. It is therefore imperative to come up with simple ways that will help these students to get the right content from the internet with little effort. In addition to putting into consideration the learning or education level of the learner and the subject of study. The proposed approach should be able to classify the content learned from the internet into the right educational level, the right subject of study and automatically propose the learned content to the learner based on the learners' preference.

Web personalization has been recognized as a way of reducing the information overload brought about by the exploding quantity of information found in the World Wide Web by providing users with what they want without the users explicitly requesting for it and in this case the research will leverage the power of web personalization by personalizing the learning experience for high school students through recommending webpages with educational content relevant to the needs of high school students, based on their educational level, subject of interest and the learning styles of the students.

1.2 Problem Statement

The web is a huge repository of information and is growing and changing rapidly something that makes access to the relevant information become increasingly difficult and for high school students it is a challenging task for them to gather relevant study information through performing of queries using search engines as they might not know how to compose their search strings appropriately in order to obtain the best search results. Furthermore, the temptation to spend most of their internet time on non-educational content is so real and highly probable since it is easier and “interesting” to access this non-educational content as opposed to accessing the educational content.

In order to try and address the identified challenges the research proposes an automated way of discovering learning content, relevant to the students, from the available internet sources and automatically propose this content to the students based on their level of education and the subject they are interested in.

To achieve this personalized automatic education content recommendation to the learners the research proposes the use of a supervised learning algorithm to learn the educational content required by the learners through filling of a simple form at the start and after the learning process has been completed a user profile of the learner is created. The system will then query the world wide web for educational information based on the profiles created and after the search results have been obtained the recommender system will use a classification algorithm to classify the learned content into various subject of interest to the learner and finally recommend the titles of the learned URLs to the learners based on the subject in question and education level of the learner. The proposed system should also be able to improve learned learner profiles and also the recommendations done to the students by reviewing, in form of learner feedback, how relevant the recommended online materials were to the learner.

1.3 System Objectives

The objectives of the study includes

- To develop an online recommender prototype that will aid high school students retrieve information easily from the World Wide Web.

- To develop an online recommender system prototype that will aid students in online learning.
- To develop an online recommender system prototype that will promote online learning amongst students by recommending learning content that is of interest to the learners.

1.4 Functional Objectives

These research will try to develop an online recommender system that will personalize the way these learners access the internet and make it easy to learn the educational content based on their educational level and skills, the system will query the web on behalf of the students then gather the URLs with educational content relevant to the learners, classify the gathered URLs into various subject of study and finally recommend the URLs to these learners based on the subject in question and education level of the learner. In general the system will carry out the following tasks,

- Learn: use a supervised learning algorithm to learn the content required by the students.
- Query: query the web to get a list of websites providing similar content.
- Classify: classify the websites queried based on different learning levels of the students (probable algorithm to be used KNN, Naive Bayes).
- Recommendation engine: recommend the websites suitable for the learner based on their profile created when they filled introduction form.

1.5 Significance

At the end of this research we aim to develop a prototype that will help to promote the adoption of technology assisted learning in Kenya schools by assisting the Kenyan learner to easily find educational content online with little or no effort.

The prototype is also aimed at helping the young Kenyan learner in realizing the benefits of the recently launched laptop program by aiding the students under this program to get the much required content from the internet considering that most of the teachers who are supposed to be aiding the students on this tasks have very little computer knowledge and as a result we can achieve a lot if we can develop a web recommender system that will

automatically learn the required content and present it to the learners. The system will also help the Kenyan learner to avoid being over reliant on their teachers since now they can easily get the required educational material from the web.

The research also has significance in creating web personalization at a local level on local content, which will aid in exposing local content available online, a move that will encourage creation of more local content since the creators of this content will have a ready market for their content in the name of the learners and this may bring up new areas of entrepreneurship which in turn will help in economy growth.

1.6 Assumptions and limitations

The assumptions made concerning this study are

- Recommendations made are relevant study material to Kenyan curriculum.
- The recommender system will only cover recommendations for mathematics only and that's for form one and form two students.

Chapter 2: LITERATURE REVIEW

2.1 Web personalization

Information retrieval techniques have matured with time and search engines have done a great job indexing online content, if the user knows exactly what they want, there is a lot of information that one can get from the internet and sometimes this information can be too much for the end user. The goal of personalization is to provide users with what they want or need without requiring them to ask for it explicitly (Mulvenna, 2000). Web personalization systems doesn't have to be fully automated but it encompasses a scenario where the user is not sure of what they exactly need and by interacting with the system, it can lead them to the items of interest.

Web Personalization has been viewed as an application of data mining and machine learning techniques to build models of user behavior that can be applied to the task of predicting user needs and adapting future interactions with the ultimate goal of improving user satisfaction (Sara jot S. A., 2005).

The web data that may need to be personalized can be grouped into three broad categories namely content data ,this is what the web pages contain, structural data which describes the graph structure among the data contained in the web pages and lastly usage data which describes user actions or behaviors and is mostly stored in server logs.

Systems that have been used to provide web personalization are termed as recommender systems since they learn the user likes or dislikes and then proposes content, items or services based on the learned user profile.

2.2 Recommender systems

With the increase in the amount of information available on the internet, there is a challenge of providing relevant and useful information to the internet users on the basis of their interest although when users wants to search data of their interest, they can do a search in whole databases, which is very tedious and time consuming too. So a system is needed to provide useful information based on user interest and such systems are called Recommendation Systems.

Recommenders systems (RS) can also be described as software agents that elicit the

preferences and interest of individual consumers and make recommendation accordingly. They have the potential to support and improve the quality of the decisions consumers make while searching for and selecting products online (Xiao & Benbasat, MISQ, 2007).

Burke (2002) defined Recommendation system as “any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options”.

RS can also be termed as a subclass of information filtering systems that seek to predict the “rating” or “Preference” that user would give an item or social element that they have not considered yet, using a model built from the characteristics of an item or user's behavior.

In recent years recommender systems have become increasingly popular with best-selling e-commerce websites like amazon.com and Netflix heavily relaying on this technology to promote personalized shopping experience to their online shoppers. These systems have proven to be a valuable means of dealing with information overload problem by pointing a user towards new, not yet experienced items that may be relevant to the current user's needs. Recommender systems are now popular in both Research as well as commercial communities and can be found in many modern applications that expose the user to a huge collection of items. These systems help users to decide an appropriate item, and ease the task of finding items in the collection and they are all typically categorized as RS, even though they provide diverse services.

According to Francesco R. et al (2011) RS is a multidisciplinary field that have benefited from results obtained from various computer science fields especially machine learning, data mining, information retrieval and human-computer interaction. RS have been used widely to help organizations achieve the following:

- To increase the number of items sold by making recommendations that suit the user's needs.
- Sell more diverse items by using collaborative techniques.
- Increase the user satisfaction through suggestion of items that the users may be looking for.
- Increase user fidelity by improving how the user interacts with the online system.
- Better understand what the user wants through analysis of online user behavior and

interactions.

2.2.1 Challenges with Recommender systems

While recommendation systems are gaining popularity in today's websites as a means of generating a better user experience on these websites and in return more user traffic to the websites, RS are faced by a number of challenges that limit their implementations.

- Scalability: A recommendation system that works well with a small data set may become inefficient or totally inapplicable of providing good recommendations on very large datasets.
- Proactive recommender systems: most RS tend to adopt a “Pull” approach whereby the users have to first interact with the system before the system can learn the correct recommendations to make to the user. This is also called the “Cold start” problem.
- Privacy concern: in the attempt to make the best personalized recommendations RS systems collect as much data as possible about the user and this have been found to raise some privacy concerns.

2.2.2 Types of Recommender Systems

RS can be can be categorized into six broad categories depending on the way they carry out their recommendations to users.

The first class of RS systems is Content based RS. This is a class of RS systems that recommend items that are similar to what the user has liked in the past. In Content-based filtering systems, a user profile represents the content descriptions of items in which that user has previously expressed interest (Blochl, M., Rumershofer, H. and Wob, W., 2003) and the content descriptions of items are represented by a set of features or attributes that characterize that item. The recommendation generation task in such systems usually involves the comparison of extracted features from unseen or unrated items with content descriptions in the user profile (Mulvenna, M., Anand S.S., Buchner, A.G, 2000) and Items that are considered sufficiently similar to the user profile are recommended to the user.

Content-based RS especially those used in the web and e-commerce applications have their content descriptions as textual features extracted from web pages or product description and

these recommendation techniques have their roots in Information-retrieval (Salton et al, 1983). In Content-based RS both user profiles, as well as, items themselves, are represented as weighted term vectors e.g. based on TF.IDF term-weighting model and Predictions of user interest in a particular item can be derived based on the computation of vector similarities e.g. using the Cosine similarity measure or using probabilistic approaches such as Bayesian classification (Pasquale L., Marco de Gemmis and Giovanni, S., 2011).

In content-based RS the recommendation process basically consists matching up the attributes of the user profile against the attributes of a content object. The Profiles are generated either implicitly (learning user behavior) or explicitly (through questionnaires or through forms to collect user preference).

The primary drawback of content-based filtering systems is their tendency to overspecialize the item selection since profiles are solely based on the user's previous rating of items and User studies have shown that users find online recommenders most useful when they recommend unexpected items (Sinha, 2001), suggesting that using content similarity alone may result in missing important "pragmatic" relationships among Web objects such as their common or complementary utility in the context of a particular task. Furthermore, content-based filtering requires that items be represented effectively using extracted textual features which are not always practical given the heterogeneous nature of Web data (Pasquale et al, 2011).

The second class of RS systems are the Collaborative Filtering (CF) systems .CF makes recommendations to users based on what other users with similar interests have liked in the past. CF calculates the similarity between two users, based on the similarity in the rating history of the users. These systems are considered the most popular and widely implemented RS systems. Schafer et al (2001) refers to this system as "people-to-people correlation" since they make their recommendations to a user based on the likes or dislikes of other people who have shown similar interest in the past.

CF systems have tried to address the shortcomings of content-based RS by matching the ratings of a current user for objects (e.g. movies or products) with those of similar users (nearest neighbors) in order to produce recommendations for objects not yet rated or seen by an active user. Traditionally, the primary technique used to accomplish this task is the standard memory-based k -Nearest-Neighbor (kNN) classification approach which compares a

target user's profile with the historical profiles of other users in order to find the top k users who have similar tastes or interests (Pasquale et al, 2011).

Collaborative filtering systems have also been found to have two major shortcomings in that they lack Scalability due to performing the user profiling as an online process and as the numbers of users and items increase, this approach may lead to unacceptable latency for providing recommendations or dynamic content during user interaction. Another limitation of CF emanates from the sparse nature of the dataset i.e. as the number of items in the database increases, the density of each user record with respect to these items will decrease. This, in turn, will decrease the likelihood of a significant overlap of visited or rated items among pairs of users resulting in less reliable computed correlations (Pasquale et al, 2011). Furthermore, collaborative filtering usually performs best when explicit non-binary user ratings for similar objects are available and In many Web sites it may be desirable to integrate the personalization actions throughout the site involving different types of objects, including navigational and content pages, as well as implicit product-oriented user events such as shopping cart changes, or product information requests.

Another common shortcoming of CF is the "Cold start" problems, in that, if there is no initial user data the system is unable to make recommendations.

The third class of RS is the Demographic recommender systems which recommends items based on the demographic profile of the user. This is a very new class of RS and very little research has been conducted in this area. An example of Demographic RS is an RS that redirects people to a particular website based on their country of origin or language.

Knowledge-based RS is another class of systems that recommend items based on specific domain knowledge about how certain item features meet user's needs, preferences and ultimately, how the item is useful for the user. Notable knowledge based recommender systems are case-based RS. In these systems a similarity function estimates how much the user needs (problem description) match the recommendations (solutions of the problem). Here the similarity score can be directly interpreted as the utility of the recommendation for the user (Alexander, F., Gerhard, F., Dietmar, J. and Markus, Z., 2011). Another notable Knowledge based RS is Constrained based RS and they make their recommendations by exploiting predefined knowledge bases that contain explicit rules about how to relate customer requirements with Item features (Alexander et al , 2011).

Knowledge based recommendation System is completely reasoning oriented and it has quantitative decision support tools (Bhargava, H.K., Sridhar, S., Herrick, C., 1999) which allow customers to recommend better support for their judgments and such recommendation systems are used to remove Statistical anomalies. Based on information, the system pursues a knowledge based approach to generating a recommendation, by reasoning about what products meet the user requirements. It avoids some of the drawbacks because it does not have a ramp up problem as recommendations does not depend on rating i.e. its judgment is independent of individual tastes.

Community based RS is another class of RS systems that takes advantage of the trust that people have in other people they know and recommends items based on the preferences of the user's friends. Sinha et al (2001), supports this approach by stating that evidence suggests that people tend to rely more on recommendations from their friends than on recommendations from similar but anonymous individuals. This observation combined with the rise in social networks are making the community based RS to increase in their popularity and use. Community based RS have been widely used in social networks and a classic example is in Facebook where users get recommendations based on what people who are their friends have shown interest in.

The last class of recommender systems is called Hybrid recommender system and this is RS that combine two or more of the above named RS techniques to make recommendations to users.

RS have been heavily relied on in the fields of e-commerce and online advisements but recently they have started to be adopted in the Educational fields mostly to assist e-learning platforms to deliver relevant content to the target audience.

2.3 Related Work

Aggarwal, c.c., wolf, j.l., wu, k.l. and yu, p.s. (1999) identified the two major approaches adopted in recommendation of learning content as collaborative filtering and content-based approaches. They identified content-based systems as systems that matches the content of candidate items against the user profile and the user profile is constructed by analyzing the content of items that the user has favored in the past or user's personal information and preferences.

The content-based approach, as Mooney and Roy (2000) noted, has the advantage of being able to recommend previously unrated items to users with unique interests and to provide explanations for its recommendations and this is the key reason why the method has been widely employed in practical applications. They (Mooney et al, 2000) described a content-based book recommending system that utilizes information extraction and a machine-learning algorithm for text categorization. Mooney et al (2000) also found Collaborative filtering approach, which works by matching current user preferences to those of other customers in making recommendations, to be capable of producing high quality recommendations during a live customer interaction.

Pao-Hua, C. and Menq-Jiun, W. (2009) identified personalized curriculum sequencing as an important research issue for web-based learning because no fixed learning path is appropriate for all learners and consequently many researchers have focused on developing e-Learning systems with personalized learning mechanisms to assist on-line web-based learning and to adaptively provide learning paths for the e-learners. Pao-Hua et al (2009) found some shortcomings with the two major approaches that have been widely used to provide web personalization i.e. collaborative filtering and content-based approach as proposed by Aggarwal et al (1999) because this two approaches perform unsatisfactorily without large amount of usage data and they suggested use of a hybrid method that will take the best of both worlds and eliminate shortcomings of each method. Pao-Hua et al (2009) suggested the use of a content-based approach in creating the learner's profile from the learner click stream and a prior knowledge about the learner's learning level and then employing a collaborative filtering technique combined with Artificial Neural Network Back Propagation technique to predict suitable recommendation or curriculum sequencing for the learner.

Chih-Ming, C., Hahn-Ming, L., Ya-Hui, C. (2004) showed that too many hyperlink structures

in Web-based learning systems place a large information burden on learners and as a result, in Web-based learning, disorientation (getting lost in hyperspace), cognitive overload, lack of an adaptive mechanism and information overload are the main research issues. Chih-Ming et al (2004) proposed a personalized e-learning system based on Item Response Theory which considers both course material difficulty and learner's ability, to provide individual learning paths for the learners. In addition Chih-Ming et al (2004) adopted Maximum likelihood estimation (MLE) to obtain the precise estimation of the learner's ability based on explicit learner's feedback and a collaborative voting approach to adjust the course material difficulty for the items recommended to the learner.

Altered Vista system was an early attempt to develop a collaborative filtering system(Recker & Walker , 2003) and it explored how to collect user-provided evaluations of learning resources and then to propagate them in the form of “word-of-mouth” recommendations about the qualities of the learning resources. The team working on Altered Vista explored several relevant issues, such as the design of its user interface, the development of non-authoritative metadata to store user provided evaluations, the design of the system and the review scheme it uses as well as results from pilot and empirical studies from using the system to recommend to the members of a community both interesting resources and people with similar tastes and beliefs (Recker et al, 2003).

Rafaeli, S., Barak, M., Dan-Gur, Y. & Toch E. (2004) developed Questions Sharing and Interactive Assignments (QSIA) for learning resources sharing, assessing and recommendation as a way of online learners helping each other to locate relevant online learning content. This system was used in the context of online communities in order to harness the social perspective in learning and to promote collaboration, online recommendation and further formation of learner communities. Instead of developing a typical automated recommender system, Rafaeli et al (2004) chose to base QSIA on a mostly user-controlled recommendation process where the user can decide whether to assume control on who advises or to use a collaborative filtering service.

In educational domains, Tang & McCalla (2005) suggests that recommendations made by the RS systems should be made not only to suit learners' interests, but also to keep them engaged and pedagogically motivated throughout the learning process. Tang et al (2005) supports the school of thought that proposes recommendation system should aim to reduce the cognitive load on the learner because they are directly and automatically guided in their actions.

Peter, D., Nicola, H., Wolfgang, N., Michael, S. (2004) proposed a service-based architecture for establishing a personalized e-Learning system, where personalization functionality is provided by various web-services. A Personal Learning Assistant integrates personalization services and other supporting services and provides a personalized access to learning resources in an e-Learning network. Peter et al (2004) aimed to realize personalized learning support in distributed learning environments based on Semantic Web technologies and used an ontology that was updated automatically to meet the learner's needs.

Khribi, M. K., Jemni, M., & Nasraoui, O. (2009) described an automatic personalization approach aiming to provide online automatic recommendations for active learners without requiring their explicit feedback. The proposed approach was taking into account both the Web access history of learners as well as the content of the learning material. Web mining techniques in combination with an open source Web information retrieval system were used to enable an implementation that is not only open and scalable, but also fast to deploy. The recommender system they aimed to develop was to be considered as an external module or plug-in that can be included easily in e-learning systems to give automatic personalization. Khribi et al (2009) created an off-line module which pre-processes data to build learner and content model in addition to an on-line module which uses these models on-the-fly to recognize student goals and predict a recommendation list in form of a list of URLs referencing educational resources (hosted and/or created inside the e-learning platform) in order to guide and support the e-learners. The recommended learning objects were obtained by using a range of recommendation strategies based mainly on content based filtering and collaborative filtering, each applied separately or in combination. Khribi et al (2009) used cosine similarity to calculate the measure of similarity of the learners and cluster them into various groups based on their profiles using k Nearest Neighbors (KNN).

Bloch, M., Rumershofer, H. & Wob, W. (2003) proposed an adaptive learning system which can incorporate psychological aspects of learning process into the user profile to deliver individualized learning resource. The user profile is placed in multi-dimensional space with three stages of the semantic decisions: cognitive style, skills and user type, however, both the means to acquire user's feedback and the algorithms to update user profile were not addressed in their presentation.

Mei-Hua, H. (2008) developed an online personalized English learning recommender system capable of providing students with reading lessons that suit their different interests and

therefore increase the motivation to learn. The system employed content-based analysis, collaborative filtering, and data mining techniques. The system analyzed real students' reading data and generates recommender scores which helped to select appropriate lessons for respective students. Mei-Hua (2008) used clustering and association rule mining to find out synchronous relationships by analyzing the random data and to use this data as reference during decision making (Agrawal, Imielinski, & Swami, 1993).

In his research Jie, L. (2004) proposes a framework of a personalized learning recommender system, which aims to help students find learning materials they would need to read and developed two related technologies under the framework: one is a multi-attribute evaluation method to justify a student's need, and another is a fuzzy matching method to find suitable learning materials to best meet each student's need with the aim of supporting students online learning more effectively and assist large class online teaching with multi-background students. Fuzzy set technique which can describe complex and uncertain relationships and also can deal with unstable classifications (Zadeh, L. A., 1965) was used by Jie, L. (2004) to evaluate the similarities of the learning material for recommendation to learners based on their requirements.

Silvia, S., Patricio, G., Analia, A.(2008) presented an eTeacher systems, an intelligent agent that provided personalized assistance to e-learning students by observing student behavior while taking online courses and automatically creating a profile for the learner. Silvia et al (2008) used Bayesian networks to infer the learning style of the student.

2.4 The Gap

Whereas web personalization has been widely deployed on e-commerce and online advertising little has been done to apply the same technology in the education sector to promote technology assisted learning. This research aims to aid bridge this gap by using web personalization to assist high school students obtain relevant learning content from the internet.

Chapter 3: METHODOLOGY

3.1 Introduction

This chapter will describe the approach used in gathering the required data and developing the prototype for recommendations. The chapter starts by presenting the various methods that will be used for initial data collection and the methods that's will be used in generating the recommendations to the students and finally goes ahead to describe the system design and architecture.

3.2 Data sources

The source of data will mainly be from the recommended school text books, questionnaires presented to high school teachers and interaction with high school teachers and students. Information obtained from this data sources will help in developing a learning algorithm that can create a user profile and use the profile created to learn the learning requirements of the learner and then make useful recommendations to the learner.

3.3 Data collection methods

3.3.1 Forms

On first interaction with the system the students will be required to fill an easy form that will be used as an explicitly way to solicit for their requirements. From the input of the students, the prototype will use a learning algorithm to create a student profile that will be used in making recommendations.

3.3.2 Focus group

To get to understand the learning process of the students, we will interact with the groups of students in their learning environment so that we get to understand common learning styles amongst them.

We will also interview some school teachers so that we can understand the content that the students are required to learn and also which learning styles are most prevalent amongst the learners.

3.3.3 Observation

We will also observe the high school students as they are taught in a class so that we can gather data on how they synthesis what is taught in class.

3.3.4 Text books review

In order to understand what the high school students are expected to learn we will review the course syllabus so that we can put into context what is required of the students. This review will also support the process of creating a user profile as it's from the items that the learner needs to learn that the system will do the learner profiling from.

3.4 System design

The system will have four iterative stages namely;

3.4.1 Stage 1: Query stage

In the query stage the prototype will use the keyword vector space profile created in the first stage to query the web for information related with what is contained in the user profile vector and then store the returned pages temporary on the local user machine as a feature vector. The crawler used to perform search query will be Google and Bing so that we can get a good hit of our search string.

After the feature vector has been obtained using the first n items returned by the search using the web crawlers TF-IDF (Term Frequency –Inverse Document Frequency) is applied as a central tool in scoring and ranking a document's relevance given a user profile.

3.4.1.1 Why TF-IDF

TF-IDF was chosen as a means of weighting the terms of the returned webpages and also determine word relevance on the web pages. TF-IDF was identified as the best algorithm to use for scoring webpage document relevance because of the following reasons

- IDF assumption - rare terms are not less relevant than frequent terms.

- TF assumption - multiple occurrences of a term in a document are not less relevant than single occurrences
- Normalization assumption - long documents are not preferred to short documents

3.4.1.2 Application of TF-IDF

N was used to represent all the first n web pages returned by the search (corpus), n_k denotes the number of webpages in the collection in which the term t_k occurs at least once and d_j presents a document j .

$$\text{TF-IDF}(t_k, d_j) = \underbrace{\text{TF}(t_k, d_j)}_{\text{TF}} \cdot \underbrace{\log \frac{N}{n_k}}_{\text{IDF}}$$

Equation 1

Maximum score is computed over the frequencies $f_{z,j}$ of all terms t_z , that occur in document d_j .

$$\text{TF}(t_k, d_j) = \frac{f_{k,j}}{\max_z f_{z,j}}$$

Equation 2

In order for the weights to fall in the $[0,1]$ interval and for the documents to be represented by vectors of equal length, weights obtained will be normalized by cosine normalization

$$w_{k,j} = \frac{\text{TF-IDF}(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} \text{TF-IDF}(t_s, d_j)^2}}$$

Equation 3

3.4.2 Stage 2: Learn stage

To start with, the user will be required to fill a form which will be used as an explicit method of collecting data from the intended system users. During this phase the system will solicit for items of interest from the expected system users, in this case the students, from the forms.

Using the explicit information obtained from the forms the system will create a user profile for that user by applying a supervised learning algorithm using the user provided information as a set of positive examples to the algorithm. The profile created at this stage will be a set of keywords presented in terms of an n-dimensional vectors space. A sample form that will be used by the system is show on figure 1 below.

The learning stage will be a continuous process and the stage will also make use of explicit feedback provided by the users on how relevant the recommended materials were.

The figure shows a registration form with the following fields:

- Name:
- Age:
- Class:
- Subject of Interest:
- Your Proficiency in the Subject of Interest:
- Sub Area of Interest:
- Your Proficiency in the Sub Area of Interest:

Figure 1: Initial User Registration form

3.4.3 Stage 3: Classification Stage

To carry out the classification we will compute the cosine similarity of the N webpages and the URLs of the web pages with the highest measure of similarity to the user profile created in stage 1 will be stored for recommendation to the user.

$$\text{sim}(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k w_{ki}^2} \cdot \sqrt{\sum_k w_{kj}^2}}$$

Equation 4

3.4.4 Stage 4: Recommendation

Recommendations will be done to the user inform of web page titles of the URLs of the pages with the highest similarity score as determined by the stage 3 of the recommender system.

3.5 Proposed System Architecture

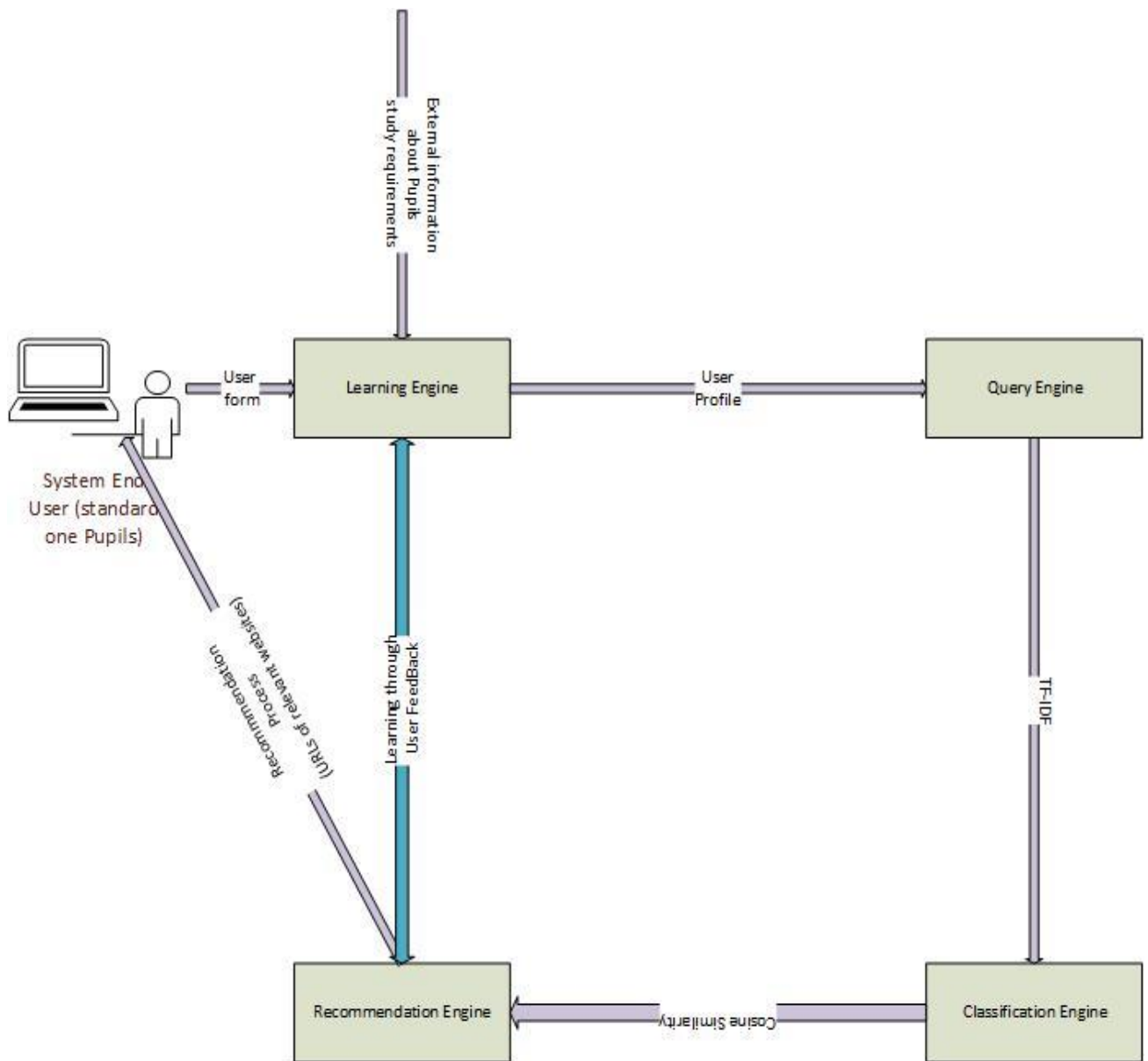


Figure 2: Proposed System Architecture Design

3.6 Development tools

The prototype will be developed using the following development tools

- Java programming Language for the web interface
- Apache Mahout as the backend system for Machine learning

Chapter 4: SYSTEM ANALYSIS AND DESIGN

4.1 System specification

4.1.1 Overview

Currently when students need to access learning material from the internet, they need to formulate a keywords search query and use a search engine to query the internet for the required learning content and after the search results have been obtained the learners have to decide on the web pages that best address their needs by reviewing some of the pages returned by the search engine.

The proposed system will minimize the need for the learner to create a search string manually and it will use a learner profile created when the user first logs into the system to query the web and make the recommendations in form of webpages that best fits the learner profile.

The system will make learning materials recommendation to students by calculating a similarity measure between the learner profile and the webpages returned from a web query performed on several search engines, in this case Google, Bing and Yahoo and return the URLs and titles of the pages with the highest similarity score.

4.1.1 Inputs and outputs

4.1.1.1 Inputs

As inputs the system will require a user to fill an initial form where their information will be captured in form of a user profile and this information will include a username which will act as a unique identifier, the subject of interest to the learner, the topic and subtopics that the user is interested in, the learner's learning level on the stated topic and their preferred learning style.

4.1.1.2 Outputs

As outputs the system will query the web for items that match the learner profile then compute a similarity score between the items queried and the learner profile, and return the Titles and URLs of the websites with the highest similarity score to the learner's profile. The learner can then follow the recommended links to access learning content that are relevant to their needs.

4.1.2 Data management

The system will carry out in-memory recommendations using Mahout as the backend

recommendation engine while the user profile and user feedback will be stored in a Mysql Database.

4.2 System analysis

The purpose of the analysis phase is to produce a set of roles whose tasks describe what the system has to do to meet its overall requirements. A role describes an entity that performs some function within the system.

System analysis will involve the following three steps:

- Identifying goals from user requirements and structuring them into a Goal Hierarchy diagram
- Identifying use cases and creating sequence diagrams to help identify an initial set of roles and communications paths
- Transforming the goals into a set of roles

4.2.1 Identifying goal

Goal identification is the first step in the analysis phase, which takes an initial system specification and transforms it into a structured set of system goals. This process will involve

a) Capturing Goals

This process begins by extracting scenarios from the initial specification and describing the goal of that scenario. The following are the scenarios from our initial specification:

- The system is responsible for making recommendations for learning items to high school students.
- The Learner's learning needs will be captured through an initial form that the learners will be required to fill on their first interaction with the system.
- Recommendations done will be based on the user profile created.
- Results obtained from the system should come from several search engine
- A Similarity score will be calculated to check how similar the items returned from the web query fit the user profile.
- The system will provide recommendation to the learner's based on the similarity score obtained.
- The system will provide a feedback mechanism where the users can gauge how useful the recommended items were.

From the above scenarios, the derived goals are

- Capture the learner's needs.
- Create a user profile.
- Query the web on behalf of the learner.
- Calculate relevance of the results obtained from the web query in relation to the learner's needs.
- Make recommendations for learning content to learners.
- Capture user feedback

b) Structure the Goals

After capturing of goals, the goals are put in a hierarchy depending on the importance and level of detail.

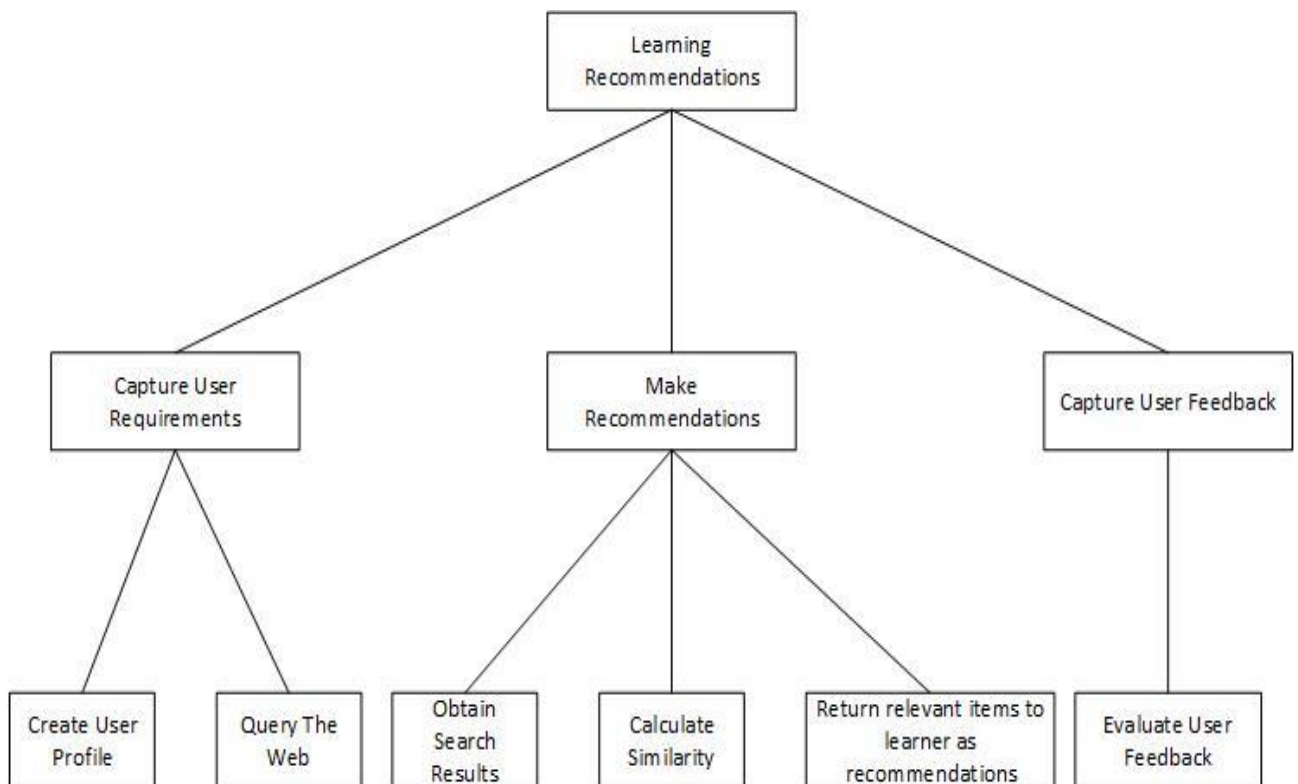


Figure 3 : Goal Hierarchy Diagram

4.2.2 Applying use cases

The application of the use cases in this case is to help us to define interactions between a role and the system and how this interaction achieves the identified goals.

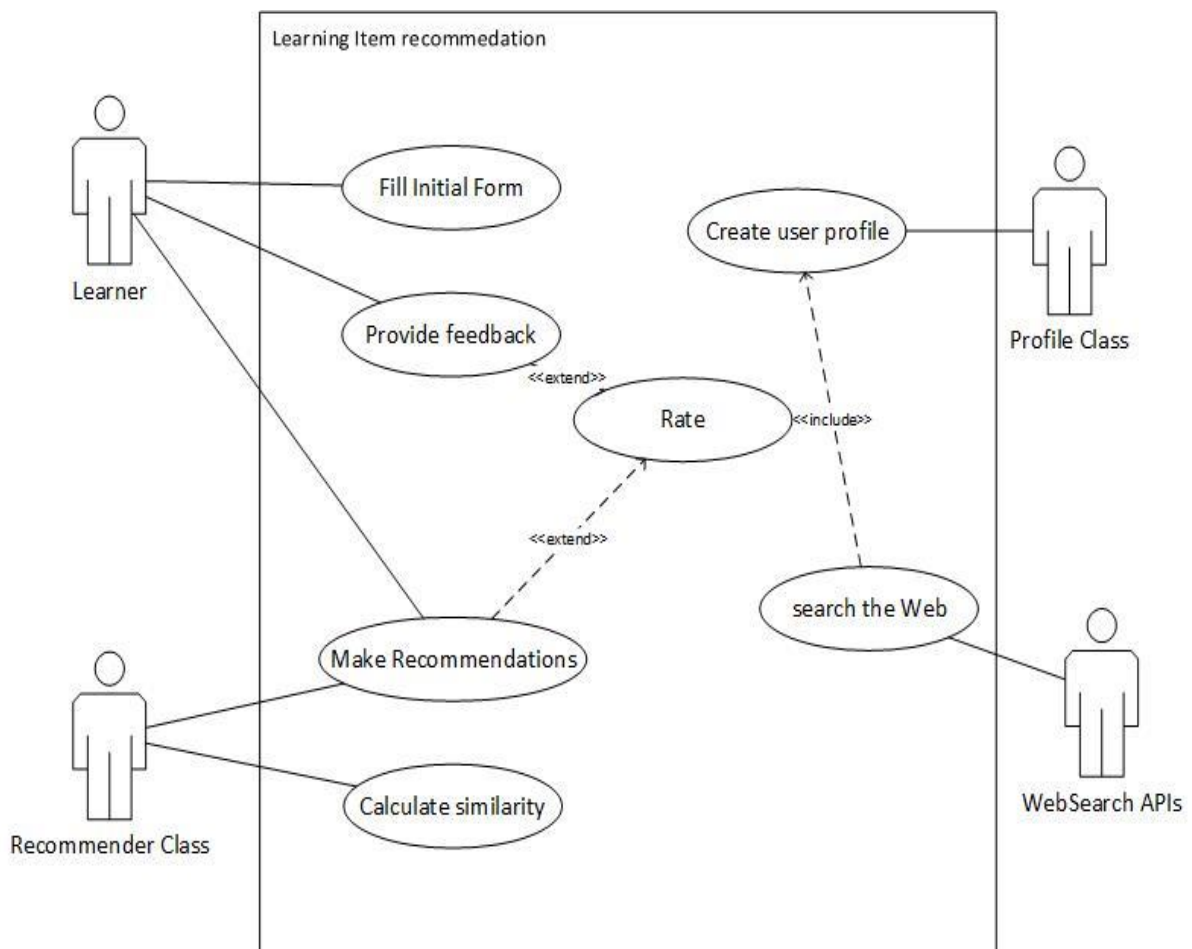


Figure 4 : Use Case Diagram

4.2.3 Refining roles

The objective of this step is to transform the structured goals and sequence diagrams into roles and their associated tasks. To generate tasks we will use the identified goals and create tasks that will be responsible for each goal and this is captured in a role model diagram as shown below.

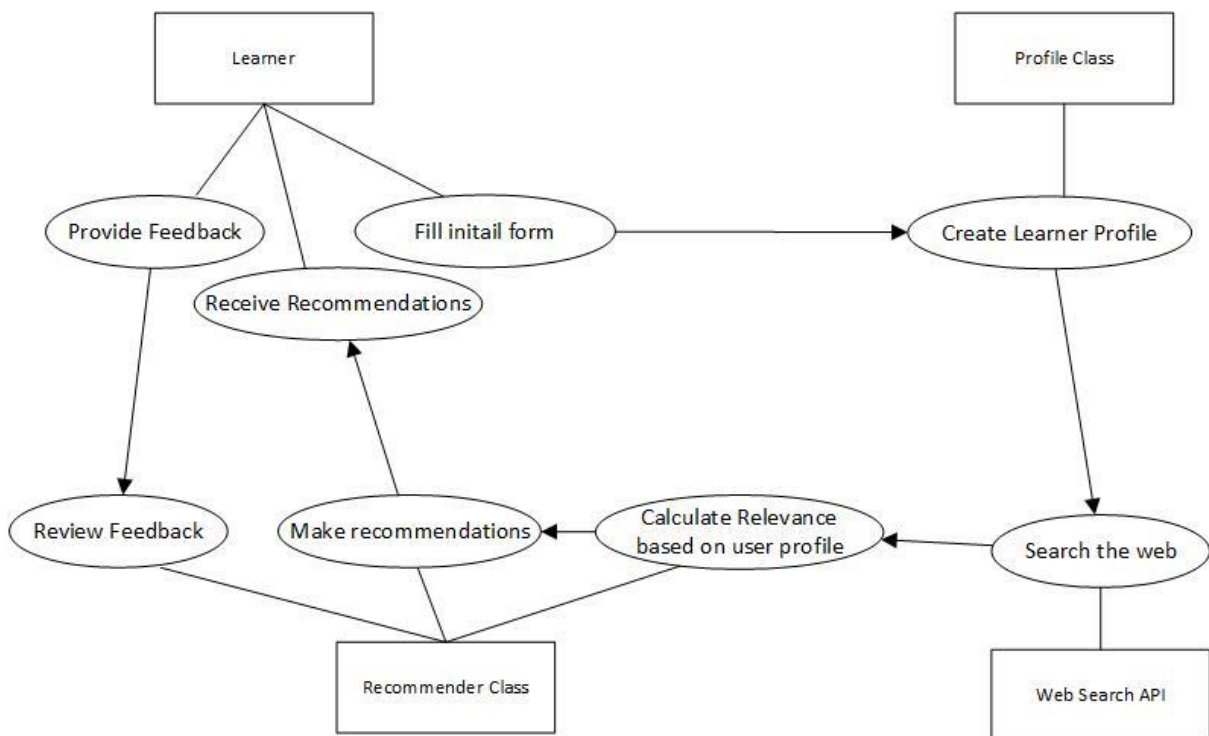


Figure 5 : Role Model Diagram

The roles of making recommendations and that of receiving feedback reviews may occur concurrently to define the role behaviours. Also the role reviewing feedback and that of searching the web are possible concurrent roles. These concurrent behaviour are expressed as below in concurrency model diagram.

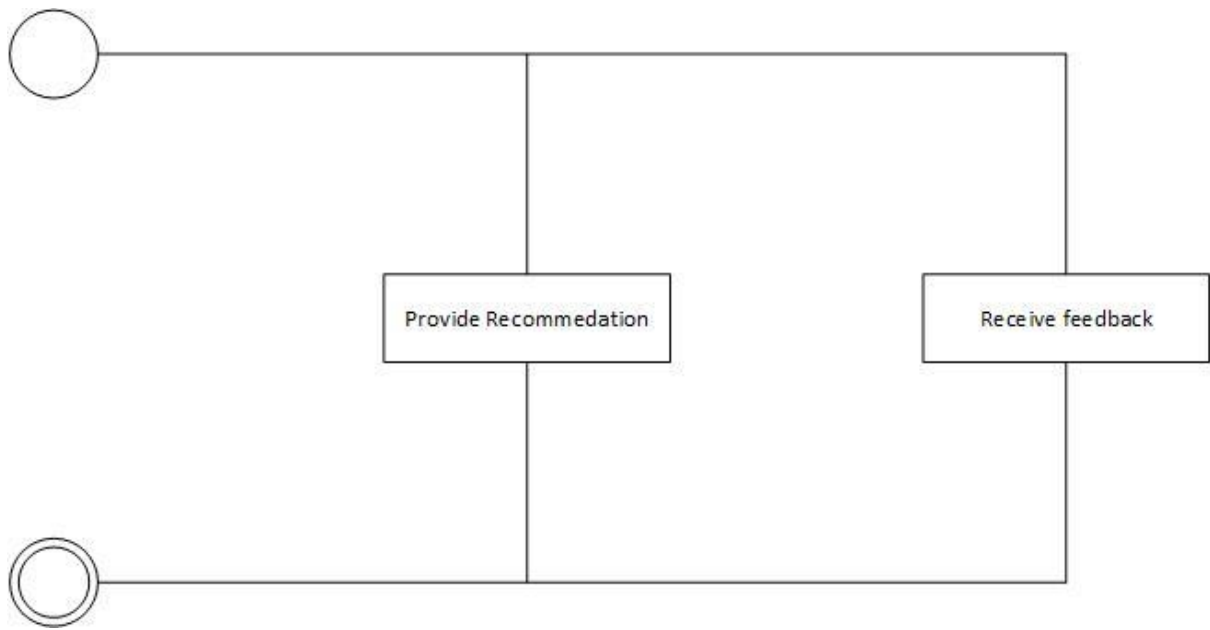


Figure 6: Concurrency Model Diagram for receiving feedback and providing feedback

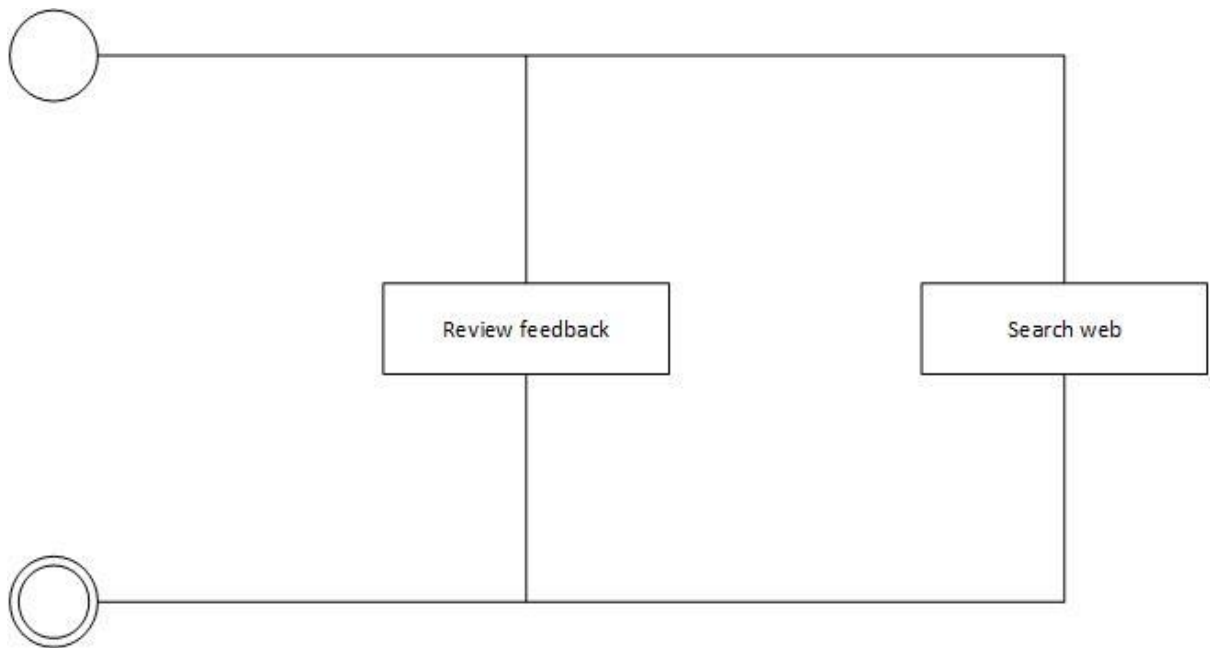


Figure 7 : Concurrency Model Diagram for searching the Web and Reviewing Feedback

4.3 System design

4.3.1 Overall System Architecture

The system will be inform of a client-server architecture where the client will be the front end forms that the learners will fill when they first interact with the system. This client will be wed based and the server will be a java mahout backed end system that queries the web, computes the similarities scores for the items returned from the web queries and makes recommendations through the front end interface for items that best suites the learner's needs. The system will also have a feedback mechanism where the leaners can give their feedback on how useful the recommendations were to their learning needs. The learner's Profile and the learners' feedback will be saved on a MySQL database.

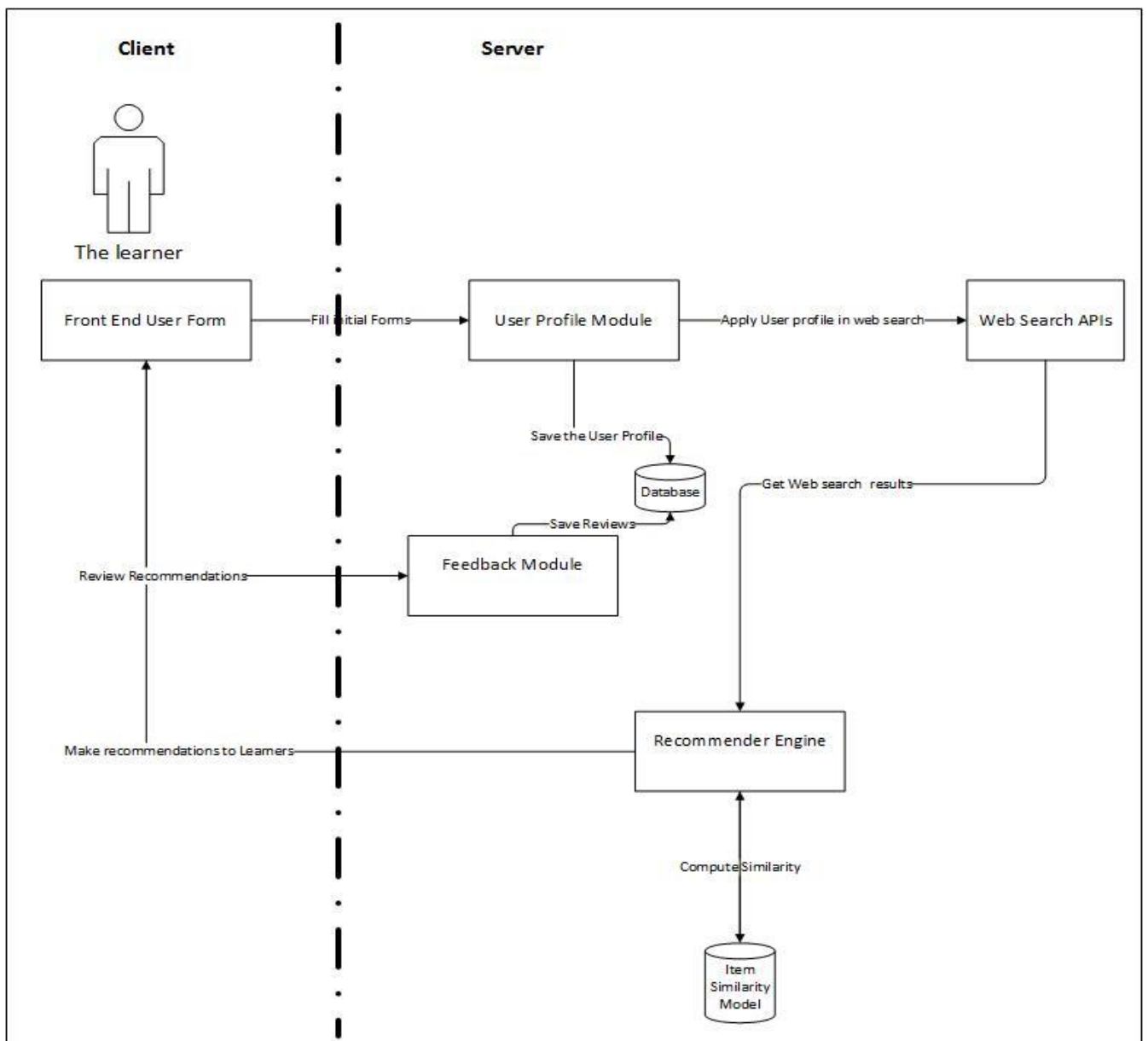


Figure 8 : System Architecture

Development tools that will be used includes

- Java – This will be the programming language
- Apache mahout – This is the recommendation engine
- MySQL- This act as the data store where the learner profile and reviews will be saved
- Apache Tomcat – This will be the webserver for the learning items

4.3.2 Flow design

The overall system flow will be as below

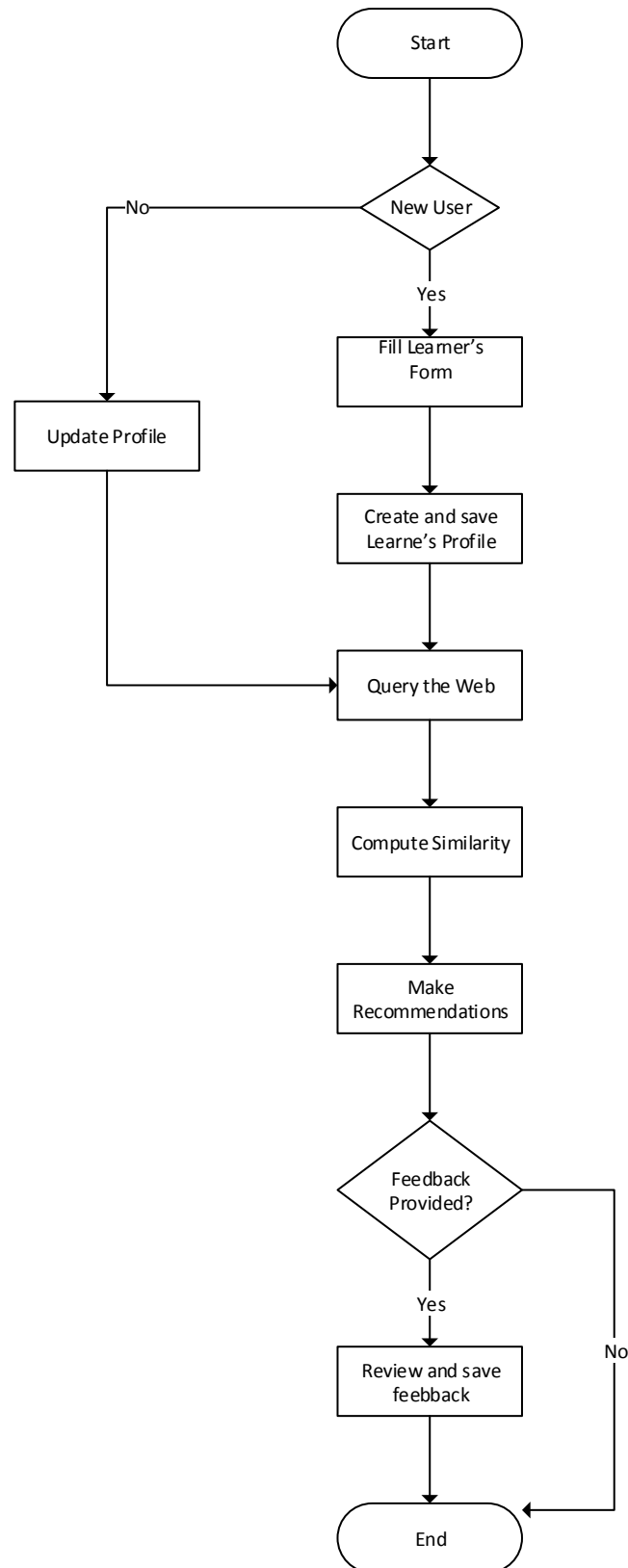


Figure 9: System Flow Diagram

4.3.2.1 Create learner profile and receive recommendations process flow

This process flow captures the activities that the learner will do until they receive the recommendations from the system

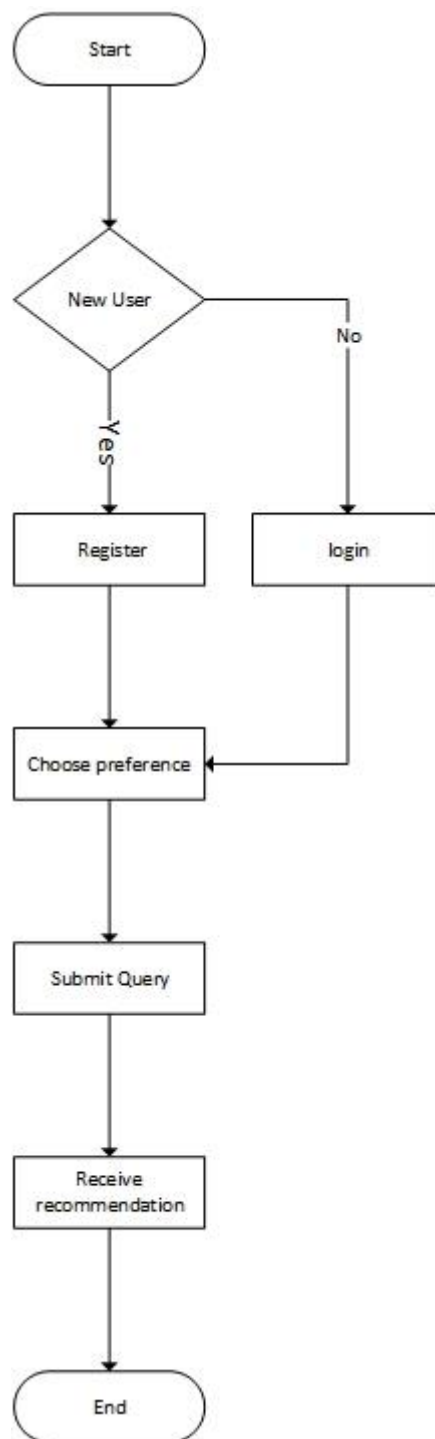


Figure 10 : learner profile and recommendations process flow

4.3.2.2 Feedback review process flow

This shows the process that a learner who decides to provide review will have to follow

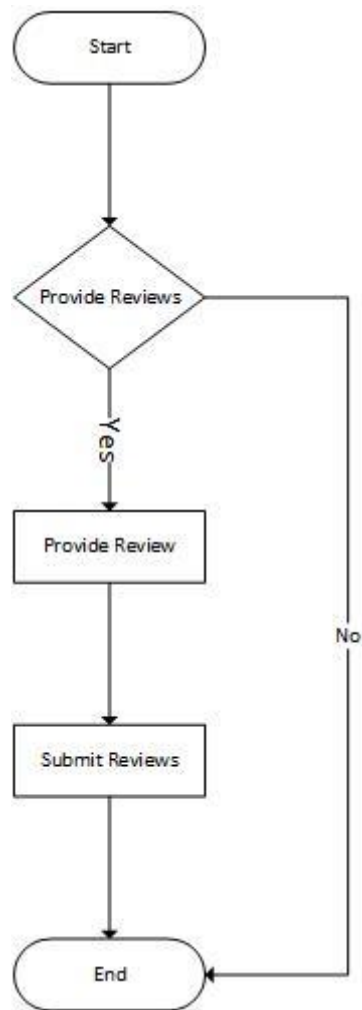


Figure 11 : Feedback review process flow

4.3.3 Creating of Sequence Diagram

A Sequence Diagram depicts the sequence of events that are transmitted between the tasks that are performed by the system. The tasks described are summed up in a sequence diagram.

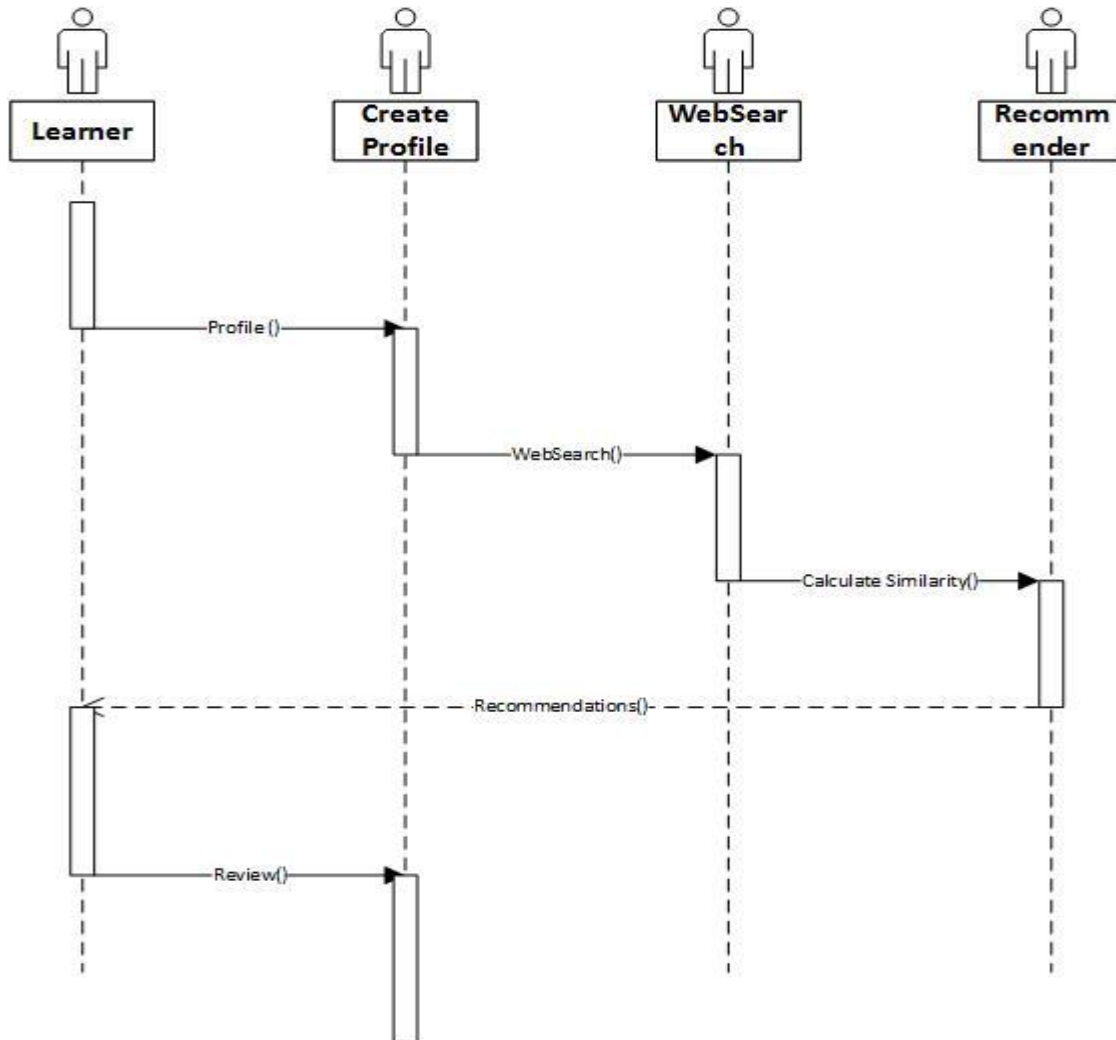


Figure 12 : Sequence Diagram

4.3.4 Database Design

The database stores the information about the system users and their login information. It also stores the user preferences and keeps track of reviews provided by the learners.

a) User's Table

This the table that will contain information about the learners that will use the system

Name	Data Type	Length	Description
Username	varchar	50	Username for the system users. This will be the primary Key
password	varchar	50	Password set by the user for logging in to system

b) User Profile Table

This Table will save the user profile created by the system showing their previous preferences

Name	Data Type	Length	Description
Username	varchar	50	Username of the learner
topic	varchar	20	The topic that the learner is interested in learning about
SubTopic	varchar	100	The specific topic that the user may be interested with , with the topic selected
Skill	varchar	20	The Skill level of the learner in the topic selected
Keywords	varchar	50	Any keywords that the learner may want included in the search string

c) User Feedback Table

This table will save the reviews that the learners provides on how useful the recommendations were in relation to their learning needs.

Name	Data Type	Length	Description
Username	varchar	50	Username of the learner
Query	varchar	100	The Query that the user submitted and is now providing review about
Rating	varchar	10	How useful was the recommendations

Chapter 5: SYSTEM IMPLEMENTATION

5.1 System development

The system will be implemented in Java Eclipse developer EE and the following prerequisites will be needed in order to run the application correctly.

- Apache Tomcat webserver – this is a webserver to aid learners to access the system through a web interface.
- MySQL – This is the data store where all the information will be stored.

The development was done in modules and the below modules were identified

5.1.1 User Profile module

This is the module that creates the learner profile which is used in performing the web searches. This module is creates the user profile by prompting the learners for their subjects of interest.

5.1.2 The Searcher module

This is the module that searches the web for the items identified in as being of importance to the learner. This module connects to APIs of the various search engines and queries for items similar with what is in the learner's profile

5.1.3 The Recommender module

This is the module that takes the search results, calculates how well the returned items satisfies the learner's needs and makes recommendation inform of the web Titles and URLs for the items that best fits the learner profile.

5.1.4 The Review module

This module is used by the learner to provide reviews on how useful the recommendations were to his learning needs. The system is supposed to make use of the reviews provided by the learners so as to improve the quality of recommendations that it makes to learners in the future

5.2 System Configuration

After the system have been developed the following needs to be performed

- Installation of Apache tomcat web server
- Installation of a support web browser

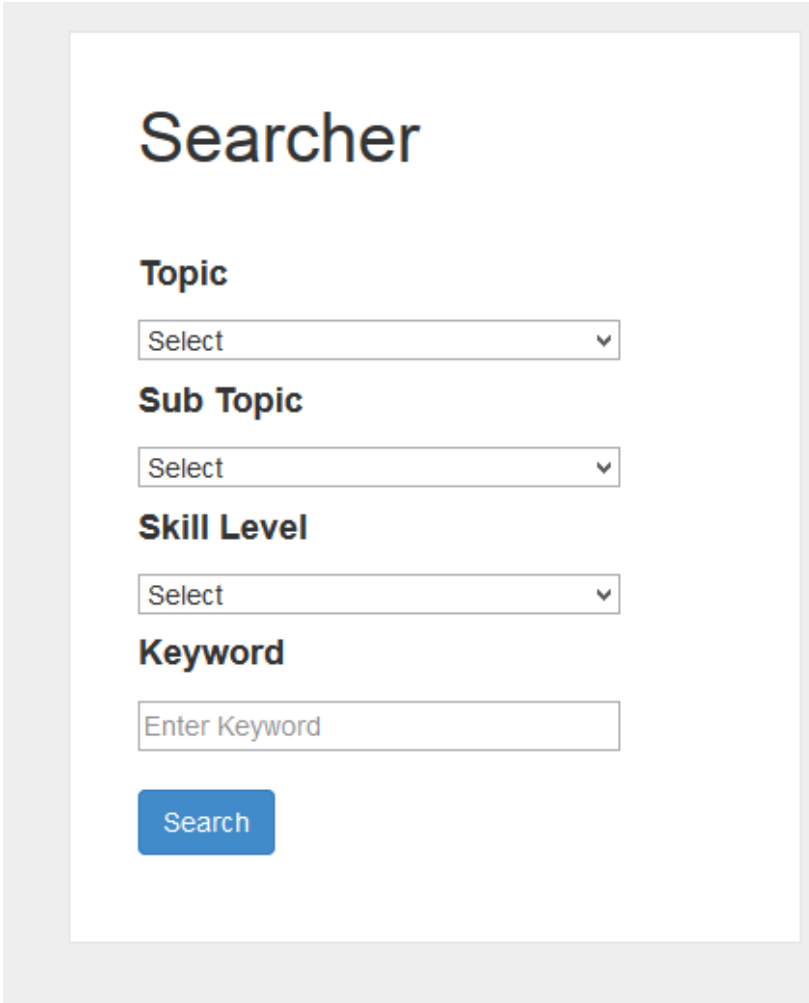
- Internet connection should be available for both the Server and the Client

5.2.1 System experimentation/interaction

The system have been setup as Client–server Architecture where the Sever is running as a Vmware Virtual machine and the client is any machine with a web browser and has a connection to both internet and the application server.

5.2.1.1 Gathering learner’s Needs

This is done by the Profile module which prompts the learners for their topic of interest, the subtopic of interest and any keywords that they might want included in the search query.



The image shows a web interface titled "Searcher". It contains four input fields and a search button. The first three are dropdown menus labeled "Topic", "Sub Topic", and "Skill Level", each with "Select" and a downward arrow. The fourth is a text input field labeled "Keyword" with the placeholder text "Enter Keyword". Below the fields is a blue button with the text "Search".

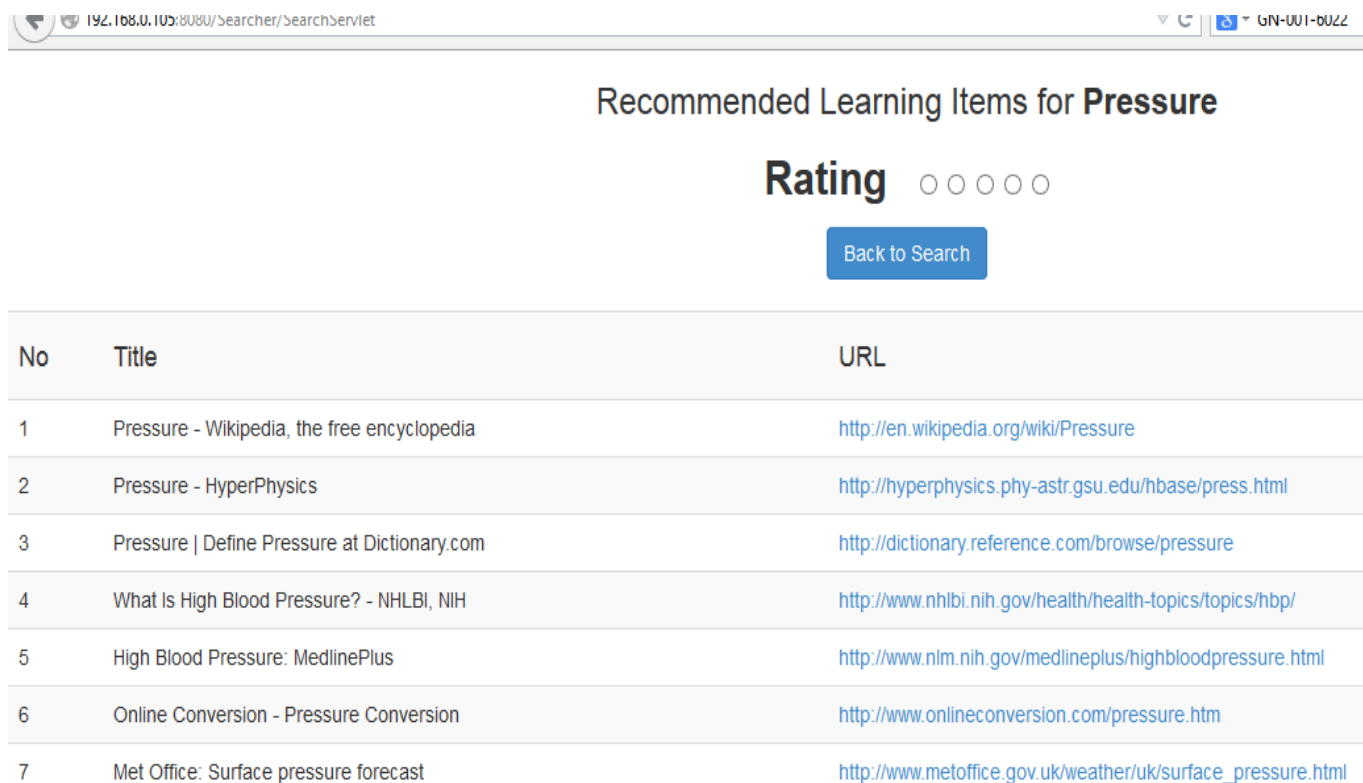
Figure 13 : the Profile module

5.2.1.2 The web search module

This is a backed module that takes the user profile and searches the web. This module connects to APIs for several search engines and uses them to perform the web query.

5.2.1.3 Making recommendations to the learner

This is done by the recommendation module and it's the one which provides the recommended URLs to the learner



The screenshot shows a web browser window with the address bar displaying '192.168.0.105:8080/Searcher/SearchServlet'. The page title is 'Recommended Learning Items for Pressure'. Below the title is a 'Rating' section with five empty circles and a 'Back to Search' button. The main content is a table with three columns: 'No', 'Title', and 'URL'. The table lists seven recommended items, each with a number, a title, and a corresponding URL.

No	Title	URL
1	Pressure - Wikipedia, the free encyclopedia	http://en.wikipedia.org/wiki/Pressure
2	Pressure - HyperPhysics	http://hyperphysics.phy-astr.gsu.edu/hbase/press.html
3	Pressure Define Pressure at Dictionary.com	http://dictionary.reference.com/browse/pressure
4	What Is High Blood Pressure? - NHLBI, NIH	http://www.nhlbi.nih.gov/health/health-topics/topics/hbp/
5	High Blood Pressure: MedlinePlus	http://www.nlm.nih.gov/medlineplus/highbloodpressure.html
6	Online Conversion - Pressure Conversion	http://www.onlineconversion.com/pressure.htm
7	Met Office: Surface pressure forecast	http://www.metoffice.gov.uk/weather/uk/surface_pressure.html

Figure 14: Recommendations Modules

5.2.1.4 The review module

The is module that requires the learner to provide feedback on how relevant the recommendations were

Recommended Learning Items for **Pressure**

Rating ○○○○○

[Back to](#) Please select one of these options.

No	Title	URL
1	Pressure - Wikipedia, the free encyclopedia	http://en.wikipedia.org/wiki/Pressure
2	Pressure - HyperPhysics	http://hyperphysics.phy-astr.gsu.edu/hbase/press.html
3	Pressure Define Pressure at Dictionary.com	http://dictionary.reference.com/browse/pressure
4	What Is High Blood Pressure? - NHLBI, NIH	http://www.nhlbi.nih.gov/health/health-topics/topics/hbp/
5	High Blood Pressure: MedlinePlus	http://www.nlm.nih.gov/medlineplus/highbloodpressure.html
6	Online Conversion - Pressure Conversion	http://www.onlineconversion.com/pressure.htm
7	Met Office: Surface pressure forecast	http://www.metoffice.gov.uk/weather/uk/surface_pressure.html

Figure 15 : Rating module

Chapter 6: RESULTS

6.1.1 Overview

This chapter presents the results of the data analysis. A survey was done to learners who evaluated the system and the survey involved the respondents answering eight questions, of which seven were multi-choice question and the eighth was an open question requiring a general response. The survey contained seven statements in which the learners stated how much they agree with the recommendation made by the system in a scale of 0 to 4.

‘0’ represented “Not Applicable”, ‘1’ represented “Not useful” whereas ‘4’ represented “very useful”.

In addition the system has an online review module where the learners can provide feedback on how useful the recommendations received were to their learning needs.

6.1.2 Results and discussion

We reviewed survey responses from 32 participants, and the following are frequencies of the responses from the participants. Below are the seven statements and the frequency of each score for the 32 respondents

		User review Frequency for each review Question				
		Not Applicable	Not Useful	A bit useful	Useful	Very Useful
Review Questions	How do you Rate The overall recommendations made by the system	0	2	4	25	1
	How easy to use is the system	2	1	5	12	12
	How easy to use is the system	1	3	7	10	11
	Would you recommend the system to other learners	2	3	5	10	12
	How relevant were the recommendation made for Mathematics by the system	0	2	9	13	8

How relevant were the recommendation made for Physics by the system	1	5	3	16	7
How relevant were the recommendation made for Geography by the system	2	3	7	17	3

The first four questions were intended to capture an overall perception of the system by the learners. This is presented graphically in the figure below. And from the data collected the number of Learners who found the prototype as “Useful” And “Very useful” out ways those that found the system as “Not useful” and this shows that the learners have a positive perception about the prototype and if the prototype is developed into a full system they are likely to use it as an online learning tool for learning items recommendations.

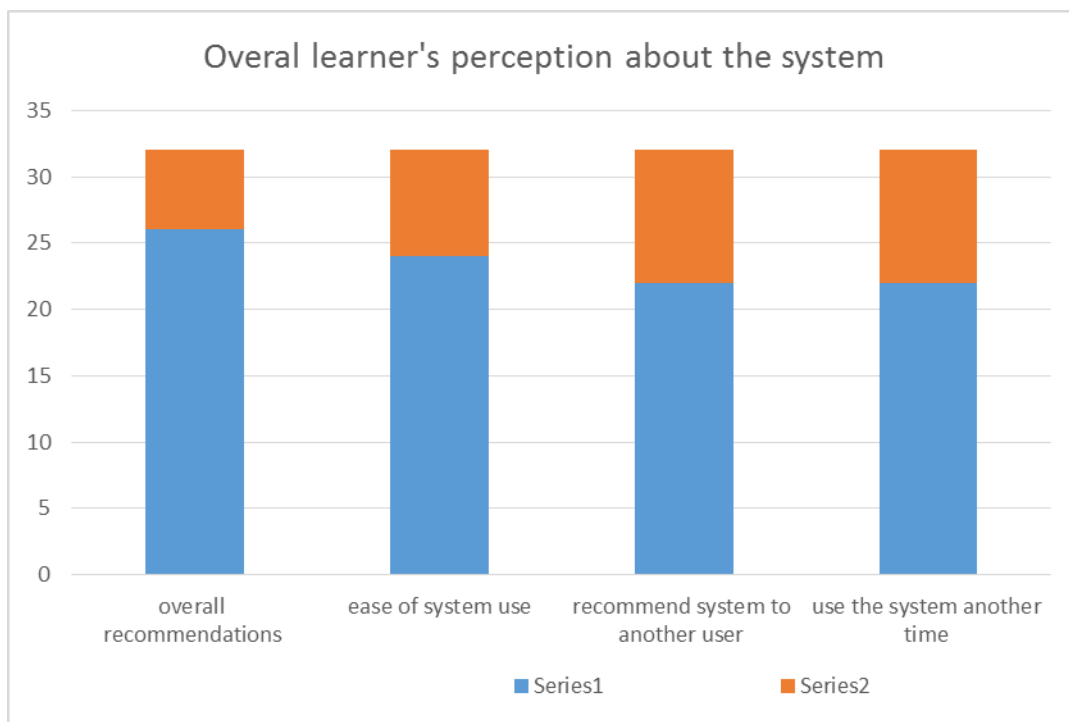


Figure 16: User General Perception about the system

Series 1 represents the learners who rated the system as either “Useful” or “Very Useful” and series2 represents learners who rated the system otherwise.

The next three questions were intended to measure how useful the recommendations were to the learners' needs and from the graphical representation it shows most learners found the recommendation for various subjects relevant to their learning needs which is an impression the prototype was useful to them and chances are they use the full system once fully implemented.

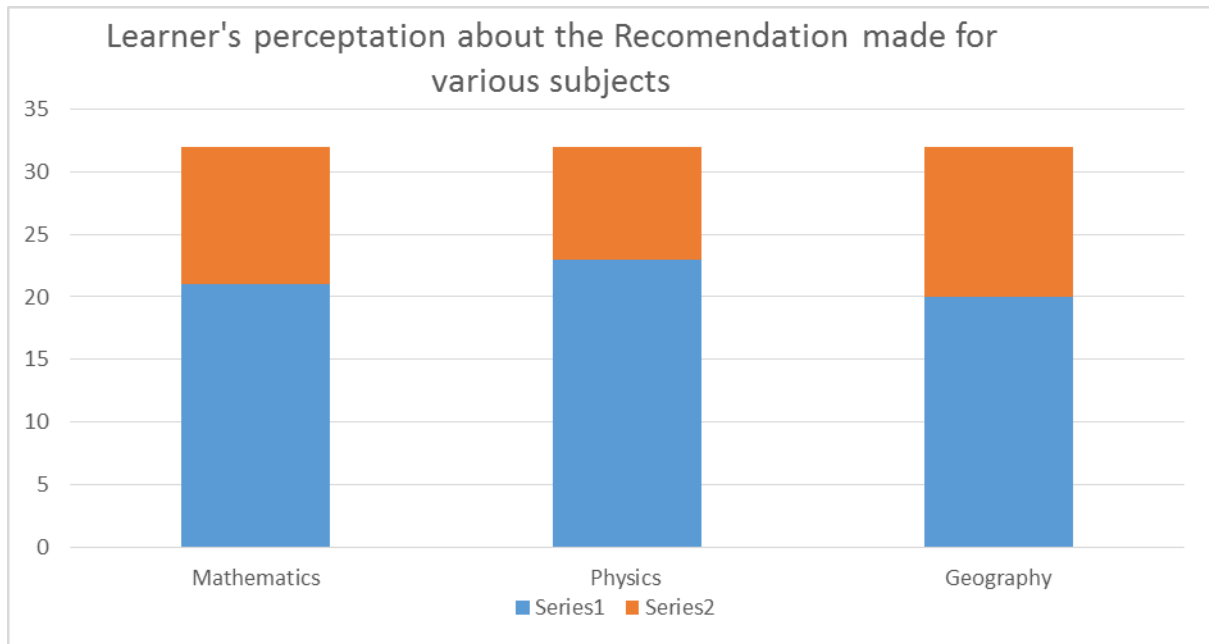


Figure 17: Learner's perception about individual subject recommendation

Series 1 represents the learners who rated the system as either "Useful" or "Very Useful" and series2 represents learners who rated the system otherwise

In addition the system has an online feedback option for learners who opt to provide online reviews for the recommendations provided by the system. Below is a graphical representation of the feedback provided by the 34 learners who opted to provide a review on how useful the recommendations received from the system were, to their learning needs.

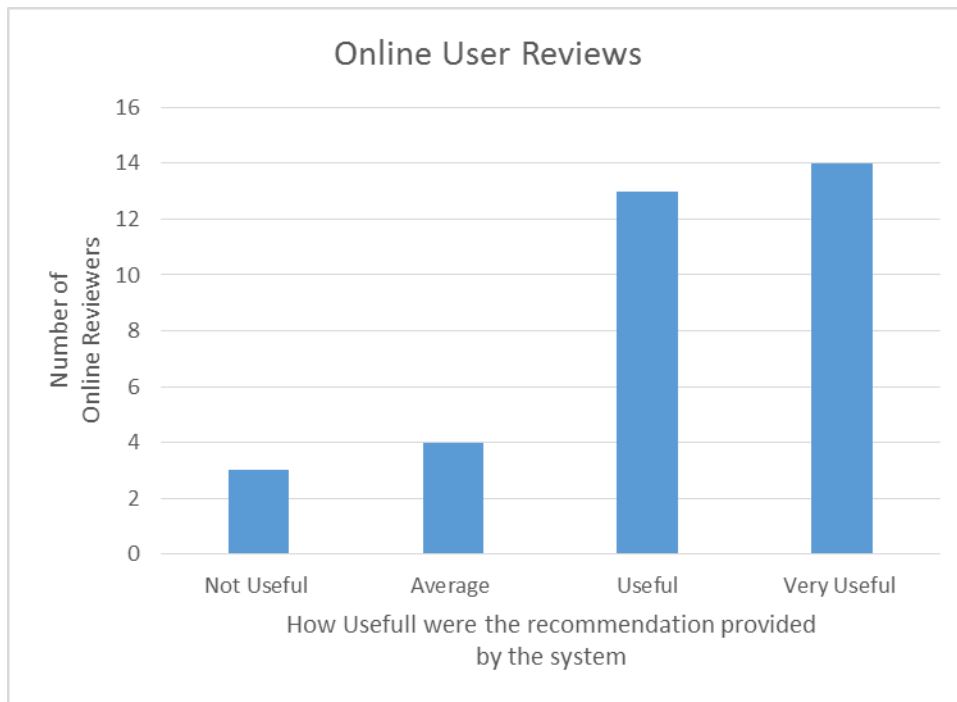


Figure 18: online Use Reviews

The last question was an open ended question which required the participants to state their general perception about the system and whether they think that the system will promote Technology Enhance Learning in Kenya. Out of the 15 respondents who provided reviews for the other questions only 7 of them opted to respond to this question and the below are their responses.

1. The system will help promote online learning amongst high school students.
2. They system will be a good tool to use when searching for learning content on the internet.
3. The system will help popularise local education content that has been published online by point the system users to this local educational content available online.
4. The number of subject and the subtopic covered by the system needs to be increased so that the system can claim to represent the Kenyan learner's learning needs.
5. I like the prototype, when will the full system be implemented?
6. The system is good, but can it be modified to work offline when there is no internet?
7. This is a good prototype and needs to be fully developed into a production system.

Five out of the seven respondent provided a positive review of the system and this is may be an indication of their intent to use the system once it fully deployed.

6.1.3 Summary

The study reveals that reaction by the users towards the system is good, with about 75% of the reviewed participants responding positively. This means that the expected positive impact of the system is high and measures needs to be put in place so as to maintain this positive impact and minimize the small negative impact that has been raised by some of the reviewers.

Chapter 7: CONCLUSION AND RECOMMENDATIONS

7.1.1 Conclusion

In this study we have reviewed how web personalization can be achieved for the Kenyan High school student who is looking for learning resources online and how we can leverage the power of recommenders systems to personalize the way these students surf the web in search of learning materials so as to reduce information overload that is common when searching for content in the world wide web , in addition to reducing the skills required when formulating a search string and lastly checking the learners from “getting Lost” in the hyperspace that is the internet. From the research we have established that the expected user response to the system is good and that it will have positive impact on the system users once fully implemented.

7.1.2 Limitations of the study

The prototype developed is limited to just three subjects under the Kenya High schools syllabus and just a few subtopics under each subject. When the system is fully developed it will be expected to cover all the subjects and subtopics in the Syllabus for High school students.

In addition the prototype used only the freely available Google API to search the web but when the system is fully implemented it will be required to leverage the power of other Search engine APIs so as take advantage of the varying web search algorithms employed by these diverse search engines and as a result get the finest learning resources available online that best fits the learner’s profile.

The learner profile created by the system is based purely on the information provided by the learner on the initial form and it does not take into consideration previous learning needs or traits of the learner neither does it capture the learner’s click stream nor leverage interests of other similar learners.

7.1.3 Further work

In order to fully represent the learning needs of the students and create a complete profile of the learner, more works needs to be done when creating the learner's profile putting into consideration some prior knowledge about the learner's learning needs and learning style when creating the learners profile. Also, once the learner's profile has been created the system needs to be improved so that it can update the created profile based on the learner click stream data and the reviews that the users provide after using the system.

In addition the number of the subjects and topics covered will need to be increased to cover the whole scope that is in the Kenya High School Syllabus.

7.1.4 Recommendations

To achieve the objectives identified in section 7.1.3 above a hybrid recommender system that leverages the strengths of both content based and collaborative based approach will prove to be of more value as it will be able to cope with the evolving learner's needs and as a result it will provide a means of constantly updating the learner's profile which is used in performing the web search and also calculating the similarity.

REFERENCES

1. Aggarwal, c.c., wolf, j.l. , wu, k.l. and yu, p.s. (1999) *Horting hatches an egg: A new graph-theoretic approach to collaborative filtering*. Proceedings of ACM KDD-99, international conference on knowledge discovery and data mining.
2. Agrawal, R., Imielinski, T., and Swami, A. (1993) *Mining association rules between sets of items in large databases*. In Proceedings of SIGMOD'93, Washington, D.C.
3. Alexander, F., Gerhard, F., Dietmar, J. and Markus, Z. (2011) *Developing Constraint-based Recommenders*. New York:Springer.
4. Bhargava, H.K. , Sridhar, S. , Herrick, C., (1999) *Beyond spreadsheets: tools for building decision support systems*, Computer, vol.32.
5. Burke, R. (2002) *Interactive Critiquing for Catalog Navigation in E-Commerce* *Artificial Intelligence Review* 18. California State University: ACM Press
6. Blochl, M., Rumershofer, H. and Wob, W. (2003) *Individualized e-learning systems enabled by a semantically determined adaptation of learning fragments*. In Proceeding of the 14th international workshop on Database and Expert Systems Applications.
7. Chih-Ming, C. , Hahn-Ming, L. , Ya-Hui, C. (2004) *Personalized e-learning system using Item Response Theory*. Taiwan: Elsevier
8. Francesco, R. , Lior, R. , Bracha, S. ,Paul, B. K. (2011) *Recommender Systems Handbook*. New York:Springer.
9. Jie, L.(2004) *A Personalized e-Learning Material Recommender System*. In Proceedings of the 2nd International Conference on Information Technology for Application.
10. Khribi, M. K., Jemni, M., & Nasraoui, O. (2009). *Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval*. International Forum of Educational Technology & Society (IFETS).
11. Mei-Hua, H. (2008) *A personalized English learning recommender system for ESL students: Expert Systems with Applications*. Department of General Education, Chang Gung Institute of Technology, 261, Wen-Hwa 1st Road, Kwei-Shan, Taoyuan, Taiwan, ROC
12. Mooney, R. J., & Roy, L. (2000) *Content-based book recommending Using*

- learning for text categorization*. In Proceedings of digital libraries. San Antonio, TX.
13. Mulvenna, M., Anand, S.S., Buchner, A.G. (2000) *Personalization on the net using web mining*. Communication of ACM 43.
 14. Pao-Hua, C. and Menq-Jiun, W. (2009) *Accessing e-Learners' Knowledge for Personalization in e-learning Environment*. Department of Mechatronics Engineering, National Changhua University of Education No. 2, Shi-da Road, Changhua City, 500, Taiwan, R.O.C.
 15. Pasquale, L., Marco de Gemmis and Giovanni S. (2011) *Content-based Recommender Systems: State of the Art and Trends*. Department of Computer Science, University of Bari "Aldo Moro", Via E. Orabona, 4, Bari (Italy).
 16. Peter, B., Alfred, K., Wolfgang, N. (2007) *The Adaptive Web*. Springer Berlin Heidelberg New York.
 17. Peter, D., Nicola, H., Wolfgang, N., Michael, S. (2004) *Personalization in Distributed e-Learning Environments*. New York, New York, USA.
 18. Rafaeli S., Barak M., Dan-Gur Y. & Toch E. (2004) *QSIA – a Web-based environment for learning, assessing and knowledge sharing in communities*. *Computers & Education*. Graduate School of Business, University of Haifa Mt. Carmel, Haifa, Israel.
 19. Recker M. & Walker A. (2003) *Supporting 'word-of-mouth' Social networks via collaborative information filtering*. Journal of Interactive Learning Research.
 20. Salton, G., McGill, M. (1983) *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY.
 21. Sara jot, S. A. and Bamshad, M. (2005) *Intelligent Techniques for Web Personalization*.
 22. Schafer, J.B., Konstan, J.A., Riedl, J. (2001) *E-commerce recommendation applications, Data Mining and Knowledge Discovery*. GroupLens Research Project, Department of Computer Science and Engineering, University of Minnesota, Minneapolis.
 23. Silvia, S., Patricio, G., Analia, A. (2008) *eTeacher: Providing personalized assistance to e-learning students*. Research Institute – Fac. Cs. Exactas - UNCPBA, Campus Universitario, Paraje Arroyo Seco, 7000 Tandil, Buenos Aires, Argentina.
 24. Sinha, R., Swearingen, K. (June 2001) *Comparing recommendations made by*

- online systems and friends*. In Proceedings of Delos-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries.
25. Tang, T. Y., and McCalla, G.I. (2005) *Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment*. International Journal on E-Learning.
26. Xiao, B., Benbasat, I. (2007) *E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact*. MIS Quarterly, Vol 31

APPENDIX

Appendix 1: Questionnaire

What are the common/preferred learning style among the students?

- Visual
- Aural
- Read / write
- Kinesthetic

Students prefer a teacher or a presenter who uses:

- Demonstrations , models or practical sessions.
- Question and answer, talk, group discussion, or guest speakers.
- Diagrams, charts or graphs.
- Hand-outs, books, or readings.

When your students want to learn a new program, skill or game on a computer. They would:

- Use the controls or keyboard.
- Read the written instructions that came with the program.
- Talk with people who know about the program.
- Follow the diagrams in the book that came with it.

Students like websites that have:

- Interesting written descriptions, lists and explanations.
- Audio channels where they can hear music, radio programs or interviews.
- Things they can click on, shift or try.
- Interesting design and visual features.

What are your major source of teaching materials?

What are the major topics in Mathematics covered by a form one student

What are the major topics in Mathematics covered by a form two student

What topics are considered easy by the students?

What topics are considered challenging by the students?

Appendix 2: Sample Code

The user profile module

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<title>Login</title>
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"
    type="text/css" />
<link href="bootstrap/css/login.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <div class="wrapper">
        <form class="form-signin" action="LoginServlet" method="post">
            <h2 class="form-signin-heading">Please login</h2>
            <input type="text" class="form-control" name="username"
                placeholder="Username" required autofocus /> <br /> <input
                type="password" class="form-control" name="password"
                placeholder="Password" required /> <label class="checkbox">
                <input type="checkbox" value="remember-me"
                id="rememberMe"
                    name="rememberMe"> Remember me
            </label>
            <button class="btn btn-lg btn-primary btn-block"
                type="submit">Login</button>
        </form>
    </div>
```

```
</body>
</html>
```

The search module

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"% >
<!DOCTYPE html>
<html>
<head>
<title>Searcher</title>
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"
    type="text/css" />
<link href="bootstrap/css/login.css" rel="stylesheet" type="text/css" />
<script type="text/javascript">
    function populateDropDown() {

        /* Clear the entire subTopic drop down*/
        document.getElementById("subTopicList").options.length = 0;

        /* adding a Default "Select" option in Topics Menu*/
        var optn = document.createElement("OPTION");
        optn.text = "Select";
        subTopicList.options.add(optn);

        /* This can be changed according to the rest of the code instead of hard coded
values*/
        var selectedIndexVal = document.getElementById("topicList").selectedIndex;
        var subjectValList =
document.getElementById("topicList").options[selectedIndexVal].value;
        var subjectValArray = subjectValList.split(",");
        for (var j = 0; j < subjectValArray.length; j++) {
            var optn = document.createElement("OPTION");
            optn.text = subjectValArray[j];
            optn.value = subjectValArray[j];
            subTopicList.options.add(optn);
        }

    }
</script>
<script type="text/javascript">
    function checkform() {
        if (document.searchForm.subTopicList.value == "Select") {
            return false;
        }
    }
</script>
```

```

<script type="text/javascript">
    function openChart() {
        window.location = '/Searcher/barChart.jsp';
    }
</script>

</head>
<body>
    <div class="wrapper">
        <form class="form-signin" action="SearchServlet" method="post"
            name="searchForm">
            <h1 align="left">Searcher</h1>
            <br />

            <div class="control-group">
                <h4>
                    <label class="control-label">Topic</label>
                </h4>
                <div class="controls">
                    <select id="topicList" name="topic" style="width:
250px" required
                    onchange="javascript:populateDropDown(this);">
                        <option value="">Select</option>
                        <option
                            value="Volumes and Surface
Area,Solution of a general Triangle,Algebra,Statistics, Calculus,CO-ORDINATES AND
GRAPHS">Maths</option>
                        <option
                            value="Measurements,Pressure,Hooke's law,Waves,Fluid Flow">Physics</option>
                        <option
                            value="Intoduction to geography,Earth
and Solar system,Maps and map work">Geography</option>
                    </select>
                </div>
            </div>
            <div class="control-group">
                <h4>
                    <label class="control-label">Sub Topic</label>
                </h4>
                <div class="controls">
                    <select id="subTopicList" name="subTopic"
                    style="width: 250px"
                    required><option value="">Select</option>
                    </select>
                </div>
            </div>
        </div>
    </div>

```

```

<h4>
  <label class="control-label">Skill Level</label>
</h4>
<div class="controls">
  <select name="skill" style="width: 250px" required>
    <option value="">Select</option>
    <option value="Novice">Novice</option>
    <option
value="Intermediary">Intermediary</option>
    <option value="Advanced">Advanced</option>
  </select>
</div>
</div>
<div class="control-group">
  <h4>
    <label class="control-label">Keyword</label>
  </h4>
  <div class="controls">
    <input style="width: 250px" type="text"
list="suggestions"
name="keyword" placeholder="Enter
Keyword">
    <datalist id="suggestions">
      <option value="Volume of a cone">
      <option value="Volume of a cylinder">
      <option value="Volume of a sphere">
      <option value="Volume of a pipe">
      <option value="Sine Rule">
      <option value="Cosine Rule">
      <option value="Area of a triangle">
      <option value="Quadratic Equations">
      <option value="Arithmetic Progression">
      <option value="Geometric progression">
      <option value="Mean">
      <option value="Variance">
      <option value="Standard deviation">
      <option value="Trapezoidal rule">
      <option value="Differentiation">
      <option value="Integration">
      <option value="Cartesian plane">
      <option value="Cartesian co-ordinates">
      <option value="Points on the Cartesian plane">
      <option value="Choice of appropriate scale">
      <option value="Table linear relation">
      <option value="Line graphs">
      <option value="Si units and symbols">
      <option value="Measuring instruments">
      <option value="Density">
      <option value="Volume">
      <option value="Area">

```

```

        <option value="spring constant">
        <option value="spring balance">
        <option value="hooke's law">
        <option value="Amplitude">
        <option value="wavelength">
        <option value="frequency">
        <option value="periodic time">
        <option value="bernoulli's effect">
    </datalist>
</div>
</div>

<br />
<div class="btn-group">
    <button type="submit" class="btn btn-primary"
        onsubmit="return checkform()">Search</button>
    <button type="button" class="btn btn-primary"
onclick="openChart()">Feedback
        Trend</button>
</div>

</form>
</div>
</body>
</html>

```

Recommendation

```
package com.mylearner.recommender;
```

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.mylearner</groupId>
    <artifactId>recommender</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>recommender</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

```

```

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.mahout</groupId>
    <artifactId>mahout-core</artifactId>
    <version>0.9</version>
  </dependency>
</dependencies>
</project>

```

```

import java.io.File;
import java.util.List;
import org.apache.mahout.cf.taste.impl.model.file.FileDataModel;
import org.apache.mahout.cf.taste.impl.neighborhood.NearestNUserNeighborhood;
import org.apache.mahout.cf.taste.impl.recommender.GenericUserBasedRecommender;
import org.apache.mahout.cf.taste.impl.similarity.PearsonCorrelationSimilarity;
import org.apache.mahout.cf.taste.model.DataModel;
import org.apache.mahout.cf.taste.neighborhood.UserNeighborhood;
import org.apache.mahout.cf.taste.recommender.RecommendedItem;
import org.apache.mahout.cf.taste.recommender.Recommender;
import org.apache.mahout.cf.taste.similarity.UserSimilarity;

public class RecommenderIntro {
    public static void main(String[] args) throws Exception {

        DataModel model = new FileDataModel(new File("input.csv"));

        UserSimilarity similarity = new PearsonCorrelationSimilarity(model);
        UserNeighborhood neighborhood = new NearestNUserNeighborhood(2, similarity, model);
        Recommender recommender = new GenericUserBasedRecommender(model, neighborhood,
        similarity);

        List<RecommendedItem> recommendations = recommender.recommend(1, 1);

        for (RecommendedItem recommendation : recommendations) {
            System.out.println(" recommendation: " + recommendation);
        }
    }
}

```

Return form

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"% >
<% @ page import="java.util.*"% >

```

```

<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ page import="javax.servlet.http.*,javax.servlet.*"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>

<html>
<head>
<title>Results</title>

<link rel=stylesheet href="bootstrap/css/bootstrap.css" />
<script type="text/javascript">
    function giveFeedback() {
        window.location = '/Searcher/ratings.jsp';
    }
</script>
<script type="text/javascript">
    function checkForm() {
        window.location = '/Searcher/search.jsp';
    }
</script>
</head>
<body>
    <form action="">
        <h3 align="center">
            <font> Google Search Results for <b> ${keyword}</b></font>
        </h3>

        <DIV ALIGN="center">
            <div class="btn-group">
                <button type="submit" class="btn btn-primary"
onclick="checkform()">Back
                    to Search</button>
                <button type="button" class="btn btn-primary"
onclick="giveFeedback()">Give Feedback</button>
            </div>

        </DIV>

        <br />

        <table class="table table-striped">
            <tr>
                <td align="left"><h4>No</h4></td>
                <td align="left"><h4>Title</h4></td>
                <td align="left"><h4>URL</h4></td>
            </tr>
            <%
                <u>Iterator itr;</u>
            %>
            <%

```



```

List data = (List) request.getAttribute("results");
for (itr = data.iterator(); itr.hasNext();) {
%>
<tr>
<td><%=itr.next()%></td>
<td><%=itr.next()%></td>
<%
String s = itr.next().toString();
%>
<td><a href=<%=s%>><%=s%></a></td>
</tr>
<%
}
%>
</table>
</form>
</body>
</html>

```

Appendix 3: Data collected during survey

		User review Frequency for each review Question					
		0	1	2	3	4	LEGEND
Review Questions	How do you Rate The overall recommendations made by the system	0	2	4	25	1	0=Not Sure
	How easy to use is the system	2	1	5	12	12	1= Not useful
	How easy to use is the system	1	3	7	10	11	2= A bit Useful
	Would you recommend the system to other learners	2	3	5	10	12	3= Useful
	How relevant were the recommendation made for Mathematics by the system	0	2	9	13	8	4= very Useful
	How relevant were the recommendation made for Physics by the system	1	5	3	16	7	
	How relevant were the recommendation made for Geography by the system	2	3	7	17	3	

Figure 19: Survey data

Appendix 4: how to run the system

For the system to run, the following prerequisites should be met

- At least 4 GB of Memory

- XAMMP software
- Java Eclipse
- MYSQL

To run the system follow the below steps

- Install Xampp on the server and start MYSQL and Apache services
- Run Java Eclipse and import the Searcher.war file into the project space
- Run the login.java applet as a server
- Open any browser from any machine and access <http://serverip/searcher.jsp>
- Login using your name and continue to search the web in a personalized way