UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

A COURSE RECOMMENDER SYSTEM AT HIGHER EDUCATION LEARNING

Kegunya Robert Meng'anyi

P52/33063/2019

Supervisor

Dr. Lawrence. Muchemi

A Project Proposal Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computational Intelligence, School of Computing and Informatics, University of Nairobi.

August , 2021

DECLARATION

This research project report is my original work and has not been presented for any award in any other University.

____

KEGUNYA ROBERT MENG'ANYI **DATE** 28/08/2021

P52/33063/2011

This research project report has been submitted in partial fulfillment of the requirements of the Masters of Science in Computational Intelligence of The University of Nairobi with my approval as the University Supervisor.

____

DR. LAWRENCE. MUCHEMI **DATE** 1st Sept 2021

School of Computing and Informatics

University of Nairobi

DEDICATION

This research project is dedicated to my wife, daughter Zuri, and parents for their unwavering support during my Master’s degree programme at The University of Nairobi.

AKNOWLEDGEMENT

I wish to express my appreciation to Dr. Lawrence Muchemi, who dedicated time as my supervisor in providing all the support and guidance. Many thanks are due also to other members of the panel and review committee Dr. Samwel Ruhiu and Prof. Waiganjo Wagacha who provided the technical guidance that I so much needed though this research. My thanks and gratitude is due also to my family for their encouragement and patience without which this work would not have been possible. My gratitude also goes to my colleague at work Mr. Peter Ndegwa for their encouragement and insights during this project and, University of Nairobi classmates; Mr. Samwel Ndiva, Mr. Kennedy Njoroge, and Mr. Paul Wanyanga for dedicating their time to share knowledge through lengthy discussions with respect to this project. Above all I thank God the Almighty for granting me good health and the spirit to work on this project.

ABSTRACT

Recommender systems have been employed in entertainment, e-commerce, agriculture, healthcare and education among other industries to provide personalised suggestions to users. Recommender systems help to solve the problem a user being overburdened with information when using online systems. Due to digitization of the course application process in the institutions of higher learning, courses are now made available online in portals for students to apply. These courses are too many for the student to do adequate research before selection. This leads to students being selected to courses that they are not interested in and thus the need for a course recommendation system that suggests a short list of courses that are relevant to the student. This study focussed on developing a knowledge base recommender system prototype for providing personalised course recommendations to students based on their interests and performance. Knowledge based system development life cycle was used to develop the prototype and knowledge acquisition was done from domain experts and documented materials. To identify the interests, a questionnaire is administered. The Hollands three letter Code is then used to identify the personality. The personalities and results are then used to suggest a short list of courses that are relevant to the student. The model developed had an accuracy of 85.12% and thus can be used to recommend courses to students.

TABLE OF CONTENTS

List of Figures

List of Tables

1.0 INTRODUCTION

1.1 Background

Recommender systems have been used to provide suggestions that are a user's preferences (Ricci, et al., 2015). With the availability of additional information to a user, it poses a challenge to them to go through all the information to identify what is relevant to them. A recommender system aims to provide a solution to the problem of a user being overburdened by information by ensuring the user experience is personalised and accurate personalised recommendations of items are delivered to the users of a system. Recommendation systems predict whether an item would be useful to a user based on information made available (Fayyaz, et al., 2020). Recommendation systems have been used in industries such as entertainment, e-commerce, healthcare, agriculture and education among others.

Students need guidance when selecting their course choices for higher education. In Kenya, students lack well planned and organised career guidance in schools (Ndung'u & Obae, 2020). According to (Crocker, 2002), the well-being of modern society is dependent not only on traditional capital and labour but also on the knowledge and ideas possessed and generated by individual workers. Therefore, education is the primary source of human capital with 90% of persons with higher education being likely to be employed than those with no formal education (Mulongo, 2013). In this regard, the government provides sponsorship through Kenya Universities and Colleges Central Placement Service (KUCCPS) that coordinates placement of government sponsored students and Universities Funding Board (UFB) which pays part of the fees required to be paid by the student to the universities (Universities Act, 2012). As recommended by (Mulongo, 2013), this enhances access to higher education and ensures that students who are victims of regional disparities caused by high cost of higher education coupled with remoteness and underdeveloped infrastructure enjoy equity.

With the introduction of the Government of Kenya's a hundred percent transition policy, there has been a tremendous increase in enrolments across all public secondary schools in Kenya. This translates to an increase in competition for the available capacities for the existing courses. A significant number of students who secure admission into the universities through KUCCPS are neither offered degree courses of their choice nor placed in their preferred university (Ndung'u & Obae, 2020).This is majorly because students missing placement due to capacity being met in all of the course choices. The students are shared to other courses that have capacities not met and are similar to the courses they applied. Further, (Lugulu & Kipkoech,

2011) found out that 63.3% of students admitted in public universities were dissatisfied with the degree courses because they were placed in degree courses, they did not choose nor had a passion for. The choice of degree course made when joining universities is one of the series of decisions made in the process of career development and is a major turning point in the students' lives which not only is a start to workplace readiness, but also establishes the student in a career path that opens as well as closes opportunities (Gacohi, 2017). (Nyamwange, 2016) recommends that students should be encouraged to make career choice decisions in areas they have or can acquire knowledge easily, skills and have interest as it is likely to promote productivity when the student is doing what they are interested in.

Career information is the provision of accurate and usable facts concerning university courses (Gibson & Mitchell, 2003). KUCCPS has developed a career book that provides insights into career opportunities, progression pathways, subject requirements for specific careers, and government-sponsored student placement processes. Students are given university course cut-offs over the previous years as insights to guide them during university course selection and to predict their placement to a university course.

An optimal course recommender system that uses student interests and performance to suggest a list of courses that match their interests and performance will ensure that students are selected to courses that they have interest and can acquire knowledge easily.

1.2 Problem statement

The main problem to be addressed by this research is that of students getting selected to courses that they have no interests, have difficulty in acquiring knowledge and information overload on students during application for admission to higher learning education institutions. Students having many choices to choose from and perform research on in order to identify a list of courses that they have interests makes it difficult for the students to make informed decisions. Information overload is caused by availability of many institutions offering many courses to be chosen from by a student. Students end up missing placement to the courses they selected due to lack of adequate guidance when selecting their course choices during an application period.

Therefore, there is a need to provide a solution that recommends a filtered list of courses to students based on their interests and performance.

1.3 Objectives

The main objective of this research is to develop a model for course recommendation that suggests a list of courses to a student based on their interests and performance.

Specific objectives:

1. To investigate how to incorporate student interests and performance to recommend courses.
2. To investigate which type of recommender system can be used to recommend courses to students based on their interests and performance.
3. To design a model for course recommendation to students.
4. To evaluate the model for course recommendation using a prototype.

1.4 Significance of study

The number of students enrolling into secondary schools has continued to rise since the introduction of the Free Primary Education in Kenya in 2003 leading to an increased demand to higher education in the country due to the realization that higher education forms the principal pillar of education. Regardless of the measures put to increase selection of students to higher education institutions, many of the students are still not guaranteed selection to courses that they have interest in. A course recommender system that will make use of the student performance and student interests will reduce the level of information overload and uncertainty by students during application for selection. This will lead to an increasing number of students getting selected to courses that they have interest in.

1.5 Stakeholders

The stakeholders that will be affected by the study will be as follows:

1. Students that will be making applications for higher education learning will have filtered and relevant information to work with as opposed to all information including what they would not choose.
2. Labour market. Students graduating will be passionate about their careers thus an increase in productivity.

1.6 Scope of Study

This paper will focus on developing and evaluating a course recommender system prototype that recommends courses to students based on student interests and their performance.

2.0 LITERATURE REVIEW

2.1 Introduction

This section will discuss recommender systems and the various literature of previous work done in recommending courses.

2.2 Course Selection

Course selection is an important step as it serves the purpose of furthering the education of the student (Hussin & Muhamad, 2019; Gacohi, 2017). This stage involves decision making to select a course from a list of available courses in a career of interest. Career choice is the outcome of a career assessment that involves evaluating a student's academic potential, interest, personality, values, expectations and available resources (Gacohi, 2017).

Student selection to higher education institutions in Kenya is based on a set criterion by the individual institutions and the professional regulatory bodies. The requirements for admission may vary from high school subject grades, high school aggregate grades, first degree and or entry examinations administered by the institutions (Gacohi, 2017). Professionals and parents of the students influence the choice of course a student makes (Gacohi, 2017; Mberia & Midigo, 2018). The process of course selection poses a dilemma to students as they face a difficulty in matching their career choices with their abilities, interests and academic performance (Mberia & Midigo, 2018).

2.2.1 Student Interests

Holland's Theory of Vocational Personalities in Work Environment popularly known as the Holland's Code was developed by John Holland and provides a framework on understanding career interests that are thereafter used in career guidance. According to Holland, vocational interests can be used to express one's personality and these interests can be conceptualised into six personalities including Realistic (R), Investigative (I), Artistic (A), Social (S), Enterprising (E), and Conventional (C). The degree of resemblance to the personality types is then assessed and a three-letter code is generated to summarise the career interest of a person. The first letter in the three-letter code would then be the person's dominant personality and would play a major role in the choice of career and satisfaction in the career. The other two letter would likely play a lesser role in career choice however, they are still significant in career selection (Leung, 2008).

Figure 1: Holland Code Six Personality Types

2.2.2 Student Selection

Selection of government sponsored students is coordinated by KUCCPS (Republic of Kenya, 2012). KUCCPS board has developed policies to guide on the process of selecting students into available courses. Eligibility of students to be placed to a course is they must have sat for the Kenya Certificate of Secondary Education (KCSE), must be a Kenyan citizen, and must meet all the admission requirements for the course approved by the regulating body as displayed in Figure 12. The admission requirements are a list of subjects taken by the student in KCSE and grades attained by the students in another set of subjects as a form of student assessment. The students are students are then selected on merit to one of the courses applied until the capacity of the course is reached. The courses are group into clusters and sub clusters. Courses with similar subject requirements are put together into one cluster and are further grouped into regulated programmes with same subject grade requirements (The Kenya Universities and Colleges Central Placement Service, 2014).

2.3 Recommender systems

Recommender systems are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user. The suggestions relate to various decision-making processes like courses to select, music to listen to, movies to watch and products to purchase (Ricci, et al., 2015).

2.3.1 Types of recommender systems

2.3.1.1 Content Based Recommender Systems

Content base recommender systems uses both the item properties and user preferences to the item while building the recommendations (Pettersen & Tvete, 2016).

Building a content-based recommender system involves recommending items similar to those that have been preferred by the user in the past. These systems are scalable, work well independent of the number of users in the system and does not experience cold start issues since it takes consideration of historical preferences of a user and property of an item. However, these systems need enough details about the item to be provided so as to differentiate products precisely without which poor accuracy is experienced (Fayyaz, et al., 2020).

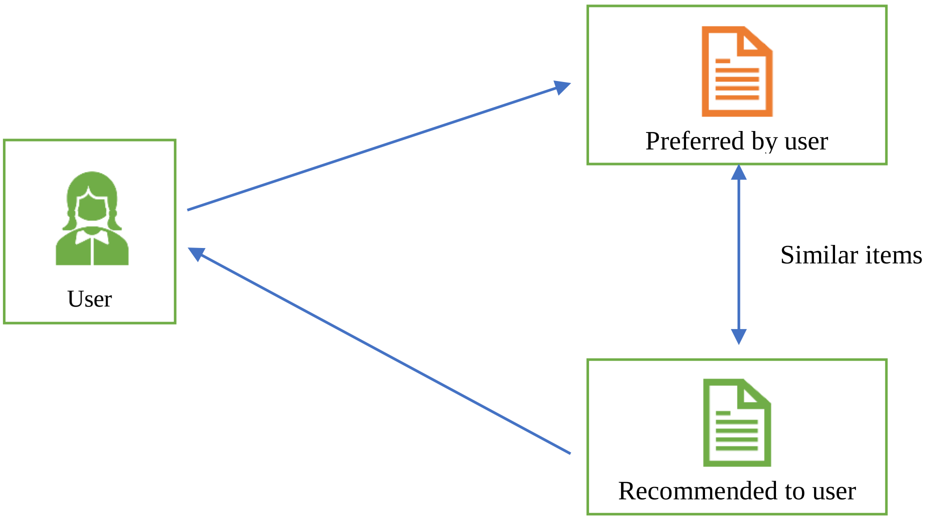

Figure 2: Content Based Recommender System

2.3.1.2 Collaborative Filtering

Collaborative filtering recommender systems recommend items based on popularity by other users (Ricci, et al., 2015). A collaborative filtering recommender system will predict user interest in new items based on recommendations from other people with similar interests. If two users shared similar interests in the past, then they will likely have similar tastes in future (Fauzan, et al., 2020).

Collaborative filtering boasts being simple to implement and accurate. However, they suffer from cold start problem where they fail to first time users whose information is not on the system (Gorakala, 2016).

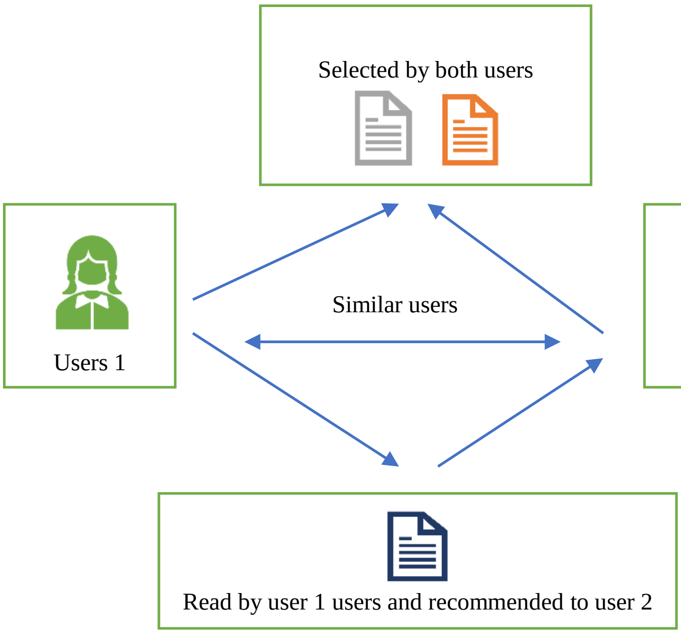

Figure 3:Collaborative filtering Recommender Systems

2.3.1.3 Knowledge based recommender systems

Knowledge based recommender systems recommends items based on logical inferences about user preferences. These systems are useful where measures of preferences are not available for the recommendation process (Aggarwal, 2016).

2.3.1.4 Hybrid recommender systems

Hybrid recommender systems combine more than one types of recommender system and ensure that they complement each other by replacing weaknesses of one type with strengths of the other type (Gorakala, 2016). This approach increases the efficiency of recommendations and compared to individual recommendation techniques.

2.4 Related work

(Bhumichitr, et al., 2017) proposed a recommendation system that recommends university elective courses based on similarities between the courses and courses taken by the student.

Collaborative based recommendation using Pearson Correlation Coefficient and Alternating Least Square (ALS) were subjected on dataset of academic records of university students. ALS was found to be the best performed and deployed in the recommender system. The researcher further proposes the use of other information apart from the student enrolment data in order to incorporate the behaviour of the student for further recommendation.

(Shankarmani, et al., 2020) in a bid to solve the problem of students having large numbers of courses to select from proposed a recommender system that maps students to courses based on their understanding of the various job domains. C-means fuzzy clustering was used to cluster all students with the same understanding of the job domains as the target student. The next step was to choose a limited number of the clusters by eliminating clusters where courses that were taken by unlike student are removed to ensure no points had fewer similarities chosen. The courses taken by majority of the students similar to the target student is then found by calculating the weighted mode values using formular:

$$wm_i = \sum_{j=1}^{n} f_j * \mu_j$$

where *wmi* is weighted mode of course *i* having frequency *fj* in cluster *j* and membership coefficient *μj*. This solution however, recommends only one programme to the student and the courses that belonged to the same domain were wrongly predicted. The research notes the significance of recommending more than one courses to the student.
(Mondal, et al., 2020) proposed a recommendation system that used a machine learning approach to suggest appropriate courses to learners based on their past learning details and past performance. K-Means clustering algorithm was used to classify learners based on the performance details and thereafter, collaboration filter techniques used on the clusters to get suitable courses for the student. The student will then be tested based on the recommended courses. The researcher proposes adding a knowledge base in future to find similarities so that other related students' area of interest and target needs are recognized.

Knowledge based systems have been used to recommend courses to students based on rule-based engines. (Omulo & Kemboi, 2013) proposed a knowledge-based system prototype to recommend courses to students based on their performance in Kenya Certificate of Secondary Education. The prototype uses subject grades and minimum requirements of the course to determine the suitability of a course to the student. (Muchemi & Winston, 2008) proposed an expert system prototype that stored highly specialised knowledge of career counselling that is acquired from career counsellor. The research aimed at administering career guidance to students at an efficient, quality and affordable way. The expert system was composed of three modules including personality analysis based on the Myers-Briggs Topology Indicator model that classifies human nature into personality, outlook, temperament and lifestyle and recommends job groups to the student based on their personality. A decision-making module gives the student an opportunity to narrow down to a small set of careers. Lastly, the college entrance expert system module further filtered the careers of interest based on the criteria for enrolment into courses in a higher learning institution. Both knowledge-based systems recommended a list of courses to the student and their performance was satisfactory (Omulo & Kemboi, 2013; Muchemi & Winston, 2008).

2.5 Gap

Recommender systems have been applied in various domains as a means to avoid information overload among users of online systems. Course recommendation is an issue that research is continuously being conducted (Bhumichitr, et al., 2017). Prototypes have been developed to recommend courses using knowledge-based systems, collaborative filtering and clustering techniques. Student performance, previous enrolment data and personality types based on the Myers Brigg model that focusses on perception have been used to recommend courses to students. Myers briggs is popularly used for job transitioning based on the perception as opposed to Holland Code model that is used to categorise jobs based on student interests (Keach, 2020) .However, there is a need to use interest of the students in recommending courses. The literature reviewed brings out the need to incorporate the interest of the students and their performance in developing course recommendations (Mondal, et al., 2020).

2.6 Conceptual Model

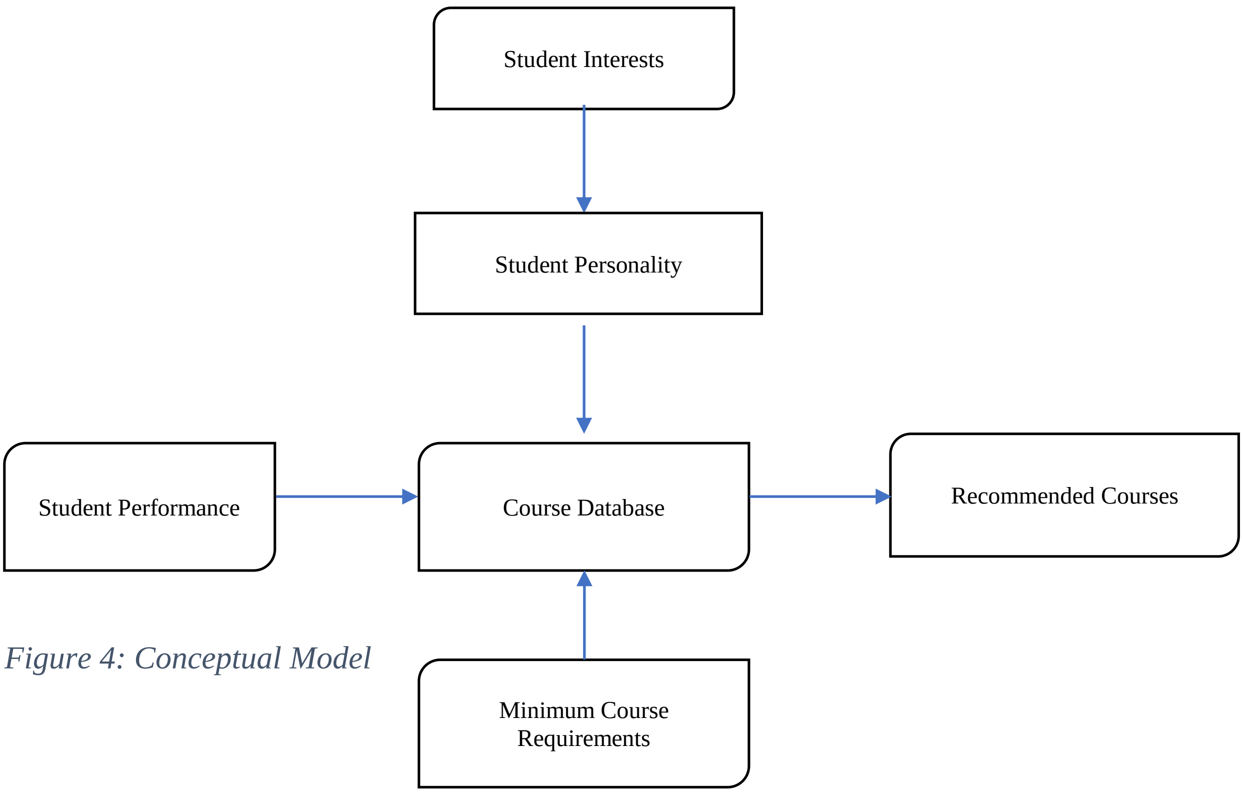

Figure 4: Conceptual Model

3 METHODOLOGY

This paper intended to develop and evaluate a recommender system that will be able to suggest relevant courses to students based on their interests and performance. Course data was collected from KUCCPS, interest questions database was collected and occupation and their three letter personalities were collected from the O*NET database. The courses collected were mapped to the three letter personality types where they correspond to the occupations. The mapping was done with the help of a domain expert from KUCCPS.

3.1 Research Framework

Knowledge Based System Development Methodology was used to guide the steps for developing the course recommender system prototype during the study.

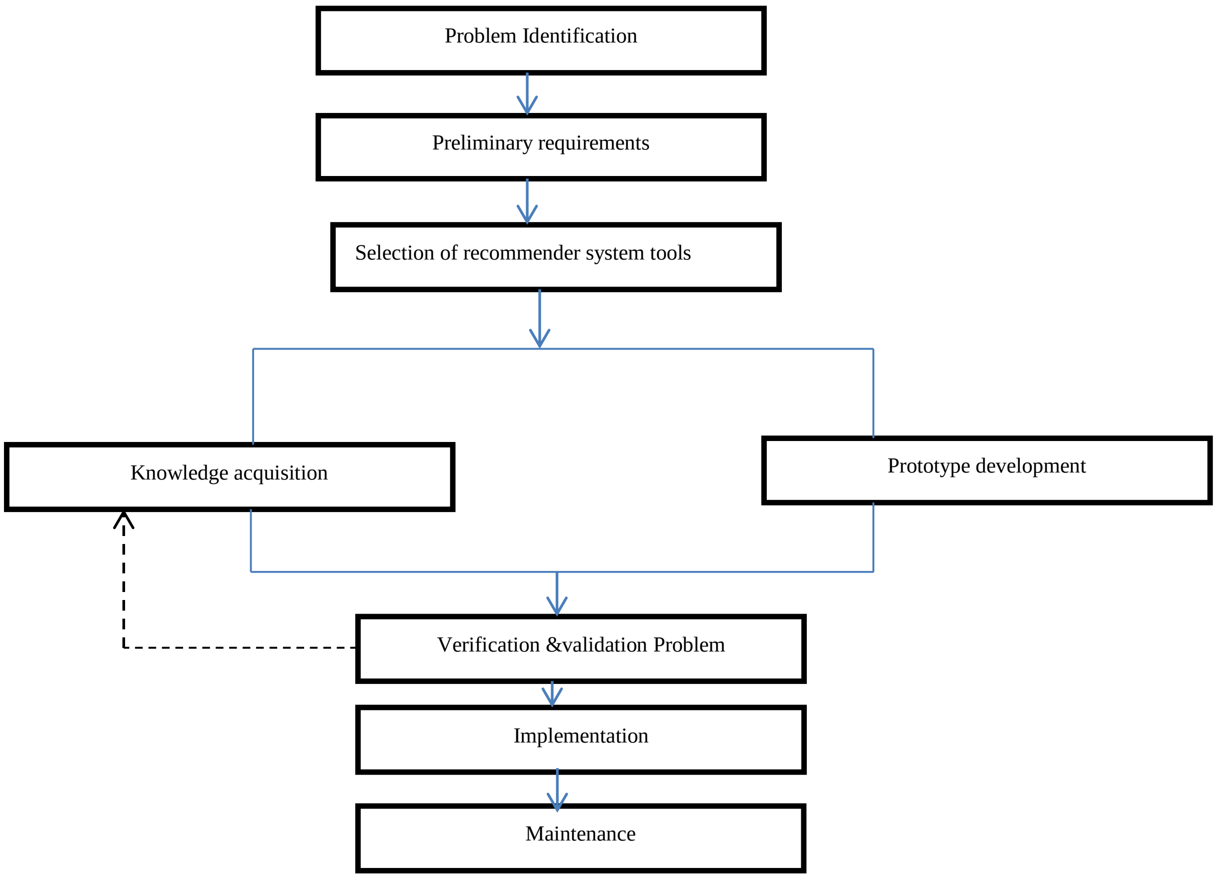

Figure 5: Knowledge Base System Development Life Cycle

- **Problem Identification**

This stage involved the identification of the problem to be solved by the knowledge base system.

- **Preliminary Requirements Analysis and Knowledge Acquisition**

This stage involved the acquisition of the initial requirement to set stage for the design of the prototype. This stage is a preliminary of the knowledge acquisition stage.

- **Selection of Recommender System Tools**

This stage involved the selection of the various tools to be used in developing the prototype. SWI Prolog was used in development. CSV files were used to store structured data which are then loaded to the knowledge base at the start-up of the system. CSV files were used to store course details, course cluster requirements, course sub cluster requirements, subject details, grade details, interests and their questions. Command Line Interface (CLI) was used as the user interface with clear instruction on the usability of the system given.

- **Knowledge Acquisition and Prototype Development**

This stage involved performing two activities in a continual process until the desired result is achieved. Knowledge acquired from the domain expert was converted into frame-based knowledge representation forms for knowledge that is object based and is structured and rule-based knowledge representation form to captures what to do at different stages of recommending courses. Knowledge acquired was continually updated to the knowledge base until the prototype worked as expected.

- **Verification And Validation**

This stage involved verifying that the knowledge acquired is valid and not ambiguous to ensure it does not violate the expert's expectation. This originates from the expert not identifying all the implicit dependencies. Further, reasoning was verified to eliminate invalid reasoning that may have arose from incorrect translation of the knowledge.

Validation was done on the prototype to ensure that hidden errors are captured and rectified before proceeding to release the prototype. This stage was done by a separate domain expert from the expert used during the development of the prototype to eliminate any biasness.

The knowledge acquisition and prototyping stage was invoked when an issue arose at this stage.

- **Implementation**

Due to the dynamic rules and procedures in the application domain the prototype is subject to regular change thus maintenance is needed which ensures that all actors periodically review the performance of the system.

3.2 Research methods

3.2.1 Population and Sampling Technique

The study focused on recommendation of courses to students based on the student's interest and their performance. The researcher selected KUCCPS and professionals and domain experts within KUCCPS in the study due to its mandate of coordinating placement of government sponsored students to colleges and universities and developing career guidance programmes for the benefit of the students.

Purposive sampling was used to select participants using a preselected criteria relevant to the research questions. Purposive sampling was used to select the domain experts for knowledge acquisition and the participants for the evaluation of the prototype. Two career development officers were selected based on their experience and profession. Ten professionals drawn from various professions were selected to evaluate the prototype. The professionals were subjected to visual interaction with the system and interviews were done to collect their responses.

3.2.2 Data collection

Course data on the description, minimum and regulatory requirements and qualification level was collected from the KUCCPS student's portal. This data contained details of the courses that have been declared by universities and colleges and are available for application by the students. Domain knowledge was acquired from the domain experts through structured and unstructured interviews.

Occupation and their three letter Holland Code personality data were collected from the O*NET database of occupations.

Interests questions database was obtained from Columbia City High School. The document lists questions that are used to perform personality analysis using the Holland's Code model.

Moreover, other sources of knowledge were from published policies and procedures obtained from the KUCCPS website.

3.2.3 Data preparation

Available courses collected from the KUCCPS students' portal were mapped to the courses that lead to the occupations. The courses were then mapped to the personality types that best fit the courses based on the interests of the student. The mapping of the occupations to courses and thereafter to the three letter holland code personality was done with the help and guidance of a Career Guidance Professional at KUCCPS.

3.2.4 Prototype Development

The prototype was developed in a continual process of knowledge acquisition and updating the knowledge base with new domain knowledge and inference knowledge obtained from the domain experts and available.

SWI Prolog was used as a tool to develop the prototype. SWI Prolog was preferred since it is popular and there are freely available packages that makes the process of programming fast. In addition, there was availability of free and easily understood documentation and community of developers.

3.2.5 Evaluation

Evaluation was undertaken to verify the effectiveness of the prototype in recommending courses to students based on their interests and performance at the secondary school. Precision, true positivity rate and false positivity rate were used to measure the performance of the prototype in recommending courses to secondary school graduate student.

3.2.6 Ethical Considerations

Confidentiality of the respondents' data was ensured by not storing their personal data during evaluation of the prototype. This is in line with the Data Protection Act of 2019 that stipulates regulations on the collection, storage and processing of data.

Anonymity was considered when collecting evaluation data by ensuring that the responses are not traceable to a participant.

4 ANALYSIS, DESIGN AND IMPLEMENTATION

This chapter outlines the process of analysis, design and implementation of the course recommender system prototype.

4.1 Design Model

The design model shows the system is constructed and describes the critical components of the prototype. The model shows the relationship between the components. The course recommender system is developed on SWI Prolog which contains the components of a knowledge-based system including a user interface, an inference engine and knowledge base. Users query the system through a Command Line Interface to complete the personality test and submit their performance results then request for recommendation of courses based on their interests and performance. The recommended courses are return and displayed through the command line interface.

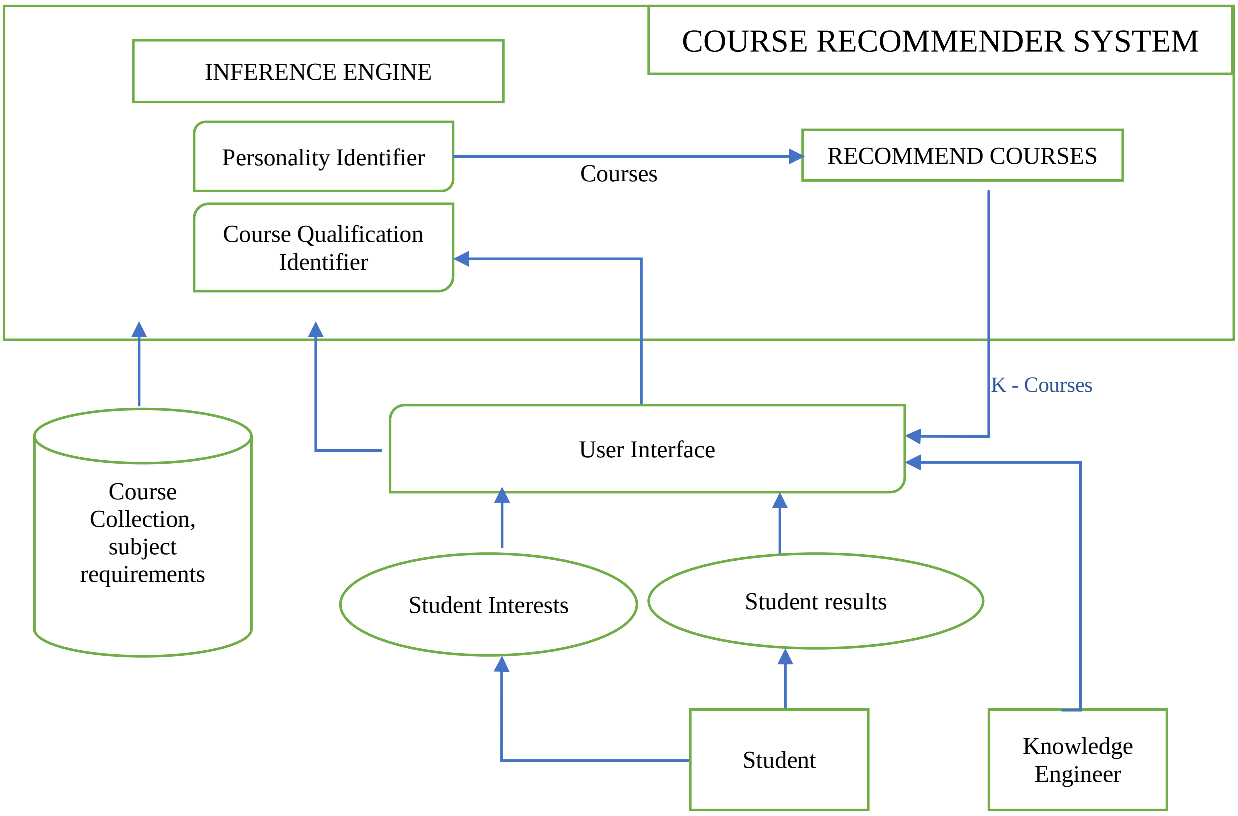

Figure 6 : Architecture of the Course Recommender Prototype

4.2 Personality Identifier

The user is subjected to a personality test with sixty questions. Each question is mapped to a personality type. Each positive response to a question is weighted as 1 and negative response is weighted as a 0. The weights of each personality are determined by aggregating the responses. The next step is to get the top three personality types with the highest weights and that is what is to represent the student personality.

```
admin@LM003 PROJECT % swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('PROJECT.pl').
true.

?- import_db.
true.

?- execute_questionnaire.
Kindly answer all the questions: (1 means Yes, and 0 means No.)
Please Enter your answer followed by a fullstop.
1. I consider myself to be athletic
|: 1.

2. I am a nature lover
|: 0.

3. I am curious about the physical world(nature, space and living things)
|: 1.

4. I am independent
|: 1.

5. I like to fix things
|: 1.

6. I like to use my hands(plant a garden, help with fixing the house)
|: 1.

7. I enjoy exercising
|: 1.

8. I like to save money
|: 1.

9. I like to work until the work gets done
|: 1.

10. I like working on my own
|: 1.

11. I am very cautious and careful
|: 1.

12. I am curious about everything
|: 1.

13. I can do complex calculations
|: 1.

14. I like to solve maths problems
|: 1.

15. I like to use computers
|: 1.

16. I like to read book all the time
```

Figure 7: How personality test is administered

All the questionnaire answers are stored in the memory and flushed out once the user session is closed.

questionnaire(Questions):- findall(D, questions(D,_,_), Questions).
administer_questionnare([]).
administer_questionnare([H|T]):-
write(H), nl, read(A), nl,questions(H,Y,_),assertz(answer(Y,A)), administer_questionnare(T).

```
execute_questionnaire :-
  write('Kindly answer all the questions: (1 means Yes, and 0 means No.)'), nl ,
  write('Please Enter your answer followed by a fullstop.'),nl,
  questionnaire (Questions),
  administer_questionnare (Questions),!.
```

Figure 8: Code Extract that Administers Personality Test

```
[?- print_answers.
R-1
R-0
R-1
R-1
R-1
R-1
R-1
R-1
R-1
R-1
I-1
I-1
I-1
I-1
I-1
I-1
I-0
I-0
I-1
I-1
A-1
A-0
A-0
A-0
A-0
A-0
A-0
A-0
A-0
A-0
S-0
S-1
S-0
S-1
S-0
S-1
S-0
S-0
S-0
S-0
E-1
E-0
E-0
E-0
E-1
E-0
E-0
E-1
E-1
E-1
C-0
C-0
C-0
C-0
C-1
C-1
C-0
C-0
C-1
C-0
true.
```

Figure 9: Answers to an administered personality test

```
print_answers :-
   findall(Y-Z, answer(Y,Z), Answers), printlist(Answers).
```

Figure 10: Code extract that prints all answers to an administered personality test.

Figure 11 shows the extract that is used to calculate the three-letter personality code for the student.

```
[[debug]  ?- get_three_letter_personality_code.
 [E,I,R]
 true.
```

```
get_three_letter_personality_code:-
   findall(E, answer('E',E), Es), findall(I, answer('I',I), Is),
   findall(A, answer('A',A), As),findall(S, answer('S',S), Ss), findall(C, answer('C',C), Cs), findall(R, answer('R',R), Rs), list_sum(Es, ET), list_sum(As, AT), list_sum(Is, IT), list_sum(Cs, CT), list_sum(Ss, ST),list_sum(Rs, RT), retractall(weight(_,_)),
assertz(weight(ET,'E')),assertz(weight(AT,'A')),assertz(weight(IT,'I')),assertz(weight(ST,'S')),assertz(weight(CT,'C')),assertz(weight(RT,'R')),
   sort_personality_by_weight(T), write(T).
```

Figure 11: Extract of code that calculates the three letter personality code and result.

4.3 Course Qualification Identifier

This module checks verifies whether the student has met all the minimum subject requirements of the course that belongs to a sub cluster of a cluster. Clusters are a group of courses with similar subject requirements. A cluster always has four subject requirements that a student has to have results. A subject requirement can only be represented by one subject.

A sub cluster is a group of courses within the same cluster and have similar minimum subject grade requirements. The sub cluster requirements mostly are requirements by a regulatory body of a profession which the student must meet.

BACHELOR OF SCIENCE (MECHANICAL ENGINEERING)

Cluster 7 - Engineering, Technology & Related | View Available Institutions

MINIMUM ENTRY REQUIREMENTS

CLUSTER SUBJECT 1	MAT A
CLUSTER SUBJECT 2	PHY
CLUSTER SUBJECT 3	CHE
CLUSTER SUBJECT 4	BIO / HAG / GEO / CRE / IRE / HRE / HSC / ARD / AGR / WW / MW / BC / PM / ECT / DRD / AVT / CMP / FRE / GER / ARB / KSL / MUC / BST
NOTE: A subject may only be considered **ONCE** in this section	

MINIMUM SUBJECT REQUIREMENTS

SUBJECT 1	MAT A	C+
SUBJECT 2	PHY	C+
SUBJECT 3	CHE	C+
SUBJECT 4	ENG / KIS	C+

Figure 12: Bachelor of Science Mechanical Engineering minimum requirements. Obtained from KUCCPS students Portal (2021)

```
print_all_subjects :-
    findall(X-Y-Z,subject(X,Y,Z),Subjects), printlist(Subjects).
```

```
?- consult('PROJECT.pl').
true.

?- print_all_subjects.
101-ENGLISH-GROUP I
102-KISWAHILI-GROUP I
121-MATHEMATICS OPTION A-GROUP I
122-MATHEMATICS OPTION B-GROUP I
231-BIOLOGY-GROUP II
232-PHYSICS-GROUP II
233-CHEMISTRY-GROUP II
236-BIOLOGY-GROUP II
237-GENERAL SCIENCE-GROUP II
311-HISTORY & GOVERNMENT-GROUP III
312-GEOGRAPHY-GROUP III
313-CHRISTIAN RELIGIOUS EDUCATION-GROUP III
314-ISLAMIC RELIGIOUS EDUCATION-GROUP III
315-HINDU RELIGIOUS EDUCATION-GROUP III
441-HOMESCIENCE-GROUP IV
442-ART & DESIGN-GROUP IV
443-AGRICULTURE-GROUP IV
444-WOODWORK-GROUP IV
445-METALWORK-GROUP IV
446-BUILDING CONSTRUCTION-GROUP IV
447-POWER MECHANICS-GROUP IV
448-ELECTRICITY-GROUP IV
449-DRAWING AND DESIGN-GROUP IV
450-AVIATION TECHNOLOGY-GROUP IV
451-COMPUTER STUDIES-GROUP IV
501-FRENCH-GROUP V
502-GERMAN-GROUP V
503-ARABIC-GROUP V
504-KENYAN SIGN -GROUP V
511-MUSIC-GROUP V
565-BUSINESS STUDIES-GROUP V
234-PHYSICAL SCIENCE-NULL
235-BIOLOGICAL SCIENCES-NULL
316-SOCIAL EDUCATION AND ETHICS-NULL
427-ECONOMICS-NULL
561-ACCOUNTING-NULL
562-COMMERCE-NULL
true.
```

Figure 13: Extract of code that prints all available subjects from the knowledge base

```
result(101,'B+').
result(102,'B+').
result(121,'B+').
result(231,'D+').
result(232,'B+').
result(233,'B+').
result(312,'B+').
result(501,'B+').
```

Figure 14: Extract of sample student results

```
[?- check_cluster(cluster_7).
[true .

[?- trace.
 true.

[[trace]  ?- check_cluster(cluster_7).
[    Call: (10) check_cluster(cluster_7) ? creep
[    Call: (11) cluster(cluster_7, _9870, cluster_subject_1, _9874) ? creep
[    Exit: (11) cluster(cluster_7, 'Cluster 7 - Engineering, Technology & Related', cluster_subject_1, 121) ? creep
[    Call: (11) cluster(cluster_7, _9958, cluster_subject_2, _9962) ? creep
[    Exit: (11) cluster(cluster_7, 'Cluster 7 - Engineering, Technology & Related', cluster_subject_2, 232) ? creep
[    Call: (11) cluster(cluster_7, _10046, cluster_subject_3, _10050) ? creep
[    Exit: (11) cluster(cluster_7, 'Cluster 7 - Engineering, Technology & Related', cluster_subject_3, 233) ? creep
[    Call: (11) cluster(cluster_7, _10134, cluster_subject_4, _10138) ? creep
[    Exit: (11) cluster(cluster_7, 'Cluster 7 - Engineering, Technology & Related', cluster_subject_4, 231) ? creep
[    Call: (11) all_different([121, 232, 233, 231]) ? creep
[    Call: (12) lists:member(121, [232, 233, 231]) ? creep
[    Fail: (12) lists:member(121, [232, 233, 231]) ? creep
[    Redo: (11) all_different([121, 232, 233, 231]) ? creep
[    Call: (12) all_different([232, 233, 231]) ? creep
[    Call: (13) lists:member(232, [233, 231]) ? creep
[    Fail: (13) lists:member(232, [233, 231]) ? creep
[    Redo: (12) all_different([232, 233, 231]) ? creep
[    Call: (13) all_different([233, 231]) ? creep
[    Call: (14) lists:member(233, [231]) ? creep
[    Fail: (14) lists:member(233, [231]) ? creep
[    Redo: (13) all_different([233, 231]) ? creep
[    Call: (14) all_different([231]) ? creep
[    Call: (15) lists:member(231, []) ? creep
[    Fail: (15) lists:member(231, []) ? creep
[    Redo: (14) all_different([231]) ? creep
[    Call: (15) all_different([]) ? creep
[    Exit: (15) all_different([]) ? creep
[    Exit: (14) all_different([231]) ? creep
[    Exit: (13) all_different([233, 231]) ? creep
[    Exit: (12) all_different([232, 233, 231]) ? creep
[    Exit: (11) all_different([121, 232, 233, 231]) ? creep
[^   Call: (11) findall(_11160, result(_11160, _11162), _11222) ? creep
[    Call: (16) result(_11160, _11162) ? creep
[    Exit: (16) result(101, 'B+') ? creep
[    Redo: (16) result(_11160, _11162) ? creep
[    Exit: (16) result(102, 'B+') ? creep
[    Redo: (16) result(_11160, _11162) ? creep
[    Exit: (16) result(121, 'B+') ? creep
[    Redo: (16) result(_11160, _11162) ? creep
[    Exit: (16) result(231, 'D+') ? creep
[    Redo: (16) result(_11160, _11162) ? creep
[    Exit: (16) result(232, 'B+') ? creep
[    Redo: (16) result(_11160, _11162) ? creep
[    Exit: (16) result(233, 'B+') ? creep
[    Redo: (16) result(_11160, _11162) ? creep
[    Exit: (16) result(312, 'B+') ? creep
[    Redo: (16) result(_11160, _11162) ? creep
[    Exit: (16) result(501, 'B+') ? creep
[^   Exit: (11) findall(_11160, user:result(_11160, _11162), [101, 102, 121, 231, 232, 233, 312, 501]) ? creep
[    Call: (11) allMembers([121, 232, 233, 231], [101, 102, 121, 231, 232, 233, 312, 501]) ? creep
[    Call: (12) allMembers([232, 233, 231], [101, 102, 121, 231, 232, 233, 312, 501]) ? creep
[    Call: (13) allMembers([233, 231], [101, 102, 121, 231, 232, 233, 312, 501]) ? creep
[    Call: (14) allMembers([231], [101, 102, 121, 231, 232, 233, 312, 501]) ? creep
[    Call: (15) allMembers([], [101, 102, 121, 231, 232, 233, 312, 501]) ? creep
[    Exit: (15) allMembers([], [101, 102, 121, 231, 232, 233, 312, 501]) ? creep
[    Call: (15) isMemberOf(231, [101, 102, 121, 231, 232, 233, 312, 501]) ? creep
```

get_passed_sub_cluster_subject_requirements(Sub_Cluster) :-

(findall(A, (sub_cluster(_,Sub_Cluster, A, Requirement,Subject_Code), result(Subject_Code,Result), grade(Requirement, RQ_AGP), grade(Result, RS_AGP), RS_AGP >= RQ_AGP), All_A), sort(All_A,Distinct_A), findall(B,sub_cluster(_,Sub_Cluster,B,_,_),All_B), sort(All_B,Distinct_B), length(Distinct_A, N), N > 0 ,Distinct_A=Distinct_B,!); sub_cluster(_,Sub_Cluster,null,null,null),!.

Figure 15: Extract of code, results and tracing of the process to check for cluster requirements

```
[[trace]  ?-
[|
[|    notrace.
 true.

[[debug]  ?- get_passed_sub_cluster_subject_requirements('7A').
 true.
```

```
[[trace]  ?- get_passed_sub_cluster_subject_requirements('7A').
[   Call: (10) get_passed_sub_cluster_subject_requirements('7A') ? creep
[^  Call: (11) findall(_5710,  (sub_cluster(_5706, '7A', _5710, _5712, _5714), result(_5714, _5726), grade(_5712, _5738), grade(_5726, _5750), _5750>=_5738), _5816) ? creep
[   Call: (17) sub_cluster(_5706, '7A', _5710, _5712, _5714) ? creep
[   Exit: (17) sub_cluster(cluster_7, '7A', '7A_subject_1', 'C+', 121) ? creep
[   Call: (17) result(121, _5726) ? creep
[   Exit: (17) result(121, 'B+') ? creep
[   Call: (17) grade('C+', _5738) ? creep
[   Exit: (17) grade('C+', 7) ? creep
[   Call: (17) grade('B+', _5750) ? creep
[   Exit: (17) grade('B+', 10) ? creep
[   Call: (17) 10>=7 ? creep
[   Exit: (17) 10>=7 ? creep
[   Redo: (17) sub_cluster(_5706, '7A', _5710, _5712, _5714) ? creep
[   Exit: (17) sub_cluster(cluster_7, '7A', '7A_subject_2', 'C+', 232) ? creep
[   Call: (17) result(232, _5726) ? creep
[   Exit: (17) result(232, 'B+') ? creep
[   Call: (17) grade('C+', _5738) ? creep
[   Exit: (17) grade('C+', 7) ? creep
[   Call: (17) grade('B+', _5750) ? creep
[   Exit: (17) grade('B+', 10) ? creep
[   Call: (17) 10>=7 ? creep
[   Exit: (17) 10>=7 ? creep
[   Redo: (17) sub_cluster(_5706, '7A', _5710, _5712, _5714) ? creep
[   Exit: (17) sub_cluster(cluster_7, '7A', '7A_subject_3', 'C+', 233) ? creep
[   Call: (17) result(233, _5726) ? creep
[   Exit: (17) result(233, 'B+') ? creep
[   Call: (17) grade('C+', _5738) ? creep
[   Exit: (17) grade('C+', 7) ? creep
[   Call: (17) grade('B+', _5750) ? creep
[   Exit: (17) grade('B+', 10) ? creep
[   Call: (17) 10>=7 ? creep
[   Exit: (17) 10>=7 ? creep
[   Redo: (17) sub_cluster(_5706, '7A', _5710, _5712, _5714) ? creep
[   Exit: (17) sub_cluster(cluster_7, '7A', '7A_subject_4', 'C+', 101) ? creep
[   Call: (17) result(101, _5726) ? creep
[   Exit: (17) result(101, 'B+') ? creep
[   Call: (17) grade('C+', _5738) ? creep
[   Exit: (17) grade('C+', 7) ? creep
[   Call: (17) grade('B+', _5750) ? creep
[   Exit: (17) grade('B+', 10) ? creep
[   Call: (17) 10>=7 ? creep
[   Exit: (17) 10>=7 ? creep
[   Redo: (17) sub_cluster(_5706, '7A', _5710, _5712, _5714) ? creep
[   Exit: (17) sub_cluster(cluster_7, '7A', '7A_subject_4', 'C+', 102) ? creep
[   Call: (17) result(102, _5726) ? creep
[   Exit: (17) result(102, 'B+') ? creep
[   Call: (17) grade('C+', _5738) ? creep
[   Exit: (17) grade('C+', 7) ? creep
[   Call: (17) grade('B+', _5750) ? creep
[   Exit: (17) grade('B+', 10) ? creep
[   Call: (17) 10>=7 ? creep
[   Exit: (17) 10>=7 ? creep
[^  Exit: (11) findall(_5710, user:(sub_cluster(_5706, '7A', _5710, _5712, _5714), result(_5714, _5726), grade(_5712, _5738), grade(_5726, _5750), _5750>=_5738), ['7A_subject_1', '7A_subject_2', '7A_subje
t_3', '7A_subject_4', '7A_subject_4']) ? creep
[   Call: (11) sort(['7A_subject_1', '7A_subject_2', '7A_subject_3', '7A_subject_4', '7A_subject_4'], _8166) ? creep
[   Exit: (11) sort(['7A_subject_1', '7A_subject_2', '7A_subject_3', '7A_subject_4', '7A_subject_4'], ['7A_subject_1', '7A_subject_2', '7A_subject_3', '7A_subject_4']) ? creep
[^  Call: (11) findall(_8234, sub_cluster(_8230, '7A', _8234, _8236, _8238), _8298) ? creep
[   Call: (16) sub_cluster(_8230, '7A', _8234, _8236, _8238) ? creep
[   Exit: (16) sub_cluster(cluster_7, '7A', '7A_subject_1', 'C+', 121) ? creep
[   Redo: (16) sub_cluster(_8230, '7A', _8234, _8236, _8238) ? creep
[   Exit: (16) sub_cluster(cluster_7, '7A', '7A_subject_2', 'C+', 232) ? creep
[   Redo: (16) sub_cluster(_8230, '7A', _8234, _8236, _8238) ? creep
```

Figure 16: Sample extract showing the tracing of the process to check for sub cluster requirements

4.4 Course Recommendation

This module is use to filter courses based on inputs received from the qualification identifier and the personality identifier modules. The available courses are reduced to a list of courses that the student matches based on their identified personality and interests and their performance. The student is recommended courses based on the level of qualification they prefer. There are three levels of qualification namely degree, diploma certificate and artisan and their course distribution are as per the Table 1.

Table 1: Table showing the distribution of courses by personality and qualification type.

	QUALIFICATION TYPE				
PERSONALITY	**ARTISAN**	**CERTIFICATE**	**DIPLOMA**	**DEGREE**	**TOTAL**
ACI				2	**2**
AEI		4	10	18	**32**
AER	5	2		2	**9**
AES			6	16	**22**
AIR			1	9	**10**
AIS			17	25	**42**
ARS		1		2	**3**
CEI	1	4	2	17	**24**
CER	6	4	5	15	**30**
CES	2	10	13	39	**64**
CIR	33	18	28	67	**146**
CIS			2	2	**4**
CRS			1	8	**9**
ECS		1		4	**5**
EIR				1	**1**
EIS		1	3	5	**9**
ERS	5	2	3	1	**11**
IAS				5	**5**
ICR				2	**2**
IRS		2	7	16	**25**
SIR			1	4	**5**
TOTAL	**52**	**49**	**99**	**260**	**460**

```
[[debug]  ?- get_course_recommendations.
 Kindly provide your preferred qualification level (degree,diploma,certificate,artisan):
[|: degree
[|: .

 [E,I,R]
 BACHELOR OF SCIENCE (GEOLOGY)
 true.

[[debug]  ?- get_course_recommendations.
 Kindly provide your preferred qualification level (degree,diploma,certificate,artisan):
[|: diploma.

 [E,I,R]
 false.

[[debug]  ?- get_course_recommendations.
 Kindly provide your preferred qualification level (degree,diploma,certificate,artisan):
[|: certificate.

 [E,I,R]
 false.

[[debug]  ?- get_course_recommendations.
 Kindly provide your preferred qualification level (degree,diploma,certificate,artisan):
[|: artisan.

 [E,I,R]
 false.
```

Figure 17: Extract of how the course recommendation is made

5 RESULTS AND DISCUSSION

This chapter presents the results of the evaluation of the prototype undertaken after development and discusses the findings in relation to the research objectives.

5.1 Evaluation of the Prototype

The prototype performance was evaluated by measuring its efficiency and effectiveness of recommending likeable courses to students. The prototype was subjected to evaluation by professionals within KUCCPS from different professions. The professionals were subjected to the visual presentation of the prototype to check whether the recommendations given were likeable to them and if there were any courses that were not recommended and are from the list of courses in the database that match their personality and performance and that they would consider to pursue. A total of 56 courses were recommended in the evaluation. The recommendation was capped to ten randomly courses achieved from shuffling the courses and recommending the top ten.

A confusion matrix of the prototype performance was developed as shown in the Table 2.

	Liked	**Not Liked**
Recommended Courses	51	5
Not Recommended Courses	13	52

Table 2: Confusion matrix of the recommender system prototype

ACCURACY	**PRECISION**	**RECALL**	**SPECIFICITY**	**F-MEASURE**
85.12	91.07	79.69	91.23	85

Table 3: Accuracy of the prototype

As shown in the table, the accuracy of the prototype is 85.12 % which is an acceptable performance. The F-Measure which is a measure of effectiveness is 85% which indicates the prototype has a good performance.

The main challenge that led to the performance observed was due to the method of selecting the top K courses for recommendation.

5.2 Discussion

From the evaluation of the performance of the system, an accuracy of 85.12 % is a good performance considering the method of selection of the top ten courses for recommendation. The accuracy achieved indicates that student interests and their performance can be used to effectively recommend courses to students.

Usage of the prototype was measured through precision, recall and specificity. Precision, recall and specificity measure whether a user considers the use of the recommendations (Gunawardana & Shani, 2015). The precision was 91.07%, recall 79.69% and specificity at 91.23 %.

Knowledge based systems can as well be used to recommend courses to students where course rating data is not available. Knowledge based course recommender system has functional knowledge on how a course meets the needs of a student and is able to reason the relationship between the student's interests and performance and a possible recommendation of courses.

6.0 CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

Recommender systems have been employed in entertainment, e-commerce, agriculture, healthcare and education among other industries to provide personalised suggestions to their users based on available information.

In education, recommender systems have been employed to suggest courses to students based on their performance, personalities and rating of the courses given by other students.

The research focussed on how to incorporated the interests and performance of the students in recommending courses to them.

A knowledge-based recommender system prototype was developed to evaluate a model for course recommendation and its performance was evaluated to measure its efficiency and effectiveness in recommending courses

The recommendation of courses ensure that the interests and ability of the student are used when making recommendation to a student. The prototype is used to recommend a list of courses to the student who then have a reduced number of courses from which they can easily work with by doing further research on them and finally selecting a preferred course from the recommended list. This ensures that students are selected to courses that they have interest in.

6.2 Recommendation

Course recommendation is key in suggesting courses relevant courses to students. This research focussed on recommending courses based on the student interests and ability. A knowledge base recommender system prototype was developed to recommend a list of courses at random to ensure serendipity.

Future studies need to explore ways of introducing ranking of the courses. This will provide a better way of recommending top courses to the student.

6.0 REFERENCES

Aggarwal, C. C., 2016. *Recommender Systems The Textbook.* Yorktown Heights, NY, USA: Springer International Publishing Switzerland.

Bhumichitr, K. et al., 2017. *Recommender Systems for University Elective Course Recommendation.* NakhonSiThammarat, Thailand, IEEE, pp. 1-5.

Crocker, J., 2002. The Costs of Seeking Self–Esteem. *Journal of Social Issues.*

Fauzan, F., Nurjanah, D. & Rismala, R., 2020. Apriori Association Rule for Course Recommender system. *Ind. Journal on Computing.*

Fayyaz, Z. et al., 2020. Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities. *Appl. Sci.*.

Gacohi, J. N., 2017. Selected Factors Influencing Degree Choice And Placement: A Comparative Study Of Regular and Self-Sponsored Students in Kenyan Public Universities, Kenya. *Egerton University.*

Gibson, R. L. & Mitchell, M., 2003. *Introduction to counseling and guidance.* 6th ed ed. Upper Saddle River, N.J: Merrill/Prentice Hall.

Gorakala, S. K., 2016. *Building Recommendation Engines.* Birmingham: Packt Publishing Ltd.

Gunawardana, A. & Shani, G., 2015. *Recommender Systems Handbook.* s.l.:Springer, Boston, MA.

Hussin, L. & Muhamad, N., 2019. Review Paper: Determinants of Students' Choice of Courses and University Selection. *https://www.researchgate.net/publication/338229377.*

Keach, J. A., 2020. *Practical strategies for transitioning. [Presentation]. Amigos Member Conference—Next Steps: Create and Cultivate a Career That Matters, Online.*. s.l.:James Madison University.

Leung, S. A., 2008. *The Big Five Career Theories. In: Athanasou J.A., Van Esbroeck R. (eds) International Handbook of Career Guidance.* s.l.:Springer, Dordrecht.

Lugulu, J. & Kipkoech, L., 2011. The effect of provision of career guidance information in secondary schools on choice of degree programme. *Journal of Emerging Trends in Educational Research and Policy Studie.*

Mberia, A. & Midigo, R., 2018. Understanding Career Choice Dilemma in Kenya: Issues of Informed Choices and Course Availability. *Journal of Education and Practice,* 9(9).

Mondal, B., Patra, O., Mishra, S. & Patra, P., 2020. *A course recommendation system based on grades.* Gunupur, India, IEEE, pp. 1-5.

Muchemi, L. & Winston, O., 2008. Career Guidance Using Expert System Approach. *Strengthening the ICCR: Role of ICT in Development.*

Mulongo, G., 2013. Effect of active learning teaching methodology on learner participation. *Journal of Education and Practice,* 4(4).

Ndung'u, K. J. & Obae, R., 2020. Effects of Career Placement by KUCCPS among the Undergraduate Students in Public Universities in Kiambu County, Kenya. *International Journal of Research and Innovation in Social Science (IJRISS).*

Nyamwange, J., 2016. Influence of Student's Interest on Career Choice among First Year University Students in Public and Private Universities in Kisii County, Kenya. *Journal of Education and Practice,* 7(4).

Omulo, E. T. O. & Kemboi, K. K., 2013. Knowledge-Base Career Choice Assistant.

Pardos, Z. A. & Jiang, W., 2020. *Designing for Serendipity in a University Course Recommendation System.* Frankfurt, Germany, ACM.

Pettersen, M. & Tvete, A. K., 2016. *A Hybrid Recommender System for Context-Aware Recommendations of Restaurants,* s.l.: Norwegian University of Science and Technology.

Republic of Kenya, 2012. *The Universities Act No. 42 of 2012.* Nairobi: Government Printers.

Ricci, F., Rokach, L. & Shapira, B., 2015. *Recommender Systems Handbook.* s.l.:Springer New York Heidelberg Dordrecht London.

Shankarmani, R., Sankhe, V., Shah, J. & Paranjape, T., 2020. Skill Based Course Recommendation System. *IEEE International Conference on Computing, Power and Communication Technologies (GUCON).*

The Kenya Universities and Colleges Central Placem, 2014. *Placement Policy.* s.l.:KUCCPS.

Wirth, R. & Hipp, J., 2000. *CRISP-DM: Towards a Standard Process Model for Data Mining.* s.l., Practical Application of Knowledge Discovery and Data Mining.

Witten, I. H., Frank, E. & Hall, M. .., 2011. *Data Mining: Practical Machine Learning Tools and Techniques.* 3rd Edition ed. Burlington: Morgan Kaufmann Publishers.