NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

THINK IN OTHER TERMS

Faculty of Applied Sciences Department of Computer Science

Bachelor of Science Honours Degree in Computer Science

Automated Curriculum Vitae Screening and Recommendation System

Student Name   : Awakhiwe Khabo
Student Number   : N0163520Y
Project Supervisor   : Mrs S. Moyo

Abstract

Human Resources Management (HRM) is people management within an organization. Recruitment as one of the main functions of HRM is the first step in building an organization's human capital. The goal is to locate and hire the best candidates, on time, and on budget. In the recruitment process, organisations are often faced with hundreds to thousands of job candidates applying for the same position. As many organisations are faced with carrying out the recruitment process manually, this makes the selection process and shortlisting not only time-consuming but also monotonous and tedious. The manual process also suffers from nepotism and bias, which then compromises the type of talent hired by the company. The standard of curriculum vitae selection is bound to be different for every applicant lacking consistency. While the recruitment officer goes through a pool of curriculum vitae, they may miss other key aspects imperative to the job profile. There is a need to automate the selection process of employees for organisations, hence the aim of this project is to design and develop a system that screens a pool of candidates' CVs and recommends candidates according to their ranking based on the defined job profile requirements. The proposed solution for the aforementioned problems faced in the employee selection processes is in the form of a web based application. The web based application as the proposed solution allows the employee selection to be achieved by a click of a button. This research project was achieved through the use of the Rapid Application Development (RAD) software development methodology. The results of the Automated Curriculum Vitae Screening and Recommendation System (ACVSR) are in the form of list of candidates screened and ranked based on their qualifications, experience, skills and comparison with other candidates Curriculum Vitae for a particular job posting. The advantages of this system if it is to be implemented by an organisation are that the system will speed up the selection processes hence shortening the hiring process as a whole, as some companies have lost possible high skilled personnel due to the time it takes to go through the manual hiring process. It will reduce the workload of the human resource department. Transparency to both internal and external stakeholders interested in the hiring procedures will be created and it will aid high return on investment through the high quality human capital employed among other merits. This research seeks to document the project that tries to address the aforementioned problems associated with the manual employee recruitment and selection processes.

Acknowledgements

Contents

List of figures

List of Tables

List of Abbreviations

ACVSRS   Automated Curriculum Vitae Screening and Recommendation System

CV   Curriculum Vitae

HRM   Human Resources Management

NLP   Natural language Processing

RAD   Rapid Application Development

CHAPTER 1

1 Introduction

1.1 Introduction

In the recruitment process, organisations are often faced with hundreds to thousands of job candidates applying for the same position. As many organisations are faced with carrying out the recruitment process manually, this makes the selection process and shortlisting not only time-consuming but also monotonous. The manual process suffers from nepotism and bias, which then compromises the type of talent hired by the company. The standard of curriculum vitae selection is bound to be different for every applicant lacking consistency. While the recruitment officer goes through a pool of curriculum vitae, they may miss other key aspects imperative to the job profile.

The Automated Curriculum Vitae Screening and Recommendation System (ACVSR) will enable a more effective way to shortlist submitted candidate Curriculum Vitae from a large number of applicants. This system will provide a consistent and fair CV ranking policy, which can be legally justified. The system will carry out the ranking process using the required experience and key skills for a particular job position. ACVSR system will rank the CV's based on the experience and other key skills which are required for a particular job profile. The proposed system will help the recruitment team to easily shortlist the candidates based on the CV ranking policy. ACVSR system will focus not only on qualification and experience but also focuses on other important aspects that are required for a particular job position. The system administrator specifies the job required qualifications, experience, and other key skills by setting up the job specification parameters into the system. This system will help the HR recruitment team to select the right candidate for a particular job profile. The job candidates will upload their CVs in a specific format into the system

for screening. The system will then scan the uploaded CVs and give the results of the selected candidates.

1.2 Background

The recruitment of employees is a core function in human resource management dealing with the sourcing of labour (Wright & Storey 1997; Armstrong 1995). Employers often receive large amounts of applications, ranging from hundreds to thousands of applications due to the high unemployment rate in most developing countries. This makes it hard for the Human Resources departments to go through all the CVs with consistency. This process of reviewing large volumes of CVs is often considered as hard manual work and very time-consuming. Sometimes companies lack the time and manpower to review such large amounts of CVs. In many cases the recruitment teams only review less than half of CV volumes that they can, leaving other applicants out of the recruitment process which can be dimmed unfair. Automated systems can be very good solutions to such problems. These solutions to candidate CV reviewing can include recommendation systems and decision support systems. A recommendation system is a subclass of an information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. Recommender systems are utilized in a variety of areas. In this case, the recommendation would be of the most suitable candidates for the HR department. This recommendation would be based on the preferred qualifications, experience, and other key skills of a specific job profile.

Several other similar systems have been developed. The main focus is to reduce the workload from the HR department when reviewing large volumes of applicant's CVs. For example, a system called "An Automated Recommendation Approach to Selection in Personnel Recruitment, is a system based on the German context and may not address the recruitment problems faced by human resources departments accurately, here in Zimbabwe. However, the approach to the

recommendation tools is much similar. In this research, the writer observes that automated recommendation is usually distinguished into content-based filtering and collaborative filtering. The scholar further explains that while content-based methods recommend objects similar to those a user has preferred in the past, methods based on collaborative filtering identify other users with tastes similar to the current user and recommend objects those users have preferred (Balabanovi & Shoham 1998).

Personality Prediction System Through CV analysis is another similar system that uses CV content like qualifications and experience among others to predict a candidate's personality. This system's approach also includes the use of psychometric tests that are taken by candidates before they are shortlisted. The focus of the system to be developed will be on the CV analysis and ranking. Other similar systems fall under automated resumé screening. Screening resumes is the process of sorting resumes to disqualify candidates using successively more detailed examinations of the resumes. The objective being to locate the most qualified candidates for an open job position. The system to be developed will not include the candidates in the shortlisting process. Unlike in the Personality Prediction System Through CV analysis system, there will be no participation from candidates for taking tests and quizzes other than just registering and uploading their CVs when applying for the job. The proposed system will only be used by the administrator and candidates when uploading their CVs. The administrator will capture the open job required qualifications, experience, and other key skills.

Automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT) is another system that seeks to address the recruitment challenges faced by the human resources departments. This system uses several techniques to implement automated candidate screening such as fuzzy matching, semantic-based and rule-based. Here the scholar focused on

using ontology mapping in screening candidates. Like this previously mentioned system. The proposed system will also consider fuzzy matching, semantic-based and rule-based to recommend the best candidates.

1.3 Problem Statement

The HR departments of various organizations have hundreds to thousands of job candidates applying for the same position. This then makes the manual selection process and shortlisting not only time-consuming but also monotonous and tedious. The manual process also suffers from nepotism, which then compromises the type of talent hired by the company. The standard of CV selection is bound to be different for every applicant lacking consistency. While the HR officer goes through a pool of CVs, they may miss other key aspects imperative to the job profile.

1.4 Aim

To design and develop a system that screens a pool of candidates' CVs and recommends candidates according to their ranking based on the defined job profile requirements.

1.5 Objectives

The system should be able to:

1. Capture job profile details.

2. Upload candidates curriculum vitae.

3. Screen uploaded CVs based on job profile requirements.

4. Generate a report of recommended candidates.

1.6 Justification

Companies sometimes receive from hundreds to thousands of applications. Employers often receive a greater response when they accept e-mail or online applications. This then makes it harder and the longer it takes to determine which candidates to invite for interviews. Some companies lack the time or manpower to thoroughly review such high volumes of applications. In such cases, companies may review less than half, or just how many they can realistically assess. Human bias is one of the major factors that affect judgement during the process of candidate selection. The ACVSR system to be developed seeks to address all the problems associated with the manual process of job candidate selection mentioned above. The advantages of this system are:

1. This system can be used in many business sectors that may require expert candidates.

2. The proposed system will reduce nepotism.

3. The proposed system will speed up the selection processes hence shortening the hiring process as a whole, as some companies have lost possible high skilled personnel due to the time it takes to go through the manual hiring process.

4. The proposed system will create transparency to both internal and external stakeholders interested in the hiring procedures.

5. This system will reduce the workload of the human resource department.

6. This system will help the human resource department to select the right candidate for a particular job profile which in turn provides an expert workforce for the organization.

7. The proposed system will aid high return on investment through the high quality human capital employed.

1.7 Scope

The ACVSR will be able to let the administrator input the details for job profiles used for CV screening. The system will allow the job candidates to upload CVs and then screened and ranked according to the job profile requirements. The system will give the output of the ranked CVs as an HR expert would. The ACVSR system will only screen and rank CVs according to the specified criterion. The system will not facilitate the interview process after the shortlisting has been done.

1.8 Expected Results

The expected results from the system are as follows:

User input

The administrator should be able to set job profile details, like qualifications, experience, and skills which are to be used as rules for CV screening. The job candidate should also be able to upload their CV onto the system.

CV Screening

The system can scan CV's received from candidates and undergo the process of CV shortlisting.

Shortlisted CV's Results

The system should be able to display all the shortlisted CV's. The shortlisting of CV's is performed by the system itself.

View result

The administrator can view the results of the CV rankings, which can be easy for an admin to select a desired candidate.

1.9 Document Structure

This is the summary of contents of this research documentation. This document consists of six chapters which give insight about different aspects of the project. The outline of the chapters in the document and their brief descriptions are as below.

1.9.1   Chapter 1: Introduction

This is the introduction to the project research that describes the justification for doing the project. The objectives, methodology and scope of the work are also described.

1.9.2   Chapter 2: Literature Review

This chapter has the literature review that is based on the background of the problem. The domain in which the project is based is explored in this chapter. Literature review also includes materials studied. There are brief reviews of methods on how other scholars approached similar problems.

1.9.3   Chapter 3: Methodology

This chapter focuses on both the research and software development methodologies of study which were used in this project.

1.9.4   Chapter 4: Systems Analysis and System Design

The chapter focuses on the system analysis and design following the selected methodologies in the previous chapter.

1.9.5   Chapter 5: Implementation and testing

The chapter focuses on the implementation of the Automated Curriculum Vitae Screening and Recommendation (ACVSR) system being guided by the analysis and design done in the previous chapter

1.9.6   Chapter 6: Conclusions and Recommendations

This section outlines the overall conclusion of the project and some of the future improvements to the system. This chapter contains an overall summary with the conclusions drawn from the research.

CHAPTER 2

2 Literature Review

2.1 Introduction

This chapter seeks to give the reader insight on what recommendation systems are. Furthermore, some of the available software that are similar to Automated Curriculum Vitae Screening and Recommendation System (ACVSR) will be discussed in this chapter. The researcher will also explain recommendation engine techniques such as Collaborative, Content-Based filtering for fuzzy matching job description with multiple resumes. Natural language processing is another area of interest that the researcher will give insight on.

2.2 Definition of Terms

Curriculum Vitae

The term Curriculum Vitae is a Latin word often abbreviated to CV, which is literally translated to the course of one's life. A CV is a very in-depth document that describes one's career journey step-by-step, including all sorts of personal information. One can look at the CV as a comprehensive description of everything one has ever done, accomplishments, including education, original research, presentations that one has given and papers or books that one has published, including all the successes and achievements.

Recommendation System

Recommender systems are software tools used to generate and provide suggestions for items and other entities to the users by exploiting various strategies (Çano & Morisio, 2017). A

recommendation system is sometimes referred to as a recommendation engine, which is a subclass of information filtering system that seeks to predict the rating or preference a user would give to an item, of which in this case would be prediction of the most preferred candidates for a job position (Wang & Zhang, 2015).

Shortlisting

Shortlisting refers to the process of identifying the candidates from one's applicant pool, who best meet the required and desired criteria for an open position or role in a company, organization or institution and who the organization wants to move forward onto the next step of the recruitment process, which is usually some form of interview (Robertson & Smith, 2001).

Collaborative Filtering

Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users (Konstan & Riedl, 2012). It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user. It looks at the items they like and combines them to create a ranked list of suggestions.

Content-Based Filtering

Content-based filtering focuses on the attributes or descriptive characteristics of items; recommendations are provided based on the similarity of an item to other items (De Gemmis, Lops, Musto, Narducci & Semeraro, 2015).

Hybrid Recommendation System

Since recommender systems are software tools used to generate and provide suggestions for items and other entities to the users by exploiting various strategies. Hybrid recommender systems

combine two or more recommendation strategies in different ways to benefit from their complementary advantages (Çano & Morisio, 2017).

Fuzzy Matching

Fuzzy matching is a method that provides an improved ability to process word-based matching queries to find matching phrases or sentences from a database (Wu, Uehara & O'donnell, 2012). When an exact match is not found for a sentence or phrase, fuzzy matching can be applied. Fuzzy matching attempts to find a match which, although not a 100 percent match, is above the threshold matching percentage set by the application.

Natural language Processing

Natural language processing (NLP) is a branch of artificial intelligence in computer science that assists computers understand, interpret and manipulate human language, in particular programming computers to process and analyse large amounts of natural language data (Reshamwala, Mishra, Pawar, 2013).

Recruitment

Recruitment is the process of finding and hiring the best and most qualified candidate for a job opening, in a timely and cost-effective manner (Martin, 2016). Recruitment can also be understood as the process of identifying an organisational gap and attracting, evaluating, and hiring employees to fill that role. It can also be defined as the process of searching for prospective employees and stimulating and encouraging them to apply for jobs in an organization.

2.3   Recruitment Function

The recruitment process which is sourcing of employees is a core function in human resource management dealing with the sourcing of labour as one of the factors of production (Wright & Storey, 1997). There are different stages of recruitment which are namely the job analysis, sourcing, screening and selection, and on-boarding. The recruitment process has two main broad phases which are namely the attraction phase and the selection phase (Faerber, Weitzel & Keim, 2003).

	Attraction phase	Selection phase
Planning activities	• Definition of overall strategy based on overall personnel planning	
	• Determination of target groups • Design of measures to attract or directly approach candidates from target group	• Determination of selection criteria and methods
Execution activities	• Employer branding • Attraction of direct applications	• Pre-screening • Final selection
	• Applicant management	

Figure 2.1 The Recruitment Function (Faerber, Weitzel & Keim, 2003)

2.3.1   The Four Stages of Recruitment

Job Analysis

The Job analysis stage involves determining the different aspects of a job position. This is done through job description and job specification. The former describes the tasks that are required for the job, while the latter describes the requirements that a person needs to do that job.

Sourcing

Sourcing involves using several strategies like attracting and or identifying candidates. Sourcing can be done by internal or external advertisement. Job advertisements are often done via local newspapers, specialist recruitment agents, professional publications, window advertisements, job centres, or the Internet.

Screening and selection

Screening and selection is the process of assessing the employees who applied for the job. The assessment is conducted to understand the relevant skills, knowledge, aptitude, qualifications, and educational or job-related experience of potential employees. Methods of screening include evaluating resumes and job applications, interviewing, and job-related or behavioural testing.

On-boarding

On-boarding takes place after screening and selection and it is a process of helping new employees become productive members of an organization. This process involves a well-planned introduction that helps new employees to quickly become fully operational.
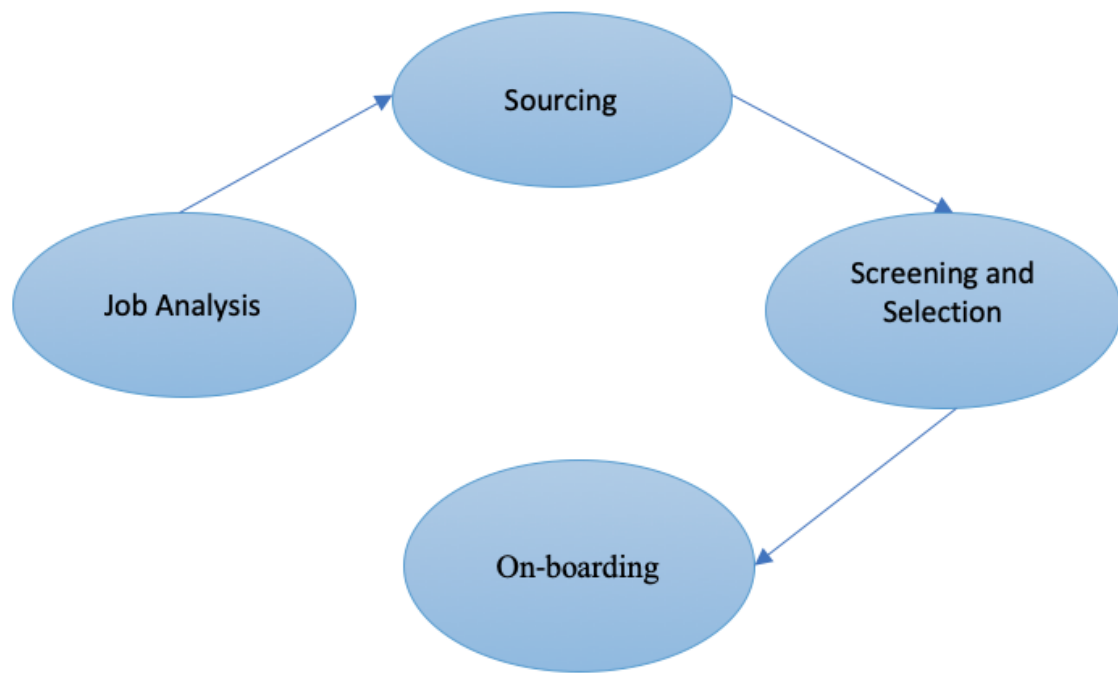
Figure 2.2 The Four Stages of Recruitment (Faerber, Weitzel & Keim, 2003)

2.4 Curriculum Vitae Screening

Curriculum vitae (CV) screening is a process of determining whether a candidate is qualified for a role based on his or her education, experience, and other information captured on their CV. In simpler terms, it is a form of pattern matching between a job position requirements and the qualifications of a candidate based on their CV. The aim of screening CVs is to decide whether to move a candidate forward, usually onto an interview or to reject them.

Screening CVs usually involves a three-step process based on the job position's minimum and preferred qualifications. Both types of qualifications should be related to on the job performance and are ideally included in the job description. Job qualifications often include the following:

1. Work experience
2. Education

3. Skills and knowledge

4. Personality traits

5. Competencies

Three-Step Process of CV Screening

Step 1: Screening CVs based on minimum qualifications

Minimum qualifications are the mandatory qualifications that a candidate must meet to be able to do the job. These types of qualifications are mandatory as they are often considered knockouts because either the candidate has them and can move forward or they don't and gets screened out of the process. Candidates that meet the minimum qualifications move onto the second step of screening CVs.

Step 2: Screening CVs based on preferred qualifications

Preferred qualifications are non-mandatory characteristics that would make someone a stronger candidate for the job. A common example of a preferred qualification is whether the candidate has prior related work experience. These types of qualifications are generally more qualitative than minimum qualifications for example, having strong communication skills. Candidates that meet both the minimum and preferred qualifications move onto the shortlisting step of CV screening.

Step 3: Shortlisting candidates based on minimum and preferred qualifications

Recruiting needs helps with deciding which candidates gets shortlisted for the interview phase. For high volume recruitment, generally all candidates that meet the minimum qualifications move forward to the interview process. For low volume recruitment, generally only the top few candidates that meet both the minimum and preferred qualifications receive an interview.

2.5 Job Candidate Manual Selection Process

After a job advert has been published, the organization starts to receive applications. Employers often receive large amounts of applications, ranging from hundreds to thousands of applications, especially when the company has electronic mail as another option for applying for the vacant post.

Curriculum vitae screening is the second stage after the company or organisation receives applications from prospective candidates. The purpose of this stage is to eliminate the applicants who don't fit the profile that the company is seeking. This process of reviewing and screening large volumes of CVs is often considered as hard manual work and very time consuming. Automated Curriculum Vitae Screening and Recommendation System as proposed by the researcher seeks to make curriculum vitae screening more easier, efficient and with less bias.

Once the applications of prospective candidates have been received, they are then carefully scrutinised by the dedicated recruitment team. The screening process by the recruitment team is crucial because it directly affects the quality of the intake and hence, often the company profits.
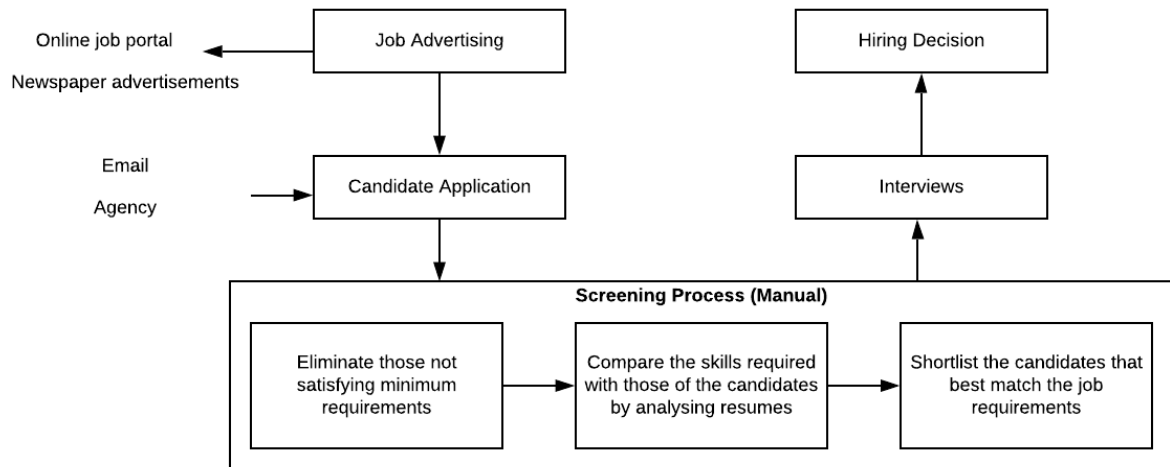
Figure 2.3 Manual Screening Process (Singh, Rose, Visweswariah, & Kambhatla 2010)

Typical manual curriculum vitae screening process.

1. Understanding job profile requirements

The recruitment screening team should understand the requirements for the open job profile in terms of the qualifications, experience criteria if any and other skills that are mandatory and those that are optional but preferable, preference for the job location among other things necessary.

2. Looking through each of the applications

After understanding the open job profile requirements, the recruitment team will then look through each of the applications and have a decision to accept and reject those who do not meet the minimum requirements like qualifications, years of experience and any other skills required for the job.

3. Finding the best match for the job

After the recruitment team has looked through each of the CVs, and rejected the applicants that do not meet the minimum requirements for the job, comes a task of finding applicants that match the

17

job profile. This task requires the recruitment team to read through the CVs in detail and compare it with the job profile. The recruitment team has to make a relative judgment on the candidates at this as the number of applicants to be interviewed is limited. The top few candidates who are shortlisted during this process, undergo further evaluation more often in the form of interviews.

Advantages of Job Candidate Manual Selection Process

1. Character traits can be easily picked by the personnel reading through the CVs and a candidate's character plays a huge role in an organisation apart from their qualifications and experience.
2. Gives the recruitment personnel a chance to see other additional qualities that are on the candidate's CV, which are beneficial to the company.

Disadvantages of Job Candidate Manual Selection Process

1. Reading through large amounts of CVs can be a huge task and time consuming.
2. Some of the important qualities and detail may be missed due to the HR personnel being tired and suffering from fatigue.
3. Due to human nature, there is an element of bias when selecting CVs.
4. The manual process of screening CVs lacks consistency in selection as an HR personnel may fail to use the selection criteria the same way they applied it to the previous prospective candidates.
5. Equal opportunity can be compromised since there is a very high chance that the screening and selection team will not go through all the thousands of CVs from all applicants, which may result in losing talent and likely highly skilled candidates
6. There is often human judgement problems which can often lead to discrimination.

2.6 Automated Recommendation Systems

Automated recommendation systems have evolved over the past years. Especially with the interactive environment of the internet. While users of this vast communication network have access to large amounts of information items and product descriptions they have difficulties to find the right information or preferred products. In this case, companies often receive vast amounts of prospective candidates information, in the form of CVs. This however, leads to difficulties when trying to find the right information or preferred candidates for the open job position.

Automated recommendation is usually differentiated into content-based filtering and collaborative filtering. Content-based methods recommend objects similar to those a user has preferred in the past, while on the other hand methods based on collaborative filtering identify other users with tastes similar to the current user and recommend objects those users have preferred (Balabanovi$f$, Shoham, Breese,1998).

2.6.1 Content-based filtering

Recommender systems are a unique type of information filtering systems. Information filtering involves the delivery of items selected from a large collection that the user is likely to find interesting or useful and can be seen as a classification task. Recommender systems have the purpose of guiding users in a personalized way to interesting objects in a large space of possible options (Meteren & Someren, 2000).

Content-based recommendation systems try to recommend items similar to those a given user has liked in the past. The basic process performed by a content-based recommender consists of

matching up the attributes of a user profile in which preferences and interests are stored, with the attributes of a content object or item, in order to recommend to the user new interesting items.

Based on training data a user model is induced that enables the filtering system to classify unseen items into a positive class c relevant to the user or a negative class c irrelevant to the user. The training set consists of the items that the user found interesting. In the case of a job candidate recommendation system the training set becomes the job opening qualification and skills required for that particular job. These items form training instances that all have an attribute. This attribute specifies the class of the item based on either the rating of the user or on implicit evidence.

A content-based filtering system selects items based on the correlation between the content of the items and the user's preferences as opposed to a collaborative filtering system that chooses items based on the correlation between people with similar preferences (Meteren & Someren, 2000).

2.6.2   Collaborative filtering

Collaborative Filtering is the process of filtering or evaluating items using the opinions of other people. While the term collaborative filtering (CF) has only been around for a little more than a decade, CF takes its roots from something humans have been doing for centuries, sharing opinions with others (Brusilovsky, Kobsa & Nejdl, 2007).

2.7 Natural language processing

Natural language processing (NLP) assists computers understand, interpret and manipulate human language, in particular programming computers to process and analyse large amounts of natural

language data (Reshamwala, Mishra, Pawar, 2013). Natural Language processing is concerned with the interactions between computers and human natural language. Natural languages are languages spoken by humans. Natural language is any language that humans learn from their environment and use to communicate with each other.

Whatever the form of the communication, natural languages are used to express our knowledge and emotions and to convey our responses to other people and to our surroundings. Natural languages are usually learned in early childhood from those around us. Currently we are not yet at the point where these languages in all of their unprocessed forms can be understood by computers. Natural language processing involves the use of numerous techniques to try and make computers understand human languages. In the proposed system, natural language processing can be used to read, understand and make sense of the curriculum vitae text content.

Natural language processing (NLP) is deep and diverse and is a collection of techniques used to extract grammatical structure and meaning from input in order to perform a useful task as a result, natural language generation builds output based on the rules of the target language and the task at hand. NLP is useful in the tutoring systems, duplicate detection, computer supported instruction and database interface fields as it provides a pathway for increased interactivity and productivity. The proposed system can make use of natural language processing to read, make meaning and analyse curriculum vitae text content. In this research the NLP approach can be implemented as a backend algorithm using Python programming language.

2.8 Similar Systems

There are several other systems that deal with candidate CV screening, selection and candidate recommendation. This section of this research will look at other systems which are similar to

Automated Curriculum Vitae Screening and Recommendation System (ACVSR). The following are similar systems related to the proposed system:

2.8.1 Personality Prediction System Through CV Analysis

Personality Prediction System Through CV Analysis enables a more effective way to short list submitted candidate CVs from a large number of applicants providing consistent and fair CV ranking. This system ranks the experience and key skills required for a particular job position. This system focuses not only on qualification and experience but also focuses on other important aspects which are required for a particular job position. This system seeks to help the human resource department to select the right candidate for a particular job profile which in turn provides an expert workforce for the organization.

In this system the candidate registers him/herself with all their details and uploads their own CV into the system which will be further used by the system to shortlist their CV. Unlike in the proposed system, here the candidate can give an online test which will be conducted on personality questions as well as aptitude questions. After completing the online test, candidates can view their own test results in graphical representation with marks.

Features:
- This system automatically determines the key skill characteristic by defining each expert's preferences and ranking decisions.
- This system automates the processes of requirements specification and applicant's ranking.
- Produces ranking decisions that are relatively highly consistent with those of the human experts.

- This system enables a more effective way to short list submitted candidate CVs from a large number of applicants providing a consistent and fair CV ranking.

How Personality Prediction System Through CV Analysis works

- It uses Artificial Neural Network

- This system assigns weight for each requirement

- Resume will be shortlisted based on overall weight

Example: -

Weight=0.75

Java

Requirement

Weight   =   0.80

C++

Resume>=2.05

Weight=0.50

Experience- 2 Years
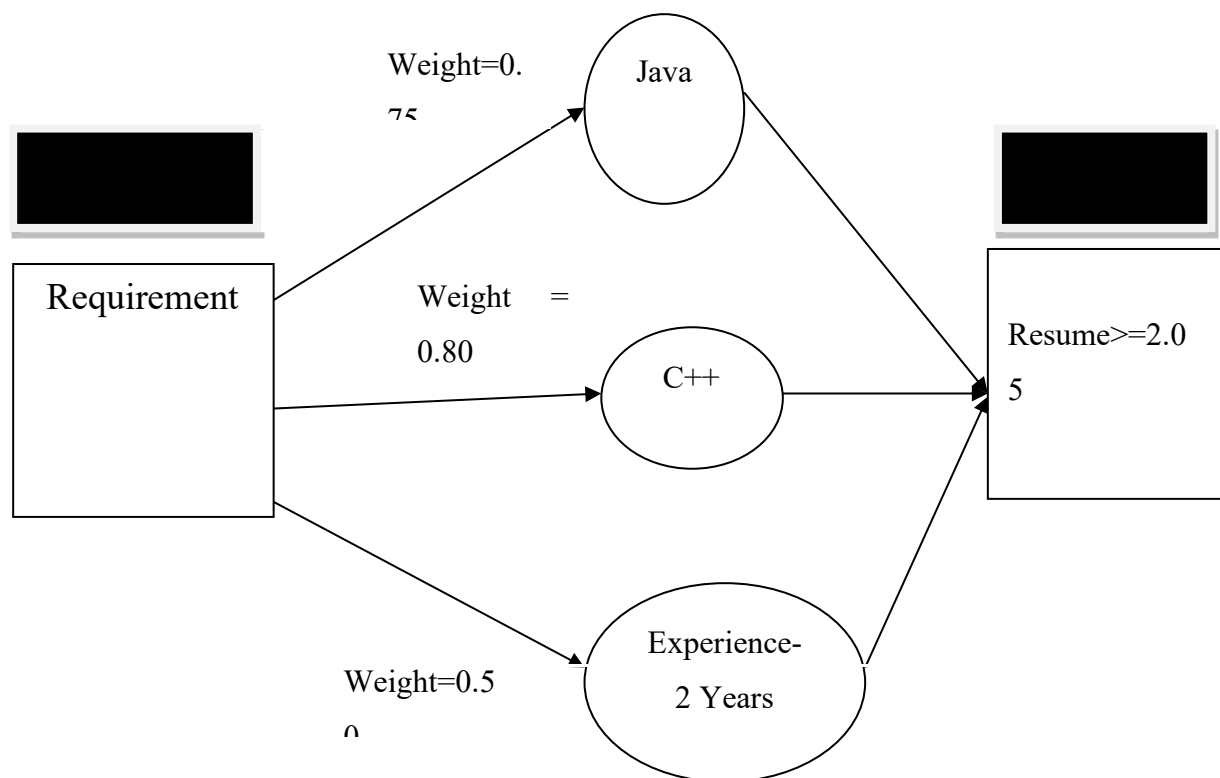
Figure 2.4 An example of how Personality Prediction System Through CV Analysis works

Personality Prediction System Through CV Analysis modules:

The system comprises of 2 major modules with their sub-modules as follows:

Admin:

- **Login:** Admin logs in with its valid credentials to access the below modules.

- **Add Personality Questions:** Here, admin can add all the personality related questions.

- **Add Aptitude Questions:** Admin can add aptitude questions of three subjects such as English, math's and programming with its respective four options.

- **Add Job Details:** Admin or any authorized person can add the requirement or job details on behalf of the company. System allows admin to job details such as position, experience, salary, etc.

- **Preferred CV's:** Admin will add some CV's which are preferable and will be used to shortlist the scanned CV's.

- **Scan CV:** Admin can scan CV's received from candidates which will undergo the process of shortlisting the CV's. Based on the company requirement, the candidate who has submitted their CV to the admin or any authorized person, will be uploaded by the admin into the system.

- **Shortlisted CV's:** Here, all the shortlisted CV's of candidates will be displayed. The shortlisting of CV's is performed by the system itself.

- **View Candidates:** Can view all the registered candidates with its details.

- **View Result:** Can view the results of individual candidates, which can be easy to admin or concern a person to select a desired candidate.

Candidate:

- **Registration:** To access the below given modules, candidate must fill up registration form and create login credentials in order to get the access to the system. While registration process, candidate need to upload their CV.

- **Login:** Candidate need to enter valid credentials to access the below given modules.

- **Give Test:** After successful login, candidate can now proceed with online test based on personality and aptitude.

- **View Results:** Once the test is completed by the candidate, the results will be displayed in a graphical representation.

Advantages:

- This system can be used in many business sectors that may require expert candidate.

- This system will reduce workload of the human resource department.

- This system will help the human resource department to select right candidate for particular job profile, which in turn provide expert workforce for the organization.

- Admin or the concern person can easily shortlist a candidate based on their online test marks and can select an appropriate candidate for desired job profile.

Disadvantage:

- This system requires large memory space as it stores data related to CV's.

- Requires an active internet connection.

- May provide inaccurate results if data not entered properly.

2.8.2 An Automated Recommendation Approach to Selection in Personnel Recruitment

This system is based on the German context and may not address the recruitment problems faced by human resources departments accurately, here in Zimbabwe. However, the approach to the recommendation tools is much similar. In this research, the writer observes that automated recommendation is usually distinguished into content-based filtering and collaborative filtering. The scholar further explains that while content-based methods recommend objects similar to those a user has preferred in the past, methods based on collaborative filtering identify other users with tastes similar to the current user and recommend objects those users have preferred (Balabanovi*f* & Shoham, 1997).

In this research, the author explains that in the first case preference profiles of users have to be compared to object attributes and in the latter case the similarity of preference profiles has to be determined. When applying this approach to the general problem of assessing human attributes or the more specific problem of predicting a candidate's aptitude, the preferences would correspond to the individual perception of someone else's attributes and to situational, for example, job-specific requirements. The object of preference would be the assessed person.

The model that the author used is built on a hybrid approach, that is to say, both concepts, content-based filtering and collaborative filtering, were applied simultaneously. This helps to partially overcome the problem of rating data sparsity by leveraging synergies between the two approaches in a combined model (Rashid, Sarwar & Melville, 2002).

Another concept that the author applies is the latent aspect model which mainly has been used for latent semantic analysis in information filtering (Hofmann, 1999). Hofmann and Puzicha (1999) as well as Popescul (2001) and Schein (2002) present latent aspect models for collaborative filtering and hybrid approaches. In a basic approach for collaborative filtering, we look at observations of user/object pairs (x,y), where X is a set of users and Y is a set of objects. For the basic model, observations are just co-occurrences of users and objects representing events like "user x has accessed object y", i.e. preference values are not explicitly considered. The aspect model can then be represented as a latent variable model using a latent aspect variable which is associated with each observation (x,y), assuming that x and y are independent conditioned on z.

Model Specifications

So far the basic principles of the latent aspect model have been explained. Hofmann and Puzicha (1999) introduced an additional variable v into the model representing a rating value.

In the case of the given scenario by the researcher, the variable x represents the assessor together with the construct the attribute to be assessed and the clinical assessment method that is used. The latter mainly defines which input data for example, a candidate's resume, is considered for assessing the attribute. The variable z stands for the latent aspects that influence the assessment, variable v is the assessed attribute value and variable y is the assessed person. By using this model structure, the assessed person is independent of x. This is a realistic assumption since the selection of the candidates to be assessed is not part of the assessment process itself (Faerber , Weitzel & Keim, 2003). The explained model still represents a pure collaborative filtering approach. In order to leverage the knowledge of the input data for a specific assessment method for example, the resume data in the case of resume screening, one replaces variable y with a variable a representing the attributes that are used as input data, see figure below.
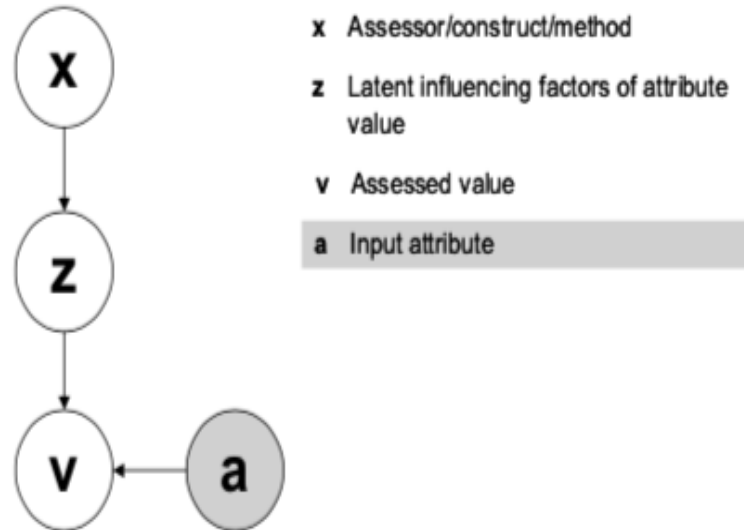
Figure 2.5 Hybrid Assessment Model

On the application of the model on synthetic candidate resumes. According to the researcher, in order to show the feasibility of applying the latent aspect model to the assessment problem, synthetic resume data and resume screening was used as a form of clinical assessment to test the model. Therefore, the researcher defined a target job description to be presented to the assessors. The target attribute to be assessed is the aptitude for this specific job with the domain consisting of the values 'qualified' and 'not qualified'. With a relatively small number of 70 candidates, the researcher had to limit the variety of possible input data and, hence, the variety of resumes. (Faerber , Weitzel & Keim, 2003).

In order to test both, the collaborative as well as the content-based effects of the model the researcher used 4 assessors $\{x1,...,x4\}$ in the model, with the construct to be assessed and assessment method remaining constant. The actual assessments, however, were conducted by only one person to eliminate the effects of too much variance in the assessments of the target attribute. The assessed values for profiles 1 to 15 were then assigned to x1, for profiles 16 to 30 to x2, and so forth in a first test run. The researcher also tested a complementary approach to deal with the

sparsity problem which still exists even though they used a hybrid approach. In order to estimate the model parameters they used the 60 assigned assessment of the 70 candidate profiles to create an original rating matrix R that assigns assessed values to profiles:

$$R = (r_{x,y,v}) \text{ with } r_{x,y,v} = \begin{cases} 1 & \text{if assessor } x \text{ assesses the target attribute} \\ & \text{of person } y \text{ with value } v \\ 0 & \text{otherwise} \end{cases}$$

The above matrix was transformed by treating the ratings of persons as ratings of all the attributes extracted from the resumes. As many attributes are assigned to several profiles and, hence, might be observed several times with different values v, the entries of the transformed matrix are actually not either 0 or 1 but take values in the interval [0;1] according to the relative frequency of value v being assigned to attribute a by assessor x. As even in a realistic scenario the transformed value matrix is still rather sparse, a simple linear interpolation to complement the matrix was used.

After estimating the model parameters, the assessments for candidate profiles 61 to 70 were then predicted by the model. The results showed the signs "+" and "-" standing for the original assessments of these profiles which are used to test the prediction quality of the model. These assessments were conducted by the same person as the ones for profiles 1 to 60.

2.8.3 Automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT)

Automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT) is another system that seeks to address the recruitment challenges faced by the human

resources departments. This system uses several techniques to implement automated candidate screening such as fuzzy matching, semantic-based and rule-based. Here the scholar focused on using ontology mapping in screening candidates.

2.9 Conclusion

This chapter summaries on the research work that has been done in the area of CV screening, candidate selection and recommendations systems. CV screening can be achieved in several approaches taken by other scholars from the literature review. These approaches include concepts like fuzzy matching. The researcher has developed interest in Natural Language Processing (NLP) as the approach chosen for this project because of its accuracy in dealing with natural languages from the amount of relevant information provided on CV's. The next chapter gives an insight on the research and software development methodology used by the researcher.

CHAPTER 3

3 Methodology

3.1 Introduction

A methodology is a systematized, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles related with a branch of knowledge (Irny & Rose 2005). Methodologies assist in guiding the researcher to derive a best solution. The structure, plan and control processes of the research and development of the proposed system will be guided by the chosen methodologies. Software development methodologies are an essential supporting structure that is used to plan, structure, and control the method of developing information systems. Methodologies provide a formal and structured way of researching and developing an information system, and they impose a disciplined process on software development with the goal of making software development more predictable and efficient. For the development of the proposed system to be a success, the researcher needs to implement the most ideal research methodology and systems development methodology in the project. In this section, candidate methodologies are reviewed and evaluated in relation to the development of the proposed system.

3.2 Research Methodology

Research refers to the systematic investigation into and study of materials and sources in order to establish facts and reach new conclusions (Johnson 2006). A research methodology is the strategy or architectural design by which the researcher maps out an approach to problem finding or problem solving (Jamshed 2014). It is important for the researcher to choose the correct research

methodology, to get the most effective results from the research outcomes. This also applies to the proposed system. Research methodologies define what activities constitute the research, how to proceed, how to measure and what constitutes the success of a project as a whole.

3.2.1 Build Methodology

The build methodology is most ideal where the research leads to development of software hence it is adaptable in this project to build the Automated Curriculum Vitae Screening and Recommendation(ACVSR) system. The methodology has good practices which are beneficial to the project. The practices include design of the software, component reuse, adequate programming language selection and testing.

The build methodology encourages planning before anything is built because failing to plan is planning to fail as far as software development is concerned. A modular approach is ideal as it simplifies testing which would be carried out later in the project. A modular approach helps in reducing the complexity of the development procedure as functions of the system are developed as modules. The methodology also employs code reuse hence we get to use other software components which are readily available when necessary. Code reuse reduces development time of the project.

The build methodology involves the practice of selecting the programming language for the proposed system therefore the student can get to use the languages he is already familiar with in order to reduce time invested in learning new languages. In this project, the student intends to make use of web technologies and scripting languages like Javascript and Angular. The student is also able to get help from the supervisor therefore learning time and development time is reduced.

Another practice that is incorporated by the build methodology is testing. Testing is essential in the development of the system both during the development and when the final system has been developed. Modular testing should be done to make sure components are performing their functions and it makes debugging of the system easier as one can debug and identify errors in each module. System testing will be done after modular testing to test interaction of modules. The build methodology is one of the highly research methodologies considered for the proposed project as it is useful because it involves the use of practices that are ideal for this project.

3.2.2 Model Methodology

Modelling is a methodological part of science, it is not the object of research. In reality, no particular single modelling approach is applicable to all systems, modelling approaches are customised. The model for the proposed system has to capture the most important aspects for the study of the most critical aspects. The focus has to be on the critical aspects because they are enough for model building. The decision of what is important and what is not constitutes the modelling strategy. The success of applying this methodology is dependent on the weighting of aspects, a misleading model is produced when important aspects are eliminated or when there is over-emphasis on the less important aspects.

After development of the model, the model will have to undergo certain operations, called simulations so as to test the functionality of the model. Experimenting with the model results in formulation of a formal description of the model to verify the functionality and correctness of the system and this task is called Model Checking. Using the Model Methodology allows the exploration of a variety of options of different designs before actual implementation of the

proposed system hence we can select the best design from those tested and ensure we build a better ACVSR system.

3.2.3 Formal Methodology

A formal methodology is used to prove facts about algorithms and systems. Formal methodologies are very common mainly in theoretical computer science. The complexity of the algorithms to be used in the development of the ACVSR system can be determined by employing this methodology and hence determine the overall complexity of the operations in the system. This methodology allows one to take particular interest in important factors for the proposed system that is the time complexity and the space complexity of the algorithms adopted by the system. Mathematical proof techniques can then be employed to determine the complexity.

3.3   Software Development Methodology

Software development methodologies are important in the development of software systems. There are several software development methodologies used by professional software engineering companies. There are advantages and disadvantages related with each of the methodologies. The purpose of each of these methodologies is to provide smooth and easy software development.

3.3.1 Scrum Methodology

The Scrum Development Methodology is straightforward and easy to understand. This methodology is best suited for development of projects that are rapidly changing or with highly emergent requirements. Scrum is based on the assumption that the environment, technologies and other variables are likely to change during the process. The Scrum software

development model begins with brief planning, meeting and concludes with a final review. Scrum development methodology is used for development of software with less time, which includes a series of iterations to create required software. Scrum is team-centred and can be implemented by small teams hence it is suitable for this project. Core members of the team are scrum master and the developer. In the case of this research, the supervisor can play the role of the scrum master as they are an expert, and the researcher, the developer.

Five phases of Scrum methodology:

Product Backlog Creation:

A product backlog refers to a list that consists of system features that should be implemented during the development process. This list is ordered by priority and every item is called a User story.

Sprint Planning and Sprint Backlog Creation:

 At this phase, the duration of sprints is determined, a short sprint will allow the researcher to release the working version of the proposed system more frequently. and hence get more feedback from the supervisor as far as meeting objectives is concerned. However, a long sprint will allow the developer to work more thoroughly. The best duration will be a compromise of the two alternatives so that we get to harvest the advantages of both. The sprint backlog will be developed at this phase and consists of objectives that will be met during the current sprint.

Working on the Sprint and Scrum Meetings:

On a daily basis, scrum meetings are carried out with the goal of getting full information about the current project status. The burndown chart shows one how many objectives are not met

and therefore it gives the team the ability to control the development process and it should be updated after every meeting.

Testing and Product Demonstration:

In scrum, the result of every sprint is a working product, hence the full life-cycle testing process is essential. At this phase the proposed system will be tested against objectives.

Retrospective and Next Sprint Planning:

The main aim of this stage is to discuss the results and determine the ways to improve the development process. Therefore, at this phase results are discussed and ways to improve the development process on the next step are determined. Here the team discusses the current state of the development and concludes what is to be improved.

Advantages of Scrum

Scrum methodology has several advantages which are:

- Scrum is good for small and fast moving projects which is perfect and an advantage for the size of the proposed project.
- Decision making lies entirely in the hands of the teams, hence the project will be controlled.
- Scrum is a lightly controlled method which totally empathizes on frequent updating of the progress, therefore, project development steps are visible in this method.
- As scrum emphasises on daily meetings. This will easily help the researcher to make it possible to measure individual productivity. This leads to the improvement in the productivity of the student.

Disadvantages of Scrum

Scrum methodology has two major disadvantages which include:

- Scrum methodology needs experienced team members only. If the team consists of people who are novices, the project cannot be completed within the forecasted time frame. In our case, it is the supervisor who is experienced and the student is not.

- Scrum often suffers if the project cost estimations and time will not be accurate.

Implementing scrum in this project will be another open option since the project has such characteristics that suit it and can be easily adopted as the methodology. The advantages of the scrum methodology outweigh its drawbacks.

3.3.2 Forever Alone Methodology

Forever alone is a methodology that is ideal for solo programmers or small teams. The forever alone methodology was derived from the Agile Unified Process which was derived from the Rational Unified Process. This methodology has four phases derived from those of the Agile Unified Process and have now been customised for small teams or a solo programmer. The forever alone methodology does not require special tools hence it is not expensive to implement.

The phases of the forever alone methodology are as follows:

Scope Definition

This phase's main objective is to identify the initial scope of the proposed project. At this phase, all team members gain general understanding of the proposed project. The objectives

are the ones that define the project scope. After defining the objectives, we can then estimate the schedule for the project and agree on project milestones.

Feasibility Study and planning

The objective is to ensure that the team can actually develop a system that satisfies the requirements. Feasibility study is done at this phase to see whether we can be able to complete the project and meet the objectives. If the project is not feasible we revise the objectives of the proposed system.

Implementation and Testing

At this stage the objective is to actually build the working proposed system within the time frame that would have been allocated to the phase. Activities of this phase include preparation of the programming and testing environment. Coding is the main activity of this phase as it is through coding that we transform the proposed system from an idea into an existing and usable product. Testing of the product is done after development so as to uncover any bugs in the system.

Deployment

The objective is to validate and deploy the proposed system on the internet. Deploying the system will allow the team to carry out beta testing. This phase marks the end of the project and maintenance of the proposed system.

3.3.3 Rapid Application Development (RAD)

Rapid application development (RAD) is a software methodology initially developed by James Martin in the 1980`s (Dave, 2005). Rapid Application Development is a software development process which allows software systems to be developed in minimal time and resources hence it suits the proposed project. RAD is a people-centred and incremental development approach hence user involvement is of great essence in the methodology and so is collaboration and co-operation between all stakeholders. The stakeholders will be the student, supervisor and users.

With RAD methodology the student can take advantage of user involvement throughout the RAD life cycle ensuring that business requirements and user expectations are clearly understood. RAD makes use of powerful application development tools to develop high quality applications rapidly. Prototyping is used to assist users with visualizing and then requesting changes to the system as it is being built, allowing applications to evolve iteratively. Techniques associated with RAD are also very successful when faced with unstable business requirements or when developing non-traditional systems. RAD ensures that developers build systems that the users really need with the emphasis of its life cycle. The RAD life cycle, through its four stages, includes all of the activities and tasks required to scope and define business requirements and design, develop, and implement the application system that supports those requirements.
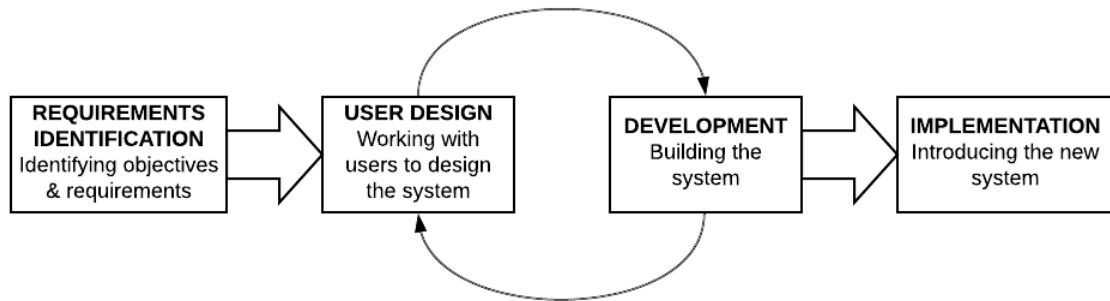
Four phases of RAD Methodology:

Figure 3.1 Rapid Application Development (Mackay, Carne & Beynon-Davies, 2000)

I.   **Requirements Planning**

This stage involves the concept definition. At this stage, the business functions and data subject areas that the system will support and determine the system's scope are defined. At this phase, the student familiarises themselves with other existing systems. This phase kicks off with a meeting between the supervisor and student.

II.   **User Design**

User design phase is also known as the Functional Design Stage. At this stage, the student uses workshops to model the system's data and processes and to build a working prototype of critical system components. The appropriate design approach of the proposed system is determined at this stage.

III. **Development**

This stage is also known as the construction stage. At this stage, the student constructs and completes the physical application system, builds the conversion system, and develops user aids and implementation work plans. The software is developed based on the objectives

defined in the project definition phase. Testing and verification of the system is done at this phase and once the system passes the tests it will be ready for deployment.

IV. Implementation

This phase is also called the deployment phase. This stage includes final user testing and training, data conversion, and the implementation of the proposed system.

Advantages of RAD

Rapid application development methodology has the following advantages when taken into consideration:

- RAD aids increased speed of development, this is due to the use of CASE tools, the goal is to capture requirements and turn them into usable code as quickly as possible.
- RAD emphasizes quality, hence increased quality. Quality is defined by both the degree to which the application to be delivered meets the needs of users as well as the degree to which the deliverable system has a low maintenance cost.
- RAD emphasizes the involvement of the user in the analysis and design stages hence increasing quality.
- RAD is hardly affected by changing requirements.
- RAD is suitable for the proposed system as the scope is narrow.

Disadvantages of Rapid Application Development

RAD has two main disadvantages:

- Reduced scalability because a RAD developed application starts as a prototype and evolves into a finished application.

41

- RAD is often associated with reduced features. This occurs due to time boxing where features are pushed to later versions in order to finish a release in a short amount of time.

Rapid Application Development methodology is the chosen software methodology for the proposed system. This decision was greatly influenced by the RAD advantages which outweigh its disadvantages. RAD is also a software development process which allows software systems to be developed in minimal time and resources hence it suits the proposed project.

3.4 Conclusion

This chapter gave an insight on the software development methodologies and models, making emphasis of the advantages and disadvantages of each. The Build research methodology is the most ideal research methodology since it has good practices which are beneficial to the project. The practices include design of the software, component reuse, adequate programming language selection and testing. RAD is the chosen software development methodology for developing the proposed system considering its strengths relating to the system to be developed. Using a prescribed methodology helps in guiding the development process as it provides a framework for making changes in a controlled manner. Although this chosen methodology may have its weaknesses or setbacks, its benefits outweigh its shortcomings as discussed. The implementation of this methodologies requires expertise so the student has to implement this methodology with the supervisor's guidance.

CHAPTER 4

4 System Analysis and Design

4.1 Introduction

With Systems Analysis and Design (SAD) analysts continue to learn new techniques and approaches in developing systems more effectively and efficiently (Dennis, Wixom & Tegarden, 2015). Systems analysis allows one to study existing systems. At this stage, the developer identifies information on and about the current system and fully examines it. The system specifications, user requirements and how the proposed system is expected to work is obtained through the systems analysis exploratory procedures. When the information has been completely obtained using these various techniques, it will help to design an appropriate ACVSR system that would meet the requirements of the users.

4.2 Objectives Definition

Objectives definition seeks to explain and describe exactly the nature, scope, and the meaning of the ACVSR system objectives. The objectives of a system must be unambiguous, feasible and measurable. There are several research techniques that can be used to come up with objectives including interviews, questionnaires, document reviews and observations. The research techniques that were used in this project to come up with the objectives include interviews and document reviews.

An interview is a structured conversation often with two parties where one participant asks questions as an interviewer, and the other provides answers as the interviewee (Webster, 2016).

Interviews were one of the techniques used to gather information for formulating the system objectives. The researcher managed to interview several HR practitioners, for example one from Proton Bakers. Document reviews as a technique focuses on the documentation generated by software engineers, including comments in the program code, as well as separate documents describing a software system (Lethbridge, 2005). Document review involved the review of the current system documentation, together with documents such as company policies, and other similar documents. Documents written about the system are very useful. Reviewed documents often contain conceptual information. They can also serve as an introduction to the software. Document reviews require one to be careful in not using outdated material as technological advancements are very rapid in this information age. Studying the documentation is often time consuming as this requires some knowledge of the source.

Online research was another method used to gather information. Online research methods involve using web browsers to access websites to read articles on the topic that one is researching on and watching online video content, on research facilities such as YouTube. Online research methods help people find the information they need quickly. Online tools provide information instantly. The manual way of gathering information would take much longer to locate using offline research methods such as searching for books in a library.

4.3 Objectives of the system

The proposed system objectives determine what the automated curriculum vitae screening and recommendation system has to accomplish after development. Objectives are defined in the objective definition phase and are checked by the supervisor to make sure that they are measurable, unambiguous and feasible. The objectives for the system are to:

- Capture job profile details.

- Upload candidates curriculum vitae.

- Screen uploaded CVs based on job profile requirements.

- Generate a report of recommended candidates.

In the ACVSR system both users, the job candidate and the administrator will have accounts. The job candidate as a user will have to create a profile on the system as the first step. Upon creation of their account, the job candidate will be able to upload their curriculum vitae information into the system and search for jobs posted by an administrator account (Skill seeker). The job candidate will be able to apply for jobs that they feel best matches their profiles. The administrators will be able to post jobs and specify each job profile requirements. The ACVSR system will allow admins to view job applicants and reports on applicants screening and recommendations based on which CVs were highly rated.

The ACVSR system will make use of natural language processing to read, make meaning and analyse curriculum vitae text content. Natural language processing (NLP) is a branch of artificial intelligence in computer science that assists computers understand, interpret and manipulate human language, in particular programming computers to process and analyse large amounts of natural language data. In this research the NLP approach is implemented as a backend model using Python programming language. Below follows a graphical representation of how natural language processing generally works.
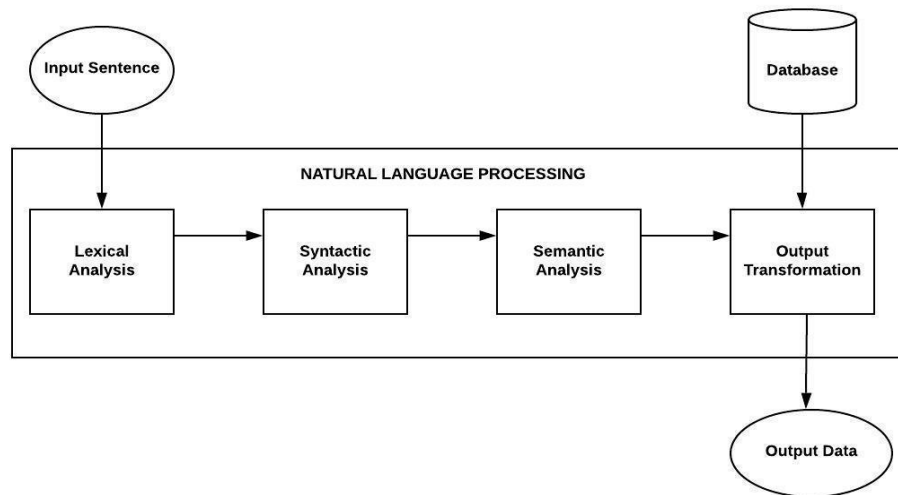
Figure 4.1 Natural Language Processing (Tapsai, Meesad & Haruechaiyasak, 2016)

4.4 Requirements Planning

As the research takes the approach of the Rapid Application Design methodology, this is the first stage of the RAD life cycle. In the requirements planning phase, the researcher met at least two users to discuss identified objectives of the proposed system and to identify information requirements arising from these objectives. This phase required intense involvement from both parties, the researcher and stakeholders. The purpose of the requirements planning phase is focused on solving business problems. Some systems and technologies may drive some of the solutions proposed, however, the focus should always remain on reaching business goals. System requirements are a more detailed description of the software systems functions, services and operational constraints. A use case is often a methodology used in system analysis to identify, clarify, and organize system requirements.

System requirements are either functional or non-functional requirements. Functional requirements indicate the services that the system shall provide such as allowing a job candidate

46

to upload their CV in the ACVSR system. Functional requirements can also specify what a system shall not do. On the other hand Non-functional requirements are not directly related to the functional behaviour of the ACVSR system. Instead Non-functional requirements represent the quality attributes of the system. Some of the non-functional requirements to describe the ACVSR system are that of the system being easily understandable to users, portable and easily accessible.

Functional Requirements

- The system shall allow users to register user accounts.

- In the case that a candidate finds a job they are interested in, the system shall facilitate the candidate's application and CV upload.

- The system shall screen applicants CVs based on the open job profile requirements.

- The system shall be able to show the administrator a report on job applicants and their CV screened ratings.

- The system shall store candidate profile information and all job applications in a database.

Non Functional Requirements

- The web page must be user friendly, simple and interactive.

- The user interface is designed in such a way that an average user should be able to view all vacant job posts and apply without having to face any challenges.

- The ACVSR system should be portable and capable to function on any operating system, be it in Windows, Linux or MacOS.

- The ACVSR system shall be able to function on any web-browser platform since it is a web based application.

4.5 System Modelling

System modelling marks the beginning of the implementation and testing phase. The analyst should model system requirements in order to focus on the system's external behaviour separately from its internal design. System modelling aids in planning the order in which features will be developed. The models will be used as a basis for system tests, making a clear relationship between the tests and the objectives and when the objectives change.

4.5.1   Use Case Diagram

A use case is an object-oriented modelling construct that is used to define the behaviour of a system. User interaction with the system is often described through a prototypical course of actions along with a set of alternative courses of action that a user can take (Grechanik et al, 2007). There are two users of the system who are the job candidate and the system administrator. The job candidate as an actor can interact with the system in the following ways, creating their user profile, viewing available jobs, applying for jobs and uploading their CV. The administrator as an actor can also create their profile, post jobs, view job applicants and view results from applications. The use case diagram for the ACVSR system is shown in the Figure below.
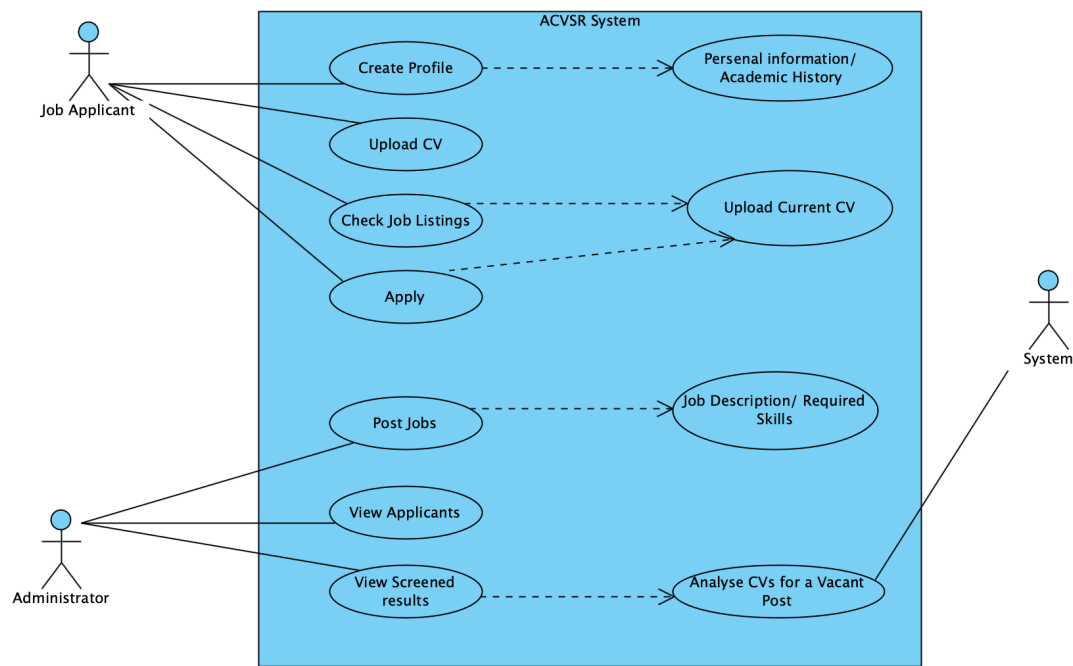
Figure 4.2 Use Case Diagram

4.5.2 Sequence Diagrams

A sequence diagram illustrates system object interactions arranged in time sequence. Sequence diagrams illustrate objects and classes involved in the scenario. Sequences of messages exchanged between the objects needed to carry out the functionality of the scenario are also shown. For the ACVSR system there are two main users of the system, the job candidate and the administrator. For each user there is a different sequence diagram showing object interactions for each user. The ACVSR system sequence diagrams are illustrated in the figures below.
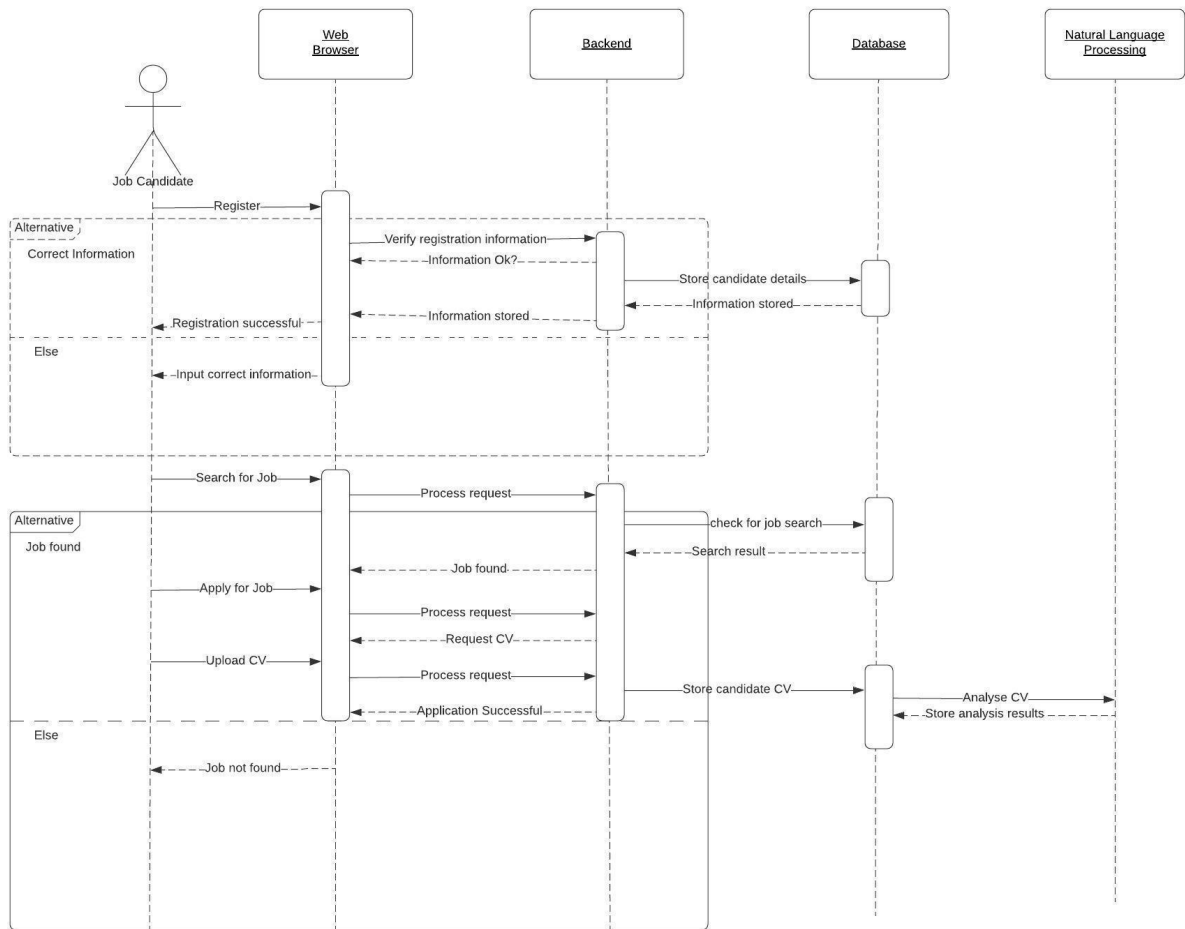
Figure 4.3 Job candidate Sequence Diagram
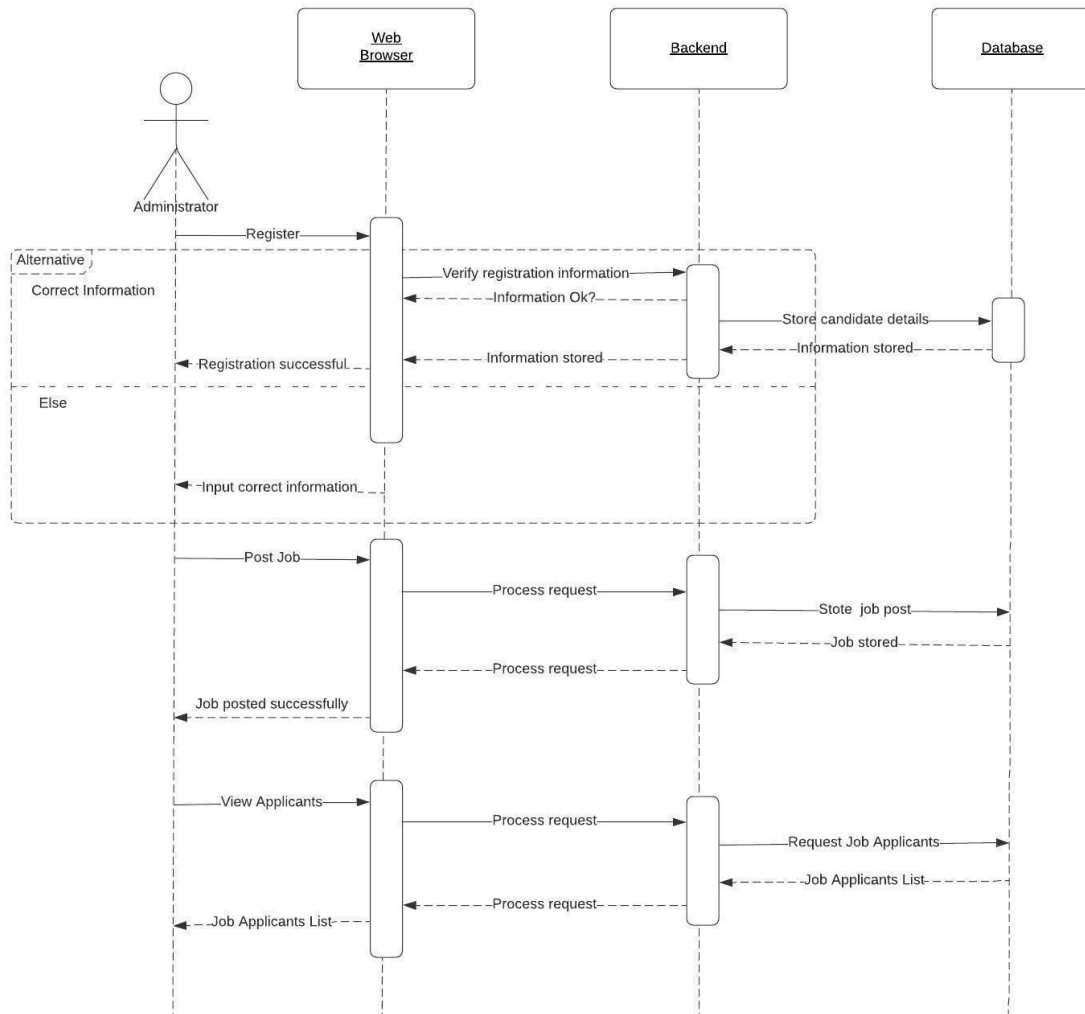
Figure 4.4 Administrator Sequence Diagram

4.5.3   Activity Diagram

An activity diagram is a behavioural diagram, this simply means that it illustrates the behaviour of a given system. An activity diagram shows the overall control flow from a start point to a finish point portraying the various decision paths that exist while the activity is being executed (Ahmad, Iqbal, Ashraf, Truscan and Porres, 2019). Below follows the sequence diagram for the ACVSR system depicting overall control flow from a start point to a finish point portraying the various decision paths.
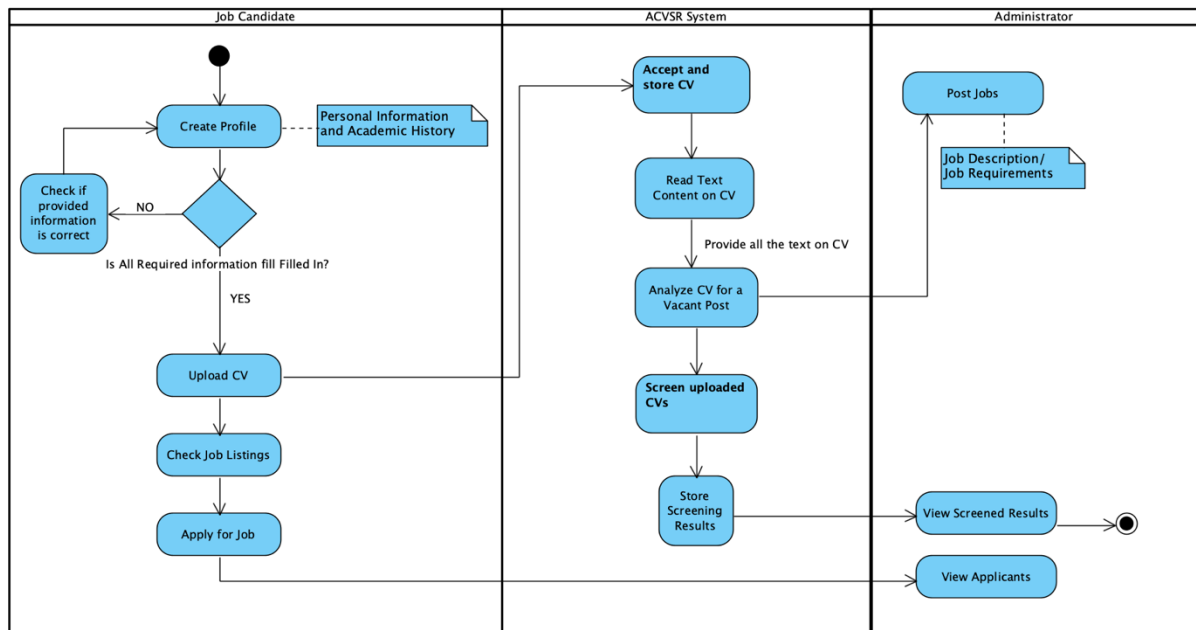
Figure 4.5 ACVSR System Activity Diagram

4.6 User Design

User design marks the second phase of the RAD methodology adopted in this research. This phase follows after the project was successfully scoped out from the requirements planning phase. This marked the beginning of development, building out the user design through various prototype iterations. Analysis refers to breaking down a system into smaller components with the intent of understanding the component's nature, functionality and interrelationships with either its environment or other components (Sommerville, 2011). Modelling user interaction helps to identify user requirements. The Unified Modelling Language (UML) is a developmental, modelling language. UML's intended purpose is to provide a standard way to visualize the design of software systems (Wesley, 2005). Therefore UML diagrams were used to model the ACVSR system.

Systems design involves defining the modules, interfaces, architecture, components, and data for a system to satisfy specified requirements (Garlan, 2003). Systems design also involves overall product like the subsystems and the system architecture that compose the product, and the way subsystems are allocated to processors are depicted using the system design. Systems analysts make use of models to assist them with understanding the functionality of the system.

4.6.1 Architecture Design

System architecture illustrates how a typical software system might interact with its users, external systems and data sources (Garlan, 2003). Software architecture is often designed into three layers, which are, from top to bottom, presentation layer, service layer and business layer as in the case of the ACVSR system. The objective of architecture design is to determine how the software components of the information system are assigned to the hardware devices of the system (Dennis, Wixom, and Roth, 2012). The following is a system architecture diagram of the automated curriculum vitae screening and recommendation system thus showing how the software components of the system are assigned to the hardware components of the system.
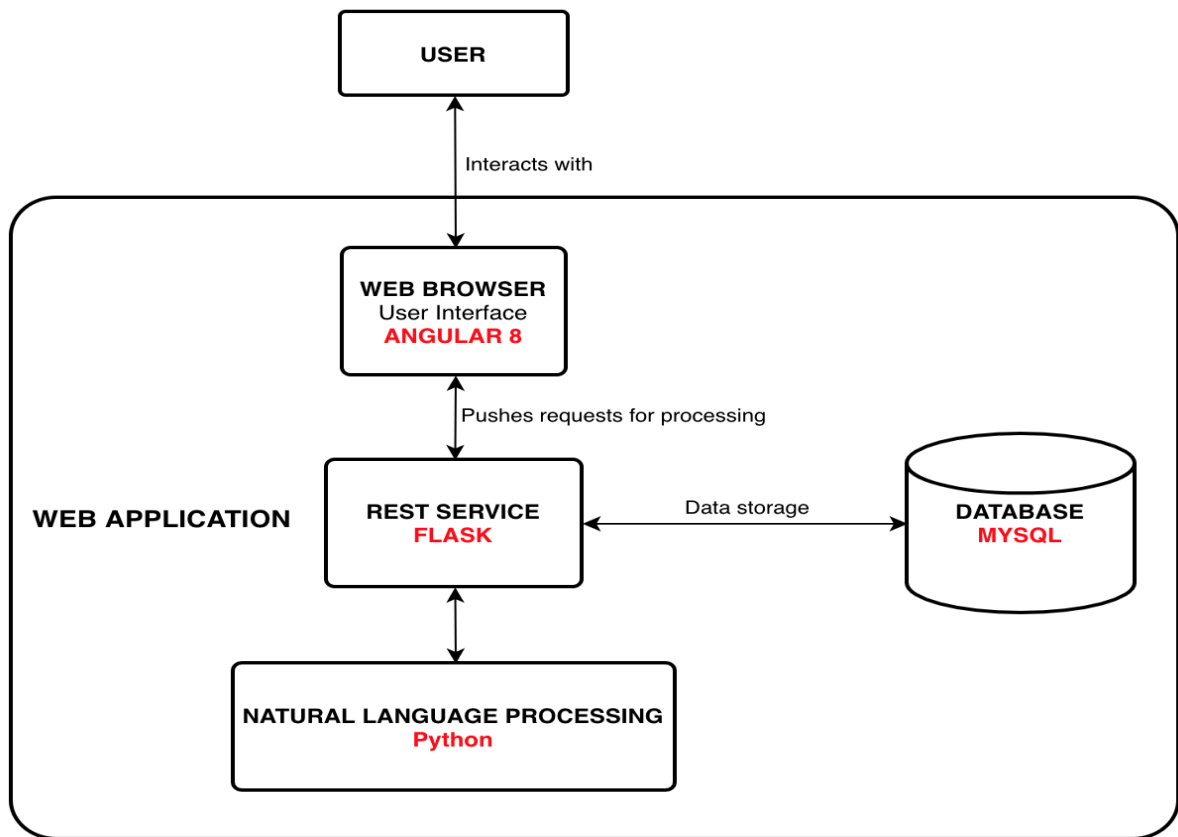
Figure 4.6 ACVSR System Architecture Diagram

4.7 Conclusion

Use case, sequence diagrams and the activity diagram are essential for the development of the ACVSR system as they help guide the scope of the project. The diagrams can also be used when testing the system. The ACVSR system should be able to allow the user to relate to the system in the way described by the system engineering illustrated in the above diagrams. If the Use case, Sequence diagrams and the Activity diagram are prepared properly the researcher will then be guided accordingly when the system is implemented. Following the RAD approach, rapid construction becomes the main task in the next chapter of this research. Phase 3 of the RAD methodology takes the prototypes and beta systems from the design phase and converts them into

the working model. Rapid construction is coupled with implementation and testing of the ACVSR

system in the next chapter.

CHAPTER 5

5 Implementation and Testing

5.1 Introduction

Implementation and testing marks the final phase of the RAD methodology being used to develop the ACVSR software project. Implementation in Rapid Application development is done in iterations. The objectives of this phase are to complete the detailed design and develop the system. At this phase the software is developed based on the objectives defined in the project definition phase. Development tools such as Visual Studio Code will be utilised in order to realise the ACVSR system. Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and MacOS (Sole, 2018). Visual Studio Code offers features support for debugging, syntax highlighting, intelligent code completion and code refactoring among others. Web browsers like Safari and Chrome will be particularly useful in debugging and testing as the ACVSR system is being developed.

5.2 Development Environment

Development environment set-ups vary from developer to developer. In most cases the development environment is influenced by developer preferences and more often the budget allocated for the tools to be used for the system development. In the case of developing the ACVSR system the student will make use of free open source technologies and software. In terms of the hardware to be used, the student will make use of their personal laptop computer. The specifications of the laptop computer to be used are as follows:

MacBook Pro 2015

Operating System   : macOS Catalina

Processor   : 2.7 GHz Dual-Core Intel Core i5

Memory   : 8 GB 1867 MHz DDR3

Graphics   : Intel Iris Graphics 6100 1536 MB

Visual Studio Code (VS Code) is a free source code editor which is used by the student. VS Code is easy to use and includes support for debugging, syntax highlighting, intelligent code completion, and code refactoring. Web browsers like Safari and Chrome will be particularly useful in debugging and testing when hot reloading from VS Code.

Since ACVSR system is a web based application, VS Code becomes the best choice of a code editor as it is free and provides basic support for HTML programming. VS Code has syntax highlighting, smart completions with IntelliSense, and customizable formatting, which makes programming and debugging easier for the student. VS Code also offers suggestions for elements, tags, some values as defined in HTML5 and Angular. In VS Code the student can also work with embedded CSS and JavaScript. The student can also control which built-in code completion providers and also override these in user or workspace settings if one prefers not to see the corresponding suggestions.

5.3 Programming Languages

The development of the ACVSR system made use of a combination of programming, scripting and web frameworks. The Python programming language was used for programming the system's logic. The Python language was preferred because the student as the developer is more familiar with the language hence reducing the development time because little time would be needed for learning some of the language's features. Other technologies   used include HTML, CSS, JavaScript, Angular, NodeJS, Flask and MySQL for the database. The table 5.1 below summarises the technologies used to develop the ACVSR system.

Table 5.1 Summary of technologies used to develop the ACVSR system

	Type	Description	Purpose
HTML	Language	HTML is Hypertext Markup Language and is the standard markup language for documents designed to be displayed in a web browser.	Designing the web pages for the ACVSR system.
CSS	Language	CSS stands for Cascading Style Sheets and is a style sheet language used for describing the presentation of a document written in a markup language like HTML.	Styling the web pages.

JavaScript	Language	JavaScript is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm.	Developing some components of the web based system.
Angular	Platform	Angular is a platform for building mobile and desktop web applications.	
NodeJS	Framework	Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a web browser.	Runtime environment.
Flask	Framework	Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.	Provides with libraries, modules and tools to help build the Web-system.

SQL	Language	SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system. MySQL is an open-source relational database management system.	Database management.

5.4 The Graphical User Interface (GUI)

The GUI was developed using the HTML, AngularJS framework and CSS. HTML is a markup language, heavily utilized for creating web pages and web applications. For this project HTML was combined with Angular and CSS, and this has become a milestone for web development. HTML has one of the most powerful features that it can be embedded into programs that are written in a scripting language like JavaScript. In this case HTML is responsible for affecting the behaviour and content of web pages. Markup languages like HTML, their document presentation and their description is written in CSS. CSS defines styles for all the web pages of the system. It also defines the design, layout and variations in display for different devices and screen sizes when one opens the ACVSR system.

Figure 5.1 ACVSR System Home Page
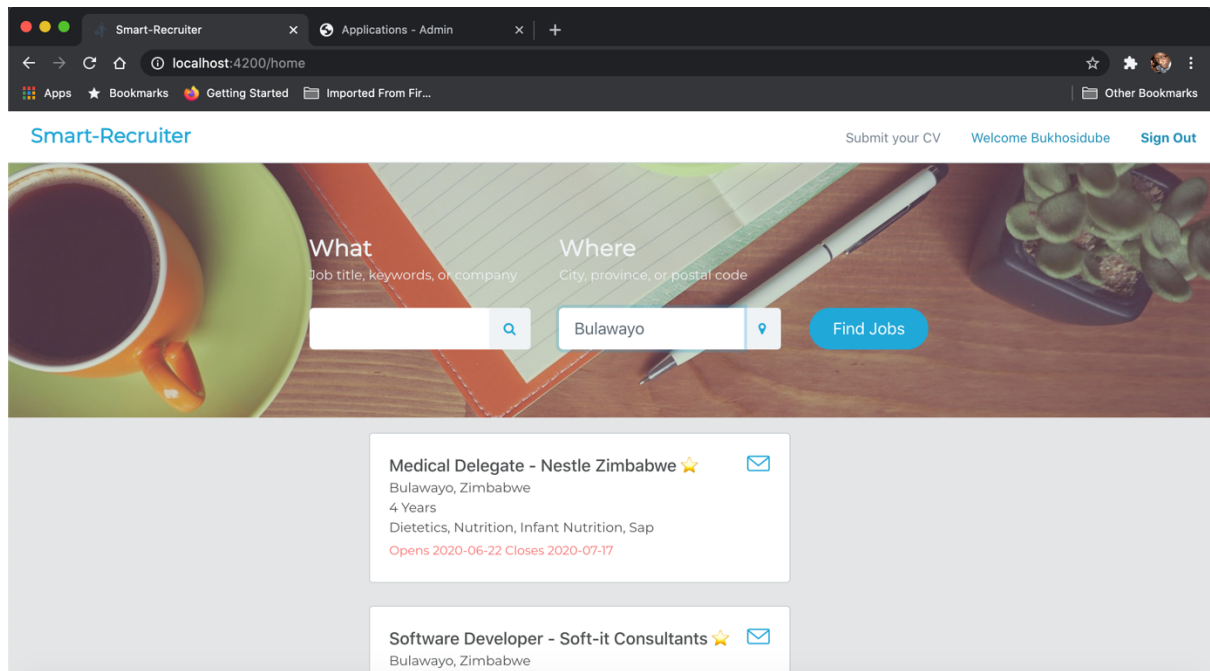
 The services of the ACVSR system are accessed upon one's registration in the system. A user first registers either as a job seeker or as an employer. The job seeker registers into the system and updates their profile. The employer, the one who is seeking talent, registers and plays a role of the administrator. Below is a screenshot of the registration page on the ACVSR system.

Figure 5.2 ACVSR System user registration page.

When users are registered in the system. They access the system by logging into the system with their preferred user credentials chosen upon their registration. The figure below shows the user login page.

Figure 5.3 ACVSR System user login screen

62

Below shows the page where the job seeker uploads their curriculum vitae information to keep their profile updated.
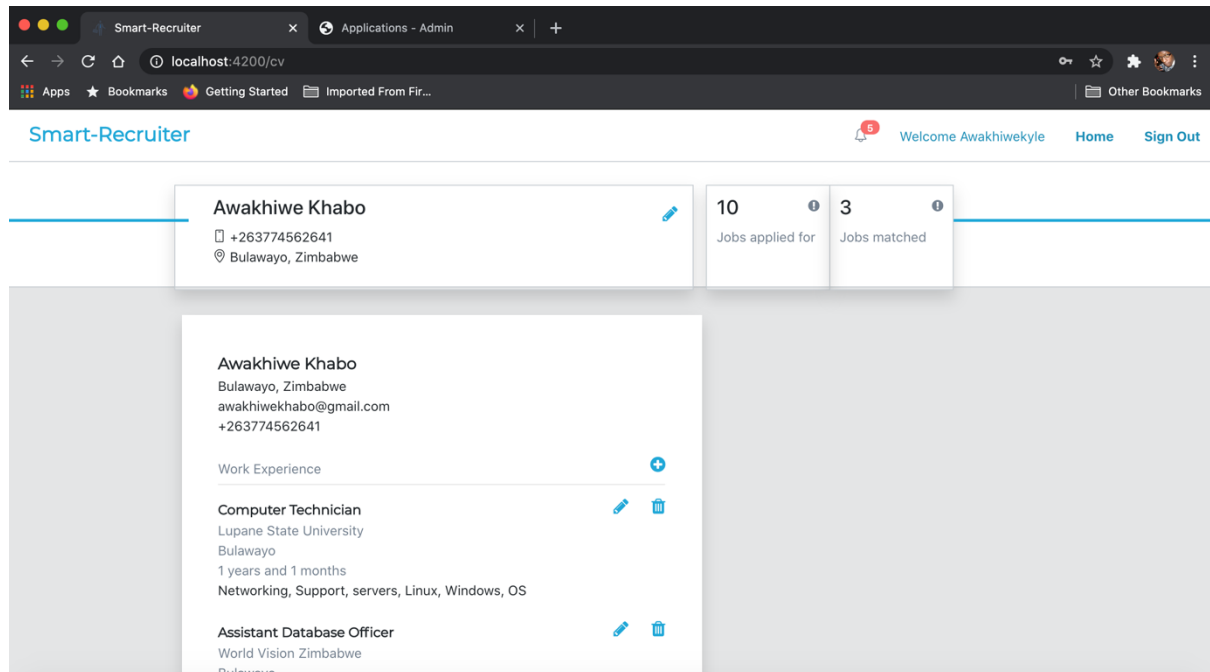
Figure 5.4 Curriculum vitae upload page

5.5 Testing

Software testing is intended to discover program defects and to find out whether the program does what it is intended to do (Sommerville, 2011). For the ACVSR system tests focused on verifying if all the functional and non-functional requirements were met. Testing was done based on the objectives of the ACVSR system. This process involved checking if the software has achieved the objectives proposed for the research. Development testing was implemented. This included all the testing activities that are carried out by a typical team development team. Development testing involved code debugging, locating problems in the code, referred to as defect testing. The researcher did have challenges during testing as there were many bugs during the development.

The researcher often used unit testing, where individual program units were tested. This approach was chosen because Unit testing focuses on functionality of objects or methods.

5.5.1   System Testing

System testing involved integrating different system components. The ACVSR system was then tested as a whole. System testing focused on testing component interactions. System testing ensured that components are compatible, interact correctly and transfer the right data at the right time across their interfaces. Computer Science department open day also gave the researcher a chance to test some of the system components with visitors as random users. For example, system components tested by random users during open day were that of registering a user and uploading a CV. This experience significantly helped the researcher to improve and perfect the aforementioned features.

Table 5.2 Test case for user registration

Test Objective	Test Condition	Input Specification	Expected Result	Pass/ Fail
User registration	User is currently on registration page	User enters their required registration details and clicks Register button	Registration successful and directs the user to the login page.	**PASS**

Table 5.3 Test case for navigation from Login to user dashboard page

Test Objective	Test Condition	Input Specification	Expected Result	Pass/ Fail
Navigation from Login to user dashboard page	User is currently on login page	User enters credentials and clicks Login button	Directs user to user profile page	**PASS**

Table 5.4 Test case for candidate uploading CV on their profile

Test Objective	Test Condition	Input Specification	Expected Result	Pass/ Fail
Candidate (User) uploading Curriculum Vitae on their profile.	User is currently on the upload CV page.	User fills in their CV details and clicks the save button.	CV upload successful and redirects the user to their profile page.	**PASS**

Table 5.5 Test case for Admin or Employer posting a job

Test Objective	Test Condition	Input Specification	Expected Result	Pass/ Fail
Admin or employer posting a job.	The employer is currently on the Admin dashboard page.	User enters all the necessary details of a job opening.	Job posting successful and redirects the user back to the jobs page.	**PASS**

Table 5.6 Test case for Employer viewing job applications

Test Objective	Test Condition	Input Specification	Expected Result	Pass/ Fail
Employer viewing job applications	The employer is currently on the Admin dashboard page.	The user searches for a specific job opening and clicks on the job to view applicants ranking.	List of applicants and their ranking is produced.	**PASS**

66

5.6 Conclusion

The development of the ACVSR system was done during the design phase. The design and implementation was divided into iterations since the researcher was using Rapid Application Development (RAD) as the software methodology. Testing was done based on the objectives of the ACVSR system. The researcher faced quite a number of difficulties implementing the ACVSR system because of amateur experience in the development of responsive web applications. Colleagues who were familiar with web development helped the researcher tackle some problems during the development.

CHAPTER 6

6 Conclusion and Recommendations

6.1 Introduction

The Automated Curriculum Vitae Screening and Recommendation system was implemented to address objectives which were derived from addressing the problems identified. During the analysis, design, implementation and testing, challenges were faced and solutions were found. Soon after system deployment, the system does undergo some changes at this stage when it is now operational. Some bugs and errors not discovered during testing can be uncovered after deployment and this aids the improvement of the system. User reviews can bring about changes of the interface or additional functionality. The changes are brought into effect through updates. In this chapter the researcher reviews challenges and limitations that were encountered. The chapter also includes recommendations that could be useful to any stakeholder who might be interested in the same project.

6.2 Review of Objectives

The aim of this research was to design and develop a system that screens a pool of candidates' CVs and recommends candidates according to their ranking based on the defined job profile requirements. The objectives for the ACVSR system are to:

- Capture job profile details.

- Upload candidates curriculum vitae.

- Screen uploaded CVs based on job profile requirements.

- Generate a report of recommended candidates.

At the time of documenting the Automated Curriculum Vitae Screening and Recommendation System, the admin user account was now able to capture job profile details when one is posting a job on the platform. The job seeker account was now able to fill in and upload curriculum vitae details. Most of the objectives had already been met and the researcher was still working on improving and perfecting the ACVSR system features.

6.3 Challenges and Limitations

The development of the ACVSR system was not as easy as the student had anticipated. The development of the system brought about quite a number of challenges for the student. One of the challenges faced during development of the system include learning natural language processing. Below follows some of the challenges faced by the researcher.

6.3.1 Angular

Angular as the platform used to build the web based application, caused a lot of problems for the student as it often caused the system not to respond and sometimes crash. The student was at amateur level and relied more on video tutorials and colleagues. The researcher solved this challenge by doing a lot of debugging though this cost the researcher a lot of time.

6.3.2 Natural language processing

Natural language processing was a research area on its own for the student. This came about to be a challenge as the student had to go through many research papers to understand what is, how it works and how it can be used. This again was time consuming and did cost the student a lot of time.

6.3.3   Natural language processing model

After the researcher had gone through numerous research papers they had to implement natural language processing by building an algorithm. This poised a lot of challenges as this was the first time for the researcher to develop an algorithm based Artificial Intelligence (AI) libraries. Some of the problems faced during building the NLP based algorithm for reading the CVs were extracting semantic meanings, setting the context and building the appropriate vocabulary. All these aspects were necessary for the algorithm to be able to read CVs, make meaning and screen them as the final stage. The researcher solved these problems by going through online tutorials, enquiring from colleagues and industry experts.

6.4 Recommendations

The researcher got recommendations on the open day specifically on uploading the CVs. The current approach to one uploading their CV on the ACVSR system is by filling in the CV information in a web form, formatted in a standard CV structure for uniformity purposes. However, this was viewed as an old way of doing things. The recommendation on this was that of allowing a user to upload their cv as a document file and the system just extracting data from these files. The researcher was informed of libraries that one can use to achieve this new approach. This was a good recommendation and could save users a lot of time. However, the researcher did not adopt this recommendation because of time and data mining being a research area on its own to embark on.

6.5 Conclusion

The ACVSR system fulfilled the objectives that were set at the beginning of the project. The system was able to allow the two system users being the admin to capture job profile details when one is posting a job on the platform and the job seeker to fill in and upload curriculum vitae details. The system was able to aid the recruitment process of screening CVs for recruiters and give results

of screened and rated CVs as recommendation. The ACVSR system will reduce nepotism and bias while creating transparency to both internal and external stakeholders interested in the hiring procedures. The ACVSR system will also improve the recruitment processes and help with selecting the right candidates for a particular job profile. This system will reduce the workload of the human resource department.

Scores of challenges were encountered in the process but thank you to the supervisor who was the guide the researcher throughout. Challenges lend a hand to the researcher to become a better programmer and problem solver. Through literature review the researcher even gained a deeper understanding of the system. Constructive critics and brainstorming helped the researcher shape the project over time and resulted in the project being a success.

Reference:

"Facebook, Pandora Lead Rise of Recommendation Engines - TIME". TIME.com. 27 May 2010. Retrieved 1 June 2015.

Agarwal, R., Prasad, J., Tanniru, M. and Lynch, J., 2000. Risks of rapid application development. Communications of the ACM, 43(11es), p.1.

Ahmad, T., Iqbal, J., Ashraf, A., Truscan, D. and Porres, I., 2019. Model-based testing using UML activity diagrams: A systematic mapping study. Computer Science Review, 33, pp.98-112.

Beynon-Davies, P., Carne, C., Mackay, H. and Tudhope, D., 1999. Rapid application development (RAD): an empirical review. European Journal of Information Systems, 8(3), pp.211-223.

Çano, E. and Morisio, M., 2017. Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis, 21(6), pp.1487-1524.

Catherine, Rose & Visweswariah, Karthik & Chenthamarakshan, Vijil & Kambhatla, Nanda. (2010). PROSPECT: A system for screening candidates for recruitment. 659-668. 10.1145/1871437.1871523.

Choudhary, D. and Kumar, V., 2011. Software testing. Journal of Computational Simulation and Modeling, 1(1), p.1.

De Gemmis, M., Lops, P., Musto, C., Narducci, F. and Semeraro, G., 2015. Semantics-aware content-based recommender systems. In Recommender Systems Handbook (pp. 119-159). Springer, Boston, MA.

Del Sole, A., 2018. Visual Studio Code Distilled: Evolved Code Editing for Windows, macOS, and Linux. Apress.

Dennis, A., Wixom, B.H. and Roth, R.M., 2018. Systems analysis and design. John wiley & sons.

Ekstrand, M.D., Riedl, J.T. and Konstan, J.A., 2011. Collaborative filtering recommender systems. Foundations and Trends® in Human–Computer Interaction, 4(2), pp.81-173.

Elio, R., Hoover, J., Nikolaidis, I., Salavatipour, M., Stewart, L. and Wong, K., 2011. About computing science research methodology.

Faliagka, E., Tsakalidis, A. and Tzimas, G., 2012. An integrated e-recruitment system for automated personality mining and applicant ranking. Internet research, 22(5), pp.551-568.

Färber, F., Weitzel, T. and Keim, T., 2003. An automated recommendation approach to selection in personnel recruitment. AMCIS 2003 proceedings, p.302.

Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, 2011, pp. 1-35

Friedman, K., 2003. Theory construction in design research: criteria: approaches, and methods. Design studies, 24(6), pp.507-522.

ft n il Ionel, N., 2008. Critical analysys of the Scrum project management methodology. Garlan, D., 2003, September. Formal modeling and analysis of software architecture: Components, connectors, and events. In the International School on Formal Methods for the Design of Computer, Communication and Software Systems (pp. 1-24). Springer, Berlin, Heidelberg.

Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T., 2004. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), pp.5-53.

Hewson, C. and Stewart, D.W., 2014. Internet research methods. Wiley StatsRef: Statistics reference online, pp.1-6.

Intelligent Data Analysis, vol. 21, no. 6, pp. 1487-1524, 2017

Jamshed, S., 2014. Qualitative research method-interviewing and observation. Journal of basic and clinical pharmacy, 5(4), p.87.

Johnson, C., 2006. What is research in computing science. Computer Science Dept., Glasgow University. Electronic resource: http://www. dcs. gla. ac. uk/~ johnson/teaching/research_skills/research. ht ml.

Konstan, J.A. and Riedl, J., 2012. Recommender systems: from algorithms to user experience. User modeling and user-adapted interaction, 22(1-2), pp.101-123.

Kumaran, V.S. and Sankar, A., 2013. Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT). International Journal of Metadata, Semantics and Ontologies, 8(1), pp.56-64.

Lethbridge, T.C., Sim, S.E. and Singer, J., 2005. Studying software engineers: Data collection techniques for software field studies. Empirical software engineering, 10(3), pp.311-341.

Lu, J., Wu, D., Mao, M., Wang, W. and Zhang, G., 2015. Recommender system application developments: a survey. Decision Support Systems, 74, pp.12-32.

Mackay, H., Carne, C., Beynon-Davies, P. and Tudhope, D., 2000. Reconfiguring the user: using rapid application development. Social studies of science, 30(5), pp.737-757.

Mahalakshmi, M. and Sundararajan, M., 2013. Traditional SDLC Vs Scrum Methodology–A Comparative Study. International Journal of Emerging Technology and Advanced Engineering, 3(6), pp.192-196.

Marshall, C., Brereton, P. and Kitchenham, B., 2015, April. Tools to support systematic reviews in software engineering: a cross-domain survey using semi-structured interviews. In Proceedings of the 19th international conference on evaluation and assessment in software engineering (pp. 1-6).

Robertson, I.T. and Smith, M., 2001. Personnel selection. Journal of occupational and Organizational psychology, 74(4), pp.441-472.

Rover, D., Ullerich, C., Scheel, R., Wegter, J. and Whipple, C., 2014, October. Advantages of agile methodologies for software and product development in a capstone design project. In 2014 IEEE Frontiers in Education Conference (FIE) Proceedings (pp. 1-9). IEEE.

Schafer, J.B., Frankowski, D., Herlocker, J. and Sen, S., 2007. Collaborative filtering recommender systems. In The adaptive web (pp. 291-324). Springer, Berlin, Heidelberg.

Senthil Kumaran, V. and Sankar, A. (2013) 'Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT)', Int. J. Metadata, Semantics and Ontologies, Vol. 8, No. 1, pp.56–64.

Sharma, L. and Gera, A., 2013. A survey of recommendation system: Research challenges. International Journal of Engineering Trends and Technology (IJETT), 4(5), pp.1989-1992.

Singh, A., Rose, C., Visweswariah, K., Chenthamarakshan, V. and Kambhatla, N., 2010, October. PROSPECT: a system for screening candidates for recruitment. In Proceedings of the 19th ACM international conference on Information and knowledge management (pp. 659-668).

So, S. and Witter, S., 2016. The evolution of human resources for health policies in post-conflict Cambodia: findings from key informant interviews and document reviews. Liverpool: ReBUILD Consortium.

Sommerville, I., 2011. Software engineering 9th Edition. ISBN-10, 137035152.

Tapsai, C., Meesad, P. and Haruechaiyasak, C., 2016. TLS-ART: Thai Language Segmentation by Automatic Ranking Trie. In 9th International Conference Autonomous Systems.

Van Meteren, R. and Van Someren, M., 2000, May. Using content-based filtering for recommendation. In Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop (pp. 47-56).

Wu, E., Michael, A.K., Oh, B.S., Uehara, S. and O'donnell, K.S., Microsoft Corp, 2012. Fuzzy matching and scoring based on direct alignment. U.S. Patent Application 12/961,878.

Appendix

Natural Language Processing algorithm for screening curriculum vitae

```python
def application_score_calc(username, job_id):
    import jieba
    import gensim
    from gensim import corpora

    skills = Skills.query.filter_by(username = username).first()

    # Getting candidate information from their profile
    skills = skills.skill
    education = Education.query.filter_by(username = username).first()
    education = education.degree + ", " + education.fieldOfStudy
    experience = Experience.query.filter_by(username = username).first()
    experience = experience.jobtitle + ", " + experience.description

    # Loading candidate information into an array for comparison
    texts = [skills, education, experience]
    texts = [jieba.lcut(text) for text in texts]
    dictionary = corpora.Dictionary(texts)
    keyword = Job.query.filter_by(id = job_id).first()
    keyword = keyword.job_title + keyword.experience_needed + keyword.skills_needed +
keyword.education_needed
    kw_vector = dictionary.doc2bow(jieba.lcut(keyword))
    feature_cnt = len(dictionary.token2id)
    corpus = [dictionary.doc2bow(text) for text in texts]
    tfidf = gensim.models.TfidfModel(corpus)
    index = gensim.similarities.SparseMatrixSimilarity(tfidf[corpus], num_features =
feature_cnt)

    sim = index[tfidf[kw_vector]]

    # Printing Scores for Skill, Education and Experience for user
    print('Skill:' + str(sim[0]))
    print('Education:' + str(sim[1]))
    print('Experience:' + str(sim[2]))

    #Return total final score for user
    return sim[0] + sim[1] + sim[2]
```