Rekommendationssystem för rekrytering inom ett pedagogiskt sammanhang

Recommendation systems for recruitment within an educational context

Gustaf Lagerqvist
Anton Stålhandske

Information Architecture 180hp
Bachelor Thesis in Computer and Information Science 15hp
Spring 2021
Supervisor: Dipak Surie
Examiner: Magnus Johnsson
Final seminar date: 05-31-2021

Abstract

Alongside the evolution of the recruitment process, different types of recommendation systems have been developed. The purpose of this study is to investigate recommendation systems within educational contexts, successful implementations of recommendation system architecture patterns, and alternatives to previous experience when evaluating candidates. The study is conducted through two separate methods; A literature review with a qualitative approach and design science research methodology focused on design and development, demonstration and evaluation. The literature review shows that, for recommendation systems, a layered architecture built within a microservice ecosystem is successfully utilized and has multiple beneficial aspects such as improved scalability, maintainability and security. Through design science research methodology, this study shows a suggested approach to implementing a layered architecture in combination with KNN and hybrid filtering. To avoid the lapse of suitable candidates, caused by demanding previous experience, this study shows an alternative approach to recruitment, within an educational context, through the use of soft skills. Within the study, this approach is successfully used to evaluate and compare students, but the same approach could possibly be applied to evaluate and compare companies. Moving forward, this study could be further expanded by looking into possible biases arising as a result of using AI and choices made during this study, as well as weighting of student-attributes.

Keywords: Recommendation systems, Artificial intelligence, Machine learning, Recruitment process, System architecture, Soft skills, Education

Tables of Content

Terms and Definitions

Human resources (HR)
Human resources can be described as the department of an organization that handles everything that involves or is related to employees [1].

Artificial intelligence (AI)
Artificial intelligence refers to the automation of a process through an intelligent agent's rational acting [2].

Machine learning (ML)
Machine learning is a subset of artificial intelligence in which the processes use predefined sets of information to build patterns and draw conclusions from previous behavior [3].

Recommendation system (RS)
Recommendation systems refer to software tools built to produce recommendations to a user of items that they might be interested in [4].

Anchoring bias
Anchoring bias involves shaping or influencing subsequent information as a consequence of previously consumed information [5].

Confirmation bias
Confirmation bias involves the seeking out of information that confirms initial judgments, positive or negative [5].

Similarity bias
Similarity bias involves unconsciously favoring similar people independent of whether the similarities are positive or not [5].

Demographic bias
Demographic bias involves discriminatory behavior based on the applicants country of origin [6]

Information overload
“Information overload, in the context of the Internet, refers to the overwhelming amount of information available to the web users” [7].

Classification
Classification is the action of assigning a specific class value to an entity based on certain attributes [8].

Clustering
Clustering is a form of unsupervised grouping that differs from classification in that it does not use predefined model data to create sets of coherent data. Instead, it creates unlabeled datasets. [8]

K-nearest neighbors (KNN)
K-nearest neighbours is a supervised classification algorithm that calculates the distance between one data point and multiple other data points where k signifies the number of neighbors wanted [9].

K-means
K-means is an unsupervised clustering algorithm that tries to partition a set of points into K sets (clusters) [10].

1. Introduction and Background

Personality traits and soft skills are, according to Gangula et al. [11] key aspects when it comes to learning new skills. G.M. Jaradat similarly claims that soft skills are deemed essential within the workplace [12]. Despite them being beneficial and help a person stand out among job seekers, well-developed soft skills are scarce in the corporate world [11].

Due to the digitalization of the recruitment process, the outreach is now far wider than it was before the step into the digital format. This has, in turn, generated an enormous increase in the number of applicants that apply for each open position which has made the process of screening each of these candidates a more or less impossible task. As an example, Google received roughly two million applications for about 14,500 jobs in 2017 [5]. With the rapid evolution of digital recruitment and its corresponding tools, the need for precision when it comes to matchmaking is evident [5].

This thesis work sprung from a conceptual idea of an online educational and networking platform. The purpose of the platform is to function as a forum for people to learn and improve abilities instead of, and, as a complement to traditional schooling. One way to achieve this is by matching students that want to practice a certain skill, with live projects run by companies. Because the purpose of the platform is to promote learning, previous experience is less relevant compared to a willingness to learn, having the right personality, and other soft skills.

Traditionally the screening and recommendation processes within recruitment have a focus on previous work experience and work history of the applicants [13]. Knowledge of existing recommendation systems architectures could prove beneficial to the case of the educational domain. However, using recommendation systems built for recruitment purposes while moving away from the traditional approaches and towards the educational domain could mean that suitable candidates are discarded.

1.1. Purpose & Research Question

The purpose of this study is to investigate recommendation systems within educational contexts, successful implementations of recommendation system architecture patterns, and alternatives to previous experience when evaluating candidates. Based on the purpose, the following research question was formulated:

Primary research question: *How to design a recommendation system for recruitment within an educational context?*

Second research question: *What architectural patterns are successfully used in similar contexts?*

Third research question: *How can a recommendation system evaluate a candidate without using previous experience?*

1.2. Structure

The following thesis work aims to answer the research question through the investigation of related research together with the implementation of a suggested design. Initially, a theoretical foundation regarding the field of study and important concepts is presented. Further on, the study presents information regarding what architectural pattern is most suitable for a web-based recommendation system, along with other findings relevant to the scope of the study. The section thereafter, presents a way to implement these findings as a suggested answer to the defined research question. The following section of the thesis includes a discussion of the results gained during the study in a general sense, along with the limitations of the study. Lastly, it contains a conclusion of the study's contribution to the field and suggested further research.

1.3. Theoretical Foundation

This section aims to provide insight into existing knowledge within the field of AI-enabled recruitment, as well as an explanation of central concepts relevant to this research.

Digitalization of the recruitment process has its origin in the mid-to-late 1990s. During the early years of the 1990s, recruitment was still an analog process of printing job openings in newspapers as a means of reaching as many candidates as possible, and relying on job referrals to get as much detail about a candidate as possible [5]. With the introduction of personal computing and the internet, during the later part of the 20th century, sharing of public and personal information reached new heights [14]. A possibility that could now be exploited, was the implementation of web-based system architectures [15]. These systems had the advantage of being portable and could be accessed, independent of geographical location, using a browser [15]. While it paved the way for a multitude of possibilities, it also created new challenges. People now gained access to an overwhelming amount of information in what has been described as "information overload" [14, 16]. Recruiting which at large, started to value reach over richness started to evolve its processes alongside the global digital transformation [5]. Due to its potential outreach, compared to non-digital means of recruitment advertising, the number of applicants increased significantly. According to J.S. Black et al. at the end of, what they call digital recruitment 2.0, online job listings received an average of 250 applications [5]. Digital tools enabled recruiters to conduct communication and management of potential candidates in a partially automated way [2]. However, the overflow of information required filtering of said information from both the candidate's and recruiters' side, to find suitable matches [5].

During the filtering and screening process recruiters spend on average six seconds doing initial scanning of job applications according to an "eye-tracking study" conducted by TheLadders [13]. The time spent during the scanning process, according to DR. J.S involves looking at four specific points or factors. These factors include "which area, job titles, companies you worked at, start/end dates, education" [17]. Research, however, has shown that from the perspective of job seekers, the factors that matter the most when choosing to accept or reject a position during an early screening process, are culture and work environment [5]. According to J.S Black et al.

research has shown that the human recruitment process also involves multitudes of cognitive biases including “anchoring bias, confirmation bias, and similarity bias.” These would, according to the research, prevent them from conducting a fair screening process [5].

Automation of the recruitment process can be achieved through the use of artificial intelligence (AI) for both employers and job seekers [2]. While AI usage is most noticeable within larger companies working within the field of technology [18, 2], the potential of AI in the application and hiring process is just beginning to be exploited [2]. A study was conducted in 2019 by E. Albert [18] to establish an understanding of the usage of AI within the field of recruitment and selection. The study findings showed that there are eleven areas within recruitment and selection where AI is applicable [18]. Most notable is the usage within areas such as chatbots and automatic screening software [18]. However, Laurim et al. [2] state that AI used in recruitment can be divided into three types. The choice of type to use depends on whether the applicants' or the recruiters’ process is to be automated. The first type matches an applicant's profile with job opportunities and ranks said opportunities for the applicant. The second type helps the recruiter by matching job opportunities with profiles of available applicants. The third, also working in the recruiters’ favor, ranks analyzed resumes by applicants’ skills.

While automation of a recruiter’s everyday tasks makes recruitment easier, there are still many questions about the place of AI within the process and whether enough trust can be established amongst the users for it to have a significant impact [2]. The secrecy regarding how companies use AI creates skepticism and doubt in terms of the fairness of the recruitment process [2, 19]. As the use of AI in recruitment grows, studies are being done to mitigate unfair treatment and the risk of biases. An example of such a study was conducted by Deshpande et al. [6] resulting in a reduction of demographic bias.

Recommendation systems and machine learning (ML) play an important part in our everyday digital lives and have been developed as a means to mitigate information overload [7]. As mentioned in the previous section there are ways the recruitment process can be automated. Two of which are automated- and self-learning recommendation systems. There are generally three types of techniques to implement such systems. These are content-based filtering, collaborative filtering, and hybrid systems that utilize a combination of both [9, 7]. According to Maruf et al. [9], content-based filtering utilizes data such as attributes and features to match entities based on their similarities. This technique can be used to implement the methods described by Laurim et al. [2], mentioned in the previous section. It does not exploit data such as user feedback or ratings. collaborative filtering, however, utilizes feedback and ratings from the user without exploiting features or attributes. Collaborative filtering is often described as having a higher accuracy in predictive results [9, 20]. This type of filtering can have a user-based approach that analyses the similarity between two users to recommend what similar users have previously consumed [21]. A third approach is a combination of the aforementioned approaches and is therefore called hybrid filtering. By using a combination of the previously mentioned methods, a better recommendation can generally be created [20]. In 2013 a proposal for a hybrid type of recommender system was presented by React Group, EPFL at the 22nd international conference on World Wide Web companion. This proposal also confirmed that the hybrid version which included both aspects of content-based and collaborative- or interaction-based filtering, outperformed systems that only relies on one of the two [22], which is also suggested by Verma et al. [20].

Some of these recommendation systems utilize machine learning, which, according to Tailor et al. [23], can be split into two practical headings. The first being “as a means of engineering rule-based software (for example in “expert systems”) from sample cases volunteered interactively” which is often referred to as unsupervised learning, and the second “as a method of data analysis whereby rule-structured classifiers for predicting the classes of newly sampled cases are obtained from a “training set” of pre-classified cases” commonly known as supervised learning [23, 24].

Classification, which is a form of supervised learning, is the action of using supervised machine learning to assign a specific class value to an entity based on similarities of certain attributes. It is thereby done through observation of pre-existing model data and content-based filtering [23]. Predictions can be produced based on a given class where the result indicates what should be recommended based on its related entities’ previous recommendations [4]. K-nearest neighbor is an example of such a classification (or regression) algorithm that uses the classification of the K nearest points to determine the classification of a point. It is considered supervised because it classifies a point based on the knowledge of the classification of other points [24]. Clustering is a form of unsupervised grouping that differs from classification in that it does not use predefined model data [8]. The goal of clustering is to produce groups of data, and assign labels, by comparing it to other non grouped data [8]. K-means is an example of a clustering algorithm that tries to partition a set of points into K sets (clusters) such that the points in the cluster are near each other. It is unsupervised in that the points have no predefined classification [10].

Systems architectures should, according to Sommerville, follow specific patterns where said patterns can be formulated as abstract descriptions for good practices when structuring software [15]. When describing architecture patterns for web-based systems Sommerville focuses on three types of patterns [15]. The first being model-view-controller (MVC), where the pattern is described as the basis of how interactions are managed within web-based systems [15]. The second architecture pattern is layered architecture (LA) which, like the MVC pattern, utilizes separation of concern but by organizing its components into layers. The third pattern is microservice architecture (MA) where the system is arranged as a collection of multiple, loosely coupled services [15]. Implementation of recommendation system architectures demands specific requirements, depending on the context of where it is used [25]. For example, within the context of education, recommendation systems can be built to take advantage of sets of pedagogical rules such as “gradually reduce the amount of orientation” [25]. In the case of Linkedin’s talent search and recommendation system, the architecture has been developed to serve the specific purpose of matchmaking between recruiters and viable candidates by utilizing both item-user relevance as well as user interest [26].

Soft skills can be used as a complement to professional knowledge to measure performance in the workplace [27]. Some soft skills have even shown a correlation to academic success in an online environment [27]. Two methods to map these soft skills are by using Dr. M.Belbin’s nine personality types [27], and the big five personality traits [28]. The different roles and their associated strengths as presented by Belbin are “resource investigator”, “teamworker”, “co-ordinator”, “plant”, “monitor evaluator”, “specialist”, “shaper”, “implementer” and “completer finisher” [27]. J.P. Oliver et al [28] describes the five personality traits as “extraversion or surgency”, “agreeableness”, “conscientiousness”, “emotional stability versus neuroticism” and “intellect or openness”. Both theories describe each role as having different key traits and strengths [27, 28]. These can be used as metrics to establish a well-functioning workplace and/or team [27, 28]. As described by L. Gutiérrez et al [27], Belbin’s roles and traits are used as measurable metrics to determine their individual correlation to engineering students’ capacity of learning.

2. Literature Review

According to Dr. Jennifer Rowley and Dr. Frances Slack, a subject area in which a study is conducted could offer related research which the researchers of the new study should take advantage of to conduct informed research [29]. Through the theoretical foundation, this study identified multiple possibilities of suitable architectural patterns for recommendation systems. The main purpose of this section was thereby to investigate the advantages and disadvantages of using MVC, LA, and MA in a big data- and recommender system context. A secondary purpose was to gain a broader knowledge of the other key concepts mentioned in the theoretical foundation.

2.1. Method discussion

Implementation of an architecture requires a level of understanding of how the architectural patterns are used within the systems. Therefore a qualitative study was decided upon. Investigation of quantitative and statistical relations between the chosen architectures would not provide the necessary level of knowledge. It would, however, possibly have provided the study with information regarding the frequency of the chosen architectures. Unfortunately, the frequency would not provide insight into how successfully or unsuccessfully, or in what way the architecture had been implemented, and would therefore not provide the study with much value other than showing popularity or unpopularity. To decide between a systematic- and a literature review, the methods were compared as per G. Basu's [30] description. As shown in table 1, a systematic review would have provided the reviewer and reader with deeper knowledge. It would also have eliminated biases more thoroughly, partly because of its extensive use of all available databases. A systematic review would however have required a timeframe that was not available, due to its much more labor-intensive requirements. These facts should be kept in mind as the reader interprets the content of the review and its result.

Table 1 shows the difference between conducting a literature review and a systematic review according to G. Basu's [30] description.

	Systematic Review	**Literature Review**
Definition	High-level overview of primary research on a focused question that identifies, selects, synthesizes, and appraises all high quality research evidence relevant to that question.	Qualitatively summarizes evidence on a topic using informal or subjective methods to collect and interpret studies.
Goals	Answer a focused question. Eliminate bias.	Provide summary or overview of topic.
Question	Clearly defined and answerable question. Recommend using PICO as a guide.	Can be a general topic or a specific question.
Components	Pre-specified eligibility criteria. Systematic search strategy. Assessment of the validity of findings. Interpretation and presentation of results. Reference list.	Introduction. Methods. Discussion. Conclusion. Reference list.
Number of Authors	Three or more to avoid bias.	One or more.
Timeline	Months to years. Average eighteen months.	Weeks to months.
Requirements	Thorough knowledge of the topic. Perform searches of all relevant databases. Statistical analysis resources (for meta-analysis).	Understanding of the topic. Perform searches of one or more databases.
Value	Connects practicing clinicians to high quality evidence. Supports evidence-based practice.	Provides a summary of literature on a topic.

2.2. Method

The information sources used in this study consist of three online databases; ACM, Google Scholar, and Libsearch. These were used to get a wide enough search range without sacrificing the specificity of the search scope. Locating information resources was done by formulating suitable queries regarding the function and functionality of different types of recommendation systems, the use of metrics similar to soft skills as comparatives between objects, the use of filtering and other algorithms, as well as other findings deemed relevant to the research and implementation of our system, within the context of the chosen architectures. Based on this, the keywords and key phrases used in the queries were; "recommendation systems", "recommender systems", "system architecture", "web-based" "model view controller", "MVC", "microservice architecture", "layered architecture", "collaborative filtering", "content-based filtering", "hybrid filtering", "artificial intelligence", "AI", "machine learning", "ML", "classification", "clustering", "soft skills" and "education". These were also used in the construction of a conceptual mind map, see figure 1.

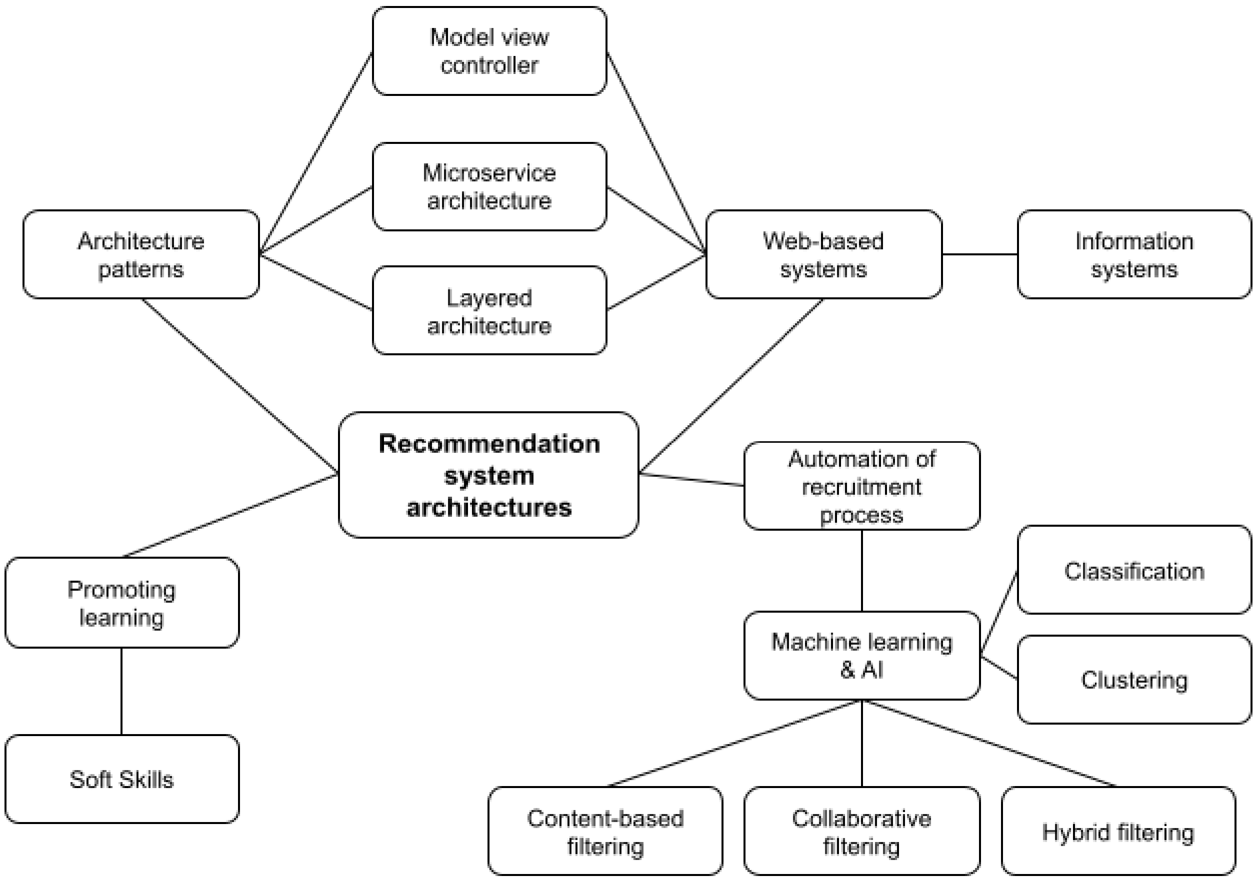

Figure 1 shows a conceptual mind map of the relationships between the key concepts of this study as interpreted by the researchers.

The gathering of information resources was limited to less than 15-year-old publications in the form of research articles/papers, dissertations, scientific reviews, and surveys. The search queries were fed into the three chosen databases, starting with broad terms and then further and further specified until the result consisted of less than 25 sources. These resources were evaluated by their title and abstract, and if deemed relevant, chosen for further evaluation. Once roughly 50 resources had been gathered, these were again subjected to evaluation based on abstract and the 15 most relevant were selected for a review of their full text. The 15 resources were then subjected to a relevance checklist to select fve texts to specifically base the choice of architecture on. The relevance was decided on the following criteria:

- Mentions MVC, LA, or MA in combination with big data or recommendation system context.
- Contains a description of the use of either MVC, LA, or MA.
- Compares and discusses MVC, LA, and MA.

Table 2 shows the information resources produced and analyzed by their full text during the literature review.

Title	**Author**	**Source type**	**Year**	**Keywords**
An Architecture for Developing Educational Recommender Systems	M. Bustos-López, R. Vásquez-Ramírez, G. Alor-Hernández	Research article	2015	Recommender systems, Educational application, Collaborative filtering
Cognitive Similarity-Based Collaborative Filtering Recommendation System	L.V. Nguyen, M. Hong, J.J.Hung, B. Sohn	Research article	2020	Cognitive similarity, Recommendation system, Collaborative filtering
Recommendation Semantic of Services in Smart City	C. Benfares, Y. El Bouzekri El Idrissi, A. Abouabdellah	Research article	2017	Smart services, Smart-city, Big data, Recommender system, Web semantic
OurPlaces: Cross-Cultural Crowdsourcing Platform for Location Recommendation Services	L.V. Nguyen, J.J. Jung, M. Hwang	Research article	2020	Recommendation systems, Crowdsourcing platform, Cognitive similarity
Scalable Dynamic User Preferences for Recommender Systems through the use of the Well-founded Semantics	M. Ilic, J. Leite, M. Slota	Research paper	2008	N/A
A Meta Learning Approach for Automating Model Selection in Big Data Environments using Microservice and Container Virtualization Technologies	S. Shahoud, H. Khalloof, M. Winter, C. Duepmeier, V. Hagenmeyer	Research paper	2020	Meta learning, Machine learning, Microservice, Web-based applications, Big data
ZenDen - A Personalized House Searching Application	K. Milkovich, S. Shirur, P.K. Desai, L. Manjunath, W. Wu	Research paper	2020	Mobile housing app, Recommender systems, Microservice architecture, CNN
A Task-oriented English Education Platform Powered by ICT&AI	Y. Liu, J, Zhao	Research article	2019	Task-oriented, Pragmatic competence, Resource integration, Microservice
How smart is e-tourism? A systematic review of smart tourism recommendation system applying data management	R.A. Hamid, A.S. Albahri, J.K. Alwan, Z.T. Al-qaysi, O.S. Albahri, A.A Zaidan, A. Alnoor, A.H. Almoodi, B.B. Zaidan	Review article	2020	E-tourism, Smart tourism, Tourism recommender systems, TRS
Synapse: A Microservices Architecture for Heterogeneous-Database Web Applications	N. Viennot, M. Lecuyer, J. Bell, R. Geambasu, J. Nieh	Research article	2015	N/A

Personalised Web Search For E-Learning Using Group-Based Recommendation Approach	M.M. Rahman	Dissertation	2019	N/A
A Résumé Evaluation System Based on Text Mining	Y. Chou, C. Chao, H. Yu	Research paper	2019	Artificial intelligence (AI), big data, text mining, web crawler
Using Recommender System for Matching Students with Suitable Specialization: An Exploratory Study at King Abdulaziz University	K. Alshaikt, N. Bahurmuz, O. Torabah, S. Alzahrani, Z. Alshingiti, M. Meccawy	Short Paper	2021	Artificial intelligence, Education, Machine learning, Recommender system
Software architectures for big data: a systematic literature review	C. Avci, B. Tekinerdogan, I.N. Athanasiadis	Scientific review	2020	Big data, Software architecture, Systematic literature review
APR: Architectural Pattern Recommender	S. Sharma, B. Sodhi	Research Paper	2017	Search-based software engineering, Architectural patterns, Recommender system, Textual entailment, Stackoverflow

Summary of the Five articles chosen for architectural evaluation

The resources chosen for architectural analysis were; *“Software architectures for big data: a systematic literature review”, “Synapse: A Microservices Architecture for Heterogeneous-Database Web Applications”, “A Task-oriented English Education Platform Powered by ICT&AI”, “Cognitive Similarity-Based Collaborative Filtering Recommendation System” & “An Architecture for Developing Educational Recommender Systems”.*

“Synapse: A Microservices Architecture for Heterogeneous-Database Web Applications”, begins with the acknowledgment that the MVC pattern has a history of being widely used in regards to the implementation of web applications. The researchers then state that there are limiting factors to using frameworks and the development of MVC-structured web applications. These factors are related to the strong connection between user interaction and system response, where controllers operate within the context of a user session. To avoid phenomena such as race conditions in operations related to the persistence layer, frameworks that utilize the MVC pattern, apply object relational mappers (ORM) where each entity stored in a database is mapped to a specified object model within the application. This implies that entities stored and managed by the model can be subjected to mutation locks which can decrease system performance and usability. Furthermore, limiting factors include a lack of scalability of the MVC pattern and the inability of following heterogeneous-database solutions where a unified query interface is presented to the user. [31]

“A Task-oriented English Education Platform Powered by ICT&AI” is a research paper where a student-centered educational platform is developed following a microservice

architecture. The system is built using four layers, a user layer, an access layer, a service layer, and a resource layer. The user layer represents the frontend web applications which concern user interactions. The access layer concerns routing of traffic and access to specific services and other third-party solutions. The service layer is specific to its area of concern, for example, a user service responsible for hosting a k-nearest neighbor (KNN) algorithm using content-based filtering to recommend suitable classes to students. Finally, the resource layer handles the retrieval and manipulation of data. Each layer has a connection to an underlying service isolated from the others in a loosely coupled fashion. The research brings to light the benefits of structuring the system in a layered manner to provide higher levels of security and flexibility. This separation emphasizes separation of concern and brings a higher level of modularity, scalability, and flexibility. [32]

"Cognitive Similarity-Based Collaborative Filtering Recommendation System", published in applied science, proposes a recommendation system utilizing nearest neighbor (NN) and collaborative filtering to produce recommendations based on cognitive similarity and user feedback, in a multi-layered architecture for the calculations. The overall system architecture is designed using an MVC pattern, built with Java, and utilizing an Apache Tomcat server and a MySQL relational database for storage. The researchers acknowledge that the most important aspect of their product, OurMovieSimilarity (OMS) system, is the interaction with the end-user to gather cognitive data from the user. This requires a simplistic and easy-to-use interface with which the user will interact. The internal layers of the recommendation system are; item layer, cognition layer, and user layer. The item layer consists of nodes representing items and relations (edges) representing the similarity between items measured by cosine similarity. The cognition layer holds a set of cognitive similarity groups which are derived based on movie item features (t=title, g=genre, d=director, a=actor, p=plot) and connected to users through collaborative preference data. The user layer holds nodes representing users and relationships through numerous kinds of cognitive similarities. [33]

"An Architecture for Developing Educational Recommender Systems" is a research article in which an educational recommender system is proposed and presented. The researchers present "a generic architecture for developing educational recommender systems". The architecture is structured in a layered fashion and consists of four layers; a presentation layer, a service layer, a data access layer, and a data layer. Their presentation layer is constructed using HTML5, JavaScript, and cascading style sheets (CSS) and used as an interaction layer for the end-user. The service layer consists of a set of APIs which allow the production of educational resource recommendations. The data access layer holds a data extractor which is responsible for inserts, updates, deletions, and queries to the data layer. Finally, the data layer is responsible for storing data and information regarding educational resources and information regarding different sources of information where educational resources might be found. The research mentions five issues that a recommendation system must address, namely; *knowledge acquisition* which is used to gather information about a user from which a profile can be produced. *Domain knowledge* is used to categorize items and create order within the system. *Profile representation* through vectors in a vector-space model allows for easy application of machine-learning techniques used to construct recommendations. *The domain* itself will contain information to be

recommended and *recommendation techniques* including rule filters, machine learning, and collaborative filtering. [25]

"Software architectures for big data: a systematic literature review" is a scientific review of 43 studies within the topic of software architecture for big data systems. A systematic review was applied within the research and findings showed that there are five main architecture patterns applied; layered architecture, cloud-based architecture, service oriented architecture, hybrid architecture, and multi-agent architecture. An overview of the reviewed studies shows that 30 out of the 43 studies reviewed used a layered architecture. It also shows that 23 architecture implementations have some part of their structure online as per the Cloud-computing architecture. Cloud-computing can be used to a varying degree as part of the implementation of other architectural patterns. [10]

2.3. Results

Table 3 shows advantages and disadvantages found regarding MA, MVC and LA during the review of the 5 chosen texts.

Text	**Pattern**	**Positive aspects**	**Negative Aspects**
"Synapse: A Microservices Architecture for Heterogeneous-Database Web Applications"	MA	Highly scalable. Suitable for large systems with a high user load.	N/A
"A Task-oriented English Education Platform Powered by ICT&AI"	MA	High level of modularity, scalability, and flexibility for full-scale systems. Easy to maintain.	N/A
"Cognitive Similarity-Based Collaborative Filtering Recommendation System"	MA	N/A	N/A
"An Architecture for Developing Educational Recommender Systems"	MA	N/A	N/A
"Software architectures for big data: a systematic literature review"	MA	Somewhat frequently used. Can be used with cloud computing. Scalable and suitable for large sets of data.	N/A
"Synapse: A Microservices Architecture for	LA	N/A	N/A

Heterogeneous-Databas e Web Applications”			
“A Task-oriented English Education Platform Powered by ICT&AI”	LA	High levels of security and flexibility. Easy to maintain. A user interface can easily be applied and exchanged.	N/A
“Cognitive Similarity-Based Collaborative Filtering Recommendation System”	LA	Clear structure through the layered structure of recommendation calculations	N/A
“An Architecture for Developing Educational Recommender Systems”	LA	Easy to organize. Scalable and easily maintained. Distributed responsibility and loosely coupled.	N/A
“Software architectures for big data: a systematic literature review”	LA	Frequently used. Cleaner manufacturing and maintenance. Can be used with cloud computing.	N/A
“Synapse: A Microservices Architecture for Heterogeneous-Databas e Web Applications”	MVC	A widely-used pattern within web-based systems.	Focuses on user session context. Less scalable.
“A Task-oriented English Education Platform Powered by ICT&AI”	MVC	N/A	N/A
“Cognitive Similarity-Based Collaborative Filtering Recommendation System”	MVC	Enables implementation of a user interface within the system.	N/A
“An Architecture for Developing Educational Recommender Systems”	MVC	N/A	N/A
“Software architectures for big data: a systematic literature review”	MVC	Can be used with cloud computing.	N/A

MA is, as shown in table 3, implemented in three out of five reviewed resources. It is used by Viennot et al [31] to address the inherent weaknesses of MVC. The product they present is used to restructure complex MVC-built web applications into MA structures. Viennot et al [31] conclude that the MA structure allows for highly scalable architectures that suit large systems with a high user load. Liu et al. [32] also bring up scalability as one of the factors for them partly using a MA, along with its maintainability, modularity, and suitability for full-scale systems. Avci et al [10] present several studies successfully using variants of MA architectures. This is used as a means to handle large sets of data efficiently in the sense of collecting, storing, and sending the data [10]. It is also mentioned as a scalable system architecture that supports the addition of new services to existing products to extend their purpose, such as the addition of data mining applications to analytic workflow web services [10]. The study also shows the use of MA combined with cloud computing [10].

LA is implemented by four out of five reviewed resources as shown in table 3. It is successfully used by Liu et al [32] as a system architecture for their product, which is part of a larger system built as a MA. Their study praises the flexibility, maintainability, and security of a layered architecture. It also displays a diagram of their product, showing that the user interface is built as a separate layer [32]. Nguyen et al [33] describe their three-layered architecture as being a clear structure to their recommendation system that also provides scalability. Bustos-López et al [25] provides similar positive thoughts to the scalability and maintainability factor of a LA and describes it as being easy to organize. They also mention the benefit of the loose coupling of the parts of a layered architecture and the ability to distribute responsibility over different layers of the system [25]. Avci et al [10] bring up the fact that 30 out of 43 resources reviewed by them, use a layered architecture in some sense. They describe the layered pattern as providing a cleanly manufactured product due to its clear structure and as being an architecture that is easy to maintain [10]. It also has the benefit of being able to be used in combination with cloud computing [10].

MVC is not implemented by any of the reviewed resources but analyzed or used as a foundation by three out of five as shown in table 3. It is according to Viennot et al [31] widely used among web-based systems. They also go on to describe it as having inherent weaknesses such as being less scalable than for example MA and having a user session-focused context making it a slightly more rigid pattern [31]. When used for analysis by Nguyen et al [33] it does, however, provide easy implementation of a user interface within the system. It is also a pattern that allows for cloud computing according to the studies done by Avci et al [10].

Other findings showed that neighboring calculations were used to cluster or classify objects in seven out of the 15 texts [32, 33, 25, 10, 34, 35, 36]. Four texts specified the type of neighboring calculations that are used. Out of those, three [32, 33, 25] use a form of KNN and one [10] use K-means. Furthermore, Nguyen et al. [33] present a recommendation system that utilizes cognitive similarities between users based on collaborative data collected from the users. The study shows that a cognitive profile can be established based on user preferences and opinions. This is also discussed by Bustos-López et al. [25] where recommendation systems are utilized to establish the cognitive states of students and recommend items based on personal interest and preferences.

2.4. Analysis & Conclusion

The findings presented in 2.3 shows that the MVC pattern has been a common and well-renowned way of structuring web-based applications. However, two out of five studies suggest that adding a layered or microservice architecture to the use of MVC provides an increased level of flexibility and scalability. The findings also show that four out of five studies reviewed, use a layered architecture within the context of recommendation systems and big data.

The literature study suggests that a common and beneficial approach to the structure of large-scale systems involves combinations of multiple architecture patterns. Most common is the implementation of a layered architecture for specified subsystems and structuring these in a higher level architecture using a microservice pattern. Five out of five studies suggest using either a layered- and/or a microservice pattern. Both of these approaches provide higher levels of maintainability, flexibility, scalability, and modularity. Individually a microservice architecture adds the benefit of being suitable for handling large amounts of data, and a layered architecture adds security. By dividing the recommendation system into three or four layers, the system is clearly and cleanly organized into strict areas of concern. Structuring the system in a layered pattern allows easy addition or removal of layers. As the work of Liu et al [32] shows, the user interface can be built separately thanks to its loose coupling with the rest of the architecture. meaning that it can be built earlier or later from the remaining logic of the recommendation system.

Other important findings derived from the literature reviewed showed that clustering and classification algorithms are frequently used and that KNN is a suitable option to implement for the scope of proof of concepts due to its frequent mentions, detailed description, and simplicity of implementation. This is however specific to supervised learning and requires existing model data. For unsupervised learning and clustering, the research shows that K-means can be used as an alternative. Furthermore, the literature review revealed insights into the use of soft skills such as personality-based attributes and personal preferences in recommendation systems context. Such attributes can be formulated through different means. For example, in the scope of this study, the use of Belbin-based attributes has been selected as a suitable option in combination with work-specific preference values provided by the student. Combining a layered architecture with the ability to utilize these values and attributes was therefore considered as the main objective of the design science method.

3. Design and implementation

The main purpose for conducting this method was to implement and demonstrate the findings presented in section two, namely a layered recommendation system architecture capable of utilizing quantified soft skills. The second purpose was to investigate the effects of filtering based on specific preference attributes in combination with matchmaking based on soft skills. The final purpose was to evaluate how adequately the implementation supports an answer to the research question. Due to the nature of the purpose, this section investigates the findings derived from the literature review with an abductive approach.

3.1. Method discussion

This study limits the scope of design science (DS) methods to design and creation and design science research methodology as these approaches are good examples of how to conduct design science within the development of information systems [37, 38, 39], and suit the needs of this study presented in table 4.

According to Vaishnavi et al. design and creation is a method that contains five steps where each step naturally leads to the next [37]. These steps are; "*awareness of the problem*" where the main goal is to produce a formal or informal proposal that is intended for use in further research. "*suggestion*" where the proposal is dissected, examined, and used as a tool within a creative approach to generating a suggested solution to the problem. "*development*" where the design derived from the suggestion step is explored, developed, and further implemented. "*evaluation*" where the said artifact is evaluated against the proposal. "*conclusion*" where the results of the research are wrapped up. The outcome of the evaluated artifact is deemed either as passable due to its alignment with the expected result and is therefore solidified as facts and/or knowledge that the researcher(s) have gained. The steps could then be repeated iteratively for continuous improvement of the proposed design [37].

An alternative approach investigated was design science research methodology (DSRM). According to Bisandu [38], DSRM is a research method within the field of information technology where the goal is the innovation of problem-solving artifacts. A proposal produced by Peffers et al. [39] describes a general approach to DSRM which includes the following six activities; "*problem identification and motivation*" which is the activity of defining the problem for the specific research and using the said definition in the development of an artifact that provides a solution. "*definition of the objectives" for a solution* which utilizes the previous step to "infer objectives of a solution" [39]. "*design and development*" is where the objectives of a solution are exploited and the artifact is created. "*demonstration*" in which the produced artifact is used to demonstrate its ability to solve single or multiple instances of the formulated problem. "*evaluation*" in which observations and measurements are done on how adequately the artifact can support a solution to the problem. "*communication*" where the research team communicates the formulated problem and its relevance, the produced artifact, and its benefits and shortcomings. This can include design patterns, functionality, and effectiveness to other researchers [39].

Research of the two methods shows that there are clear similarities between them. Most notable are the similarities in the activities contained in each sub-section of the methods. Evaluation of the DS methods showed that DSRM had a higher focus on the demonstration of the produced artefact(s) compared to design and creation which did not include a clearly defined activity of that kind. Since both methods suited the needs of this study in high regard, either choice could be seen as a suitable approach. Therefore the deciding factor found during the evaluation was the activity regarding demonstration of the artifact(s) which was only included and outlined in DSRM. It could however be argued that the evaluation is questionable as the relevance mark that is set is subjective; this is however not regarded as having a significant negative impact on the validity of this study due to the similarity of the methods.

Table 4 presents the research needs from a DS method and a motivation as to why the requirement was relevant for this study. It is followed by a relevance mark between low and high where high indicates the strongest relevance to this study.

Description	**Motivation**	**Relevance**
Clear outline into how the method is implemented and used	For the method to yield good results the method should be described in a way that is well understood by the implementers.	High
The method includes an optional activity regarding the implementation of one or more artifacts	Since this study benefits from a practical way of testing the suggested architecture an activity regarding implementation of an artifact is positive.	Medium
Involves showcasing of artifacts	A practical presentation of a produced artifact’s solution to a problem can be used to strengthen the artifact’s relevance and the findings to the field.	High
Includes multiple iterations of activities	This study involves a literature review in which objectives regarding implementation of a suitable architecture pattern are presented; this requirement can be used for testing variations of that pattern but testing could also be done to compare objectives against one variation.	Medium

3.2. Design science research methodology

The problem-centered initiation and objective-centered solution are important approaches to the formulation of the purpose and need. In the case of this study, problem-centered initiation is examined and evaluated throughout the introduction in section 1.0. The objective-centered solution is the focus area of the literature review presented in section 2.0. The communication stage of DSRM is presented through this thesis discussion & conclusion. The implementation stage of this study focuses on design- and development, demonstration, and evaluation.

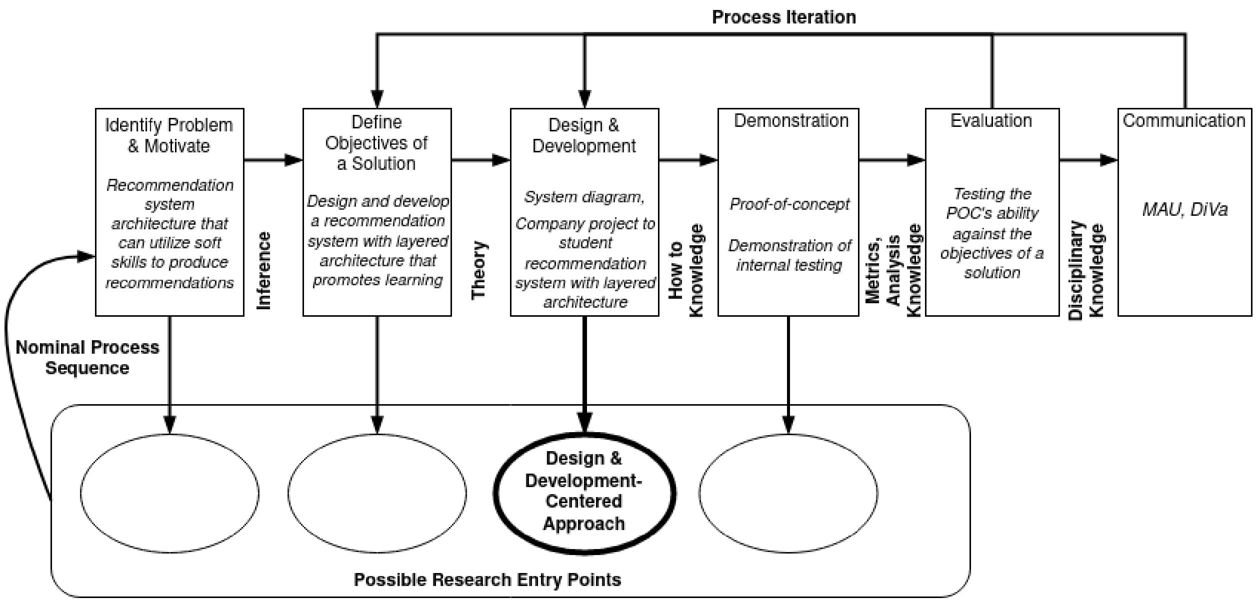

Figure 2 shows the DSRM process model as conducted in this study with performed activities displayed in square boxes and the selected research entry point marked with a bold circle.

Design and development was conducted based on the defined objectives for a solution. A DSRM process model, shown in figure 2. was produced to provide an overview of the research entry point, the activities performed, the process iteration, and the design process as a whole. The conclusion of the findings derived from the theoretical foundation and the conducted literature review showed that a layered architecture is a preferable pattern when implementing a recommendation system. They also suggested that the system could benefit from being built as a separate service in a microservice architecture. Therefore the aim of this research activity was the design and implementation of a proof of concept for a recommendation system with a layered architecture while considering compatibility within a microservice architecture. Included within this activity was the design of diagrams used to illustrate the architecture and execution flow of the recommendation service, see figures 3 and 4. The implemented architecture is structured into four layers;

- A network layer with defined endpoints used to communicate with the API through a representational state transfer (REST) using HTTP protocol and JSON format, used for portability and compatibility in a microservice architecture.
- A service layer holding multiple services responsible for specific areas of concern; *authorization and security*, *matchmaking and recommendations*, *student modeling*, *company modeling*, *project modeling*.
- A storage access layer including a repository class responsible for data insertion, extraction, and manipulation.
- A data layer including a database file using SQLite for storage of our machine learning models and training data.

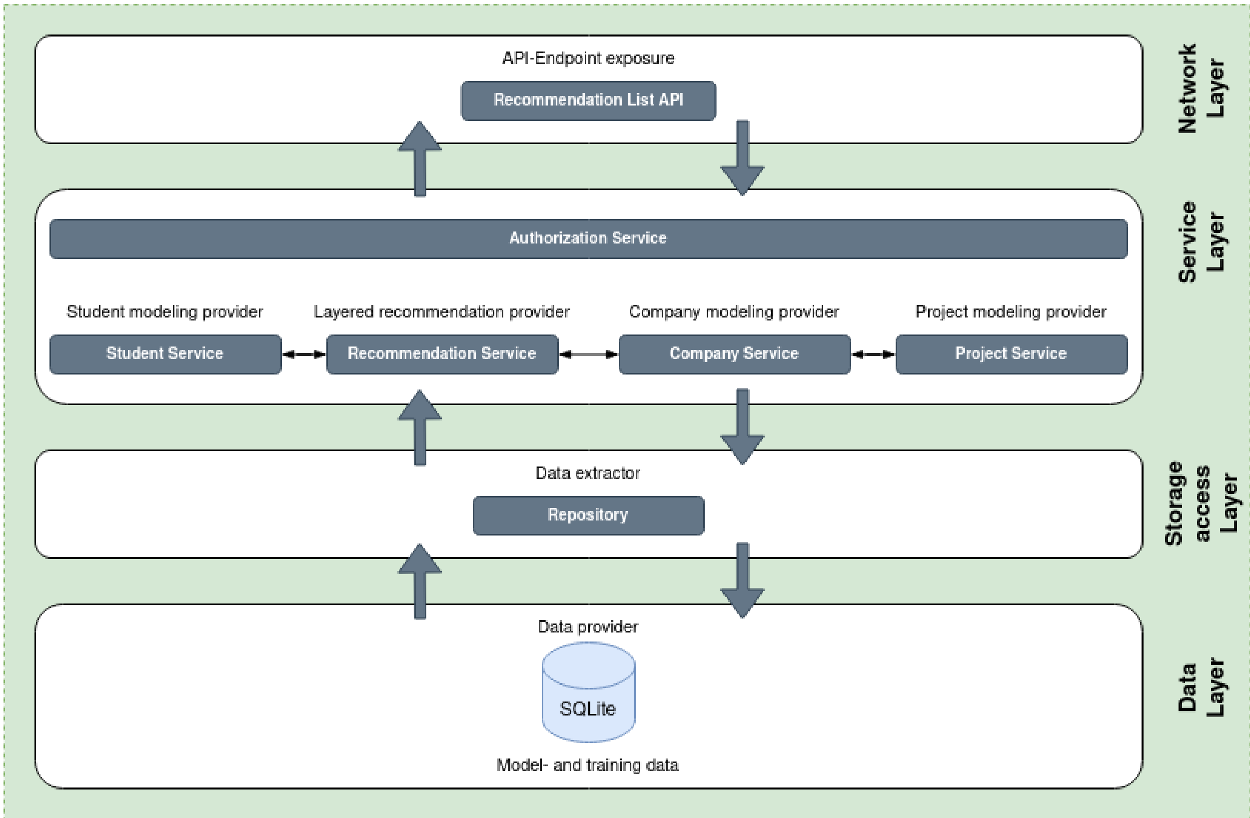

Figure 3 illustrates the four layers of the layered architecture pattern, their arrangement and structure, and the flow of communication through the recommendation system.

Regarding tools used for the implementation, this study limited the option of languages to three main candidates that had support for big data analytics, machine learning implementation, and support for API definitions. The top three candidates that met these requirements were *Java, Python,* and *Golang* [40, 41, 42]. These candidates were subjected to a support- and previous knowledge checklist as presented in table 5.

The boolean value representing previous experience was only set to a positive value if both of the researchers had previous knowledge of it. Once this was completed all attributes were summed up and the language with the highest total score count was the resulting language used during the implementation.

The selection of frameworks followed a similar comparative approach, presented in table 6, with the following requirements; *previous knowledge, lightweight and simple, support for REST, support for a multi-layered pattern.* This study limited the selection to three candidates specific to python; *Bottle, Flask,* and *Django*

Finally, storage solutions were selected through a similar comparative approach, presented in table 7, with the following requirements; *previous knowledge, file-based and modular, supports simple data entry and update, supports data retrieval.* This study limited the selection to two candidates identified through the specified requirements; *CSV* and *SQLite*

Table 5 shows each language candidate as they are compared against the formulated requirements followed by a total score.

Language	Previous knowledge	Supports big data analytics	Supports machine learning	Supports API definitions	Total score
Python	Yes	Yes	Yes	Yes	4
Java	No	Yes	Yes	Yes	3
Golang	No	Yes	Yes	Yes	3

Table 6 shows each framework candidate as they are compared against the formulated requirements followed by a total score.

Framework	Previous knowledge	Lightweight and simple	Supports REST	Supports multi-layered structure	Total score
Bottle	Yes	Yes	No	Yes	3
Flask	Yes	Yes	Yes	Yes	4
Django	No	No	Yes	Yes	2

Table 7 shows the two candidates for data storage as they are compared against the formulated requirements followed by a total score.

Data storage	Previous knowledge	File-based and modular	Supports simple data entry and update	Supports data retrieval	Total score
SQLite	Yes	Yes	Yes	Yes	4
CSV	No	Yes	No	Yes	2

Based on the theoretical foundation, hybrid filtering was applied. Attributes used for calculations of content-based filtering between students were written in the form of personality traits- and roles inspired by DR. M Belbin [27] that is presented in the theoretical foundation. This resulted in nine predefined roles applicable for classification and 36 quantifiable personality traits ranging in value from one to nine, see appendix A. For the sake of demonstration, these values are assumed to have been collected by self-evaluation, where each student rates how well the listed attribute corresponds with their personality. The collaborative data consists of a project history for each student and is used to make predictions. The attributes set for each student preference and project description were used as a weighting factor within the preference layer of the recommendation service. These attributes are; distance work, full time, part time, paid position, previous experience. These preference/description values were set as either true or false (written as one or zero in the database). Based on the findings from the literature review, K-nearest neighbor was used as a means to conduct a classification of students according to Belbin's predefined roles. To measure the distance between objects, needed partly for the KNN algorithm, euclidean distance was used but could be exchanged with manhattan distance or minkowski with similar results [9].

Demonstration of the implemented recommendation system architecture was conducted through internal testing with mock data and each test case was performed with the same mocked student entity, see appendix B. The demonstration includes a flow diagram (Figure 3) of the underlying recommendation service architecture.

The final order of execution within the internal recommendation service is presented in the text below as well as in figure 4.

- *Calculations of similarity scores between students* include three steps of a hybrid filtering method. The algorithm accepts a student objects' 36 values of attributes as parameters where each attribute holds a value between one and nine. These values are separately compared to the corresponding values of the nine predefined student archetypes using a KNN algorithm including calculations of distance metrics using euclidean distance. The algorithm produces a value between zero and 288 (8x36) where 288 represents the largest distance possible and zero represents the smallest distance possible. Dividing the total difference with the maximum possible difference (288) will produce a number representing the similarity of two student objects. Finally, the algorithm returns a list, sorted in ascending order with the closest student archetype holding index position zero.

- *Classification of students* is executed based on the absolute nearest student archetype derived from the similarity score calculations using KNN. This function also includes saving the classified student to the database together with distance values to the rest of the archetypes.

- *Finding projects that students with the same class have worked on* is performed by utilizing collaborative filtering and fetching the projects that students with the same classification have added to their watch list. Said work history would therefore be of interest to the newly classified student based on their similarity.

- *Running recalculation based on project occurrence* is performed to recalibrate the distance based on project occurrence within the recommendation list previously produced based on student object similarity.

- *Running euclidean distance calculations on students with the same assigned class* is performed to increase the accuracy of the produced recommendation list. The system utilizes the euclidean distance calculations in another iteration to find the most suitable project based on the similarity of each student object with the same class.

- *Matching project information with student preference* refers to the hard values defined in each project; paid position, part time, full time, experience needed, and remote work. These values are added to the student object sent to the recommendation system and used as weight factors within the matchmaking calculation to increase the accuracy of the match.

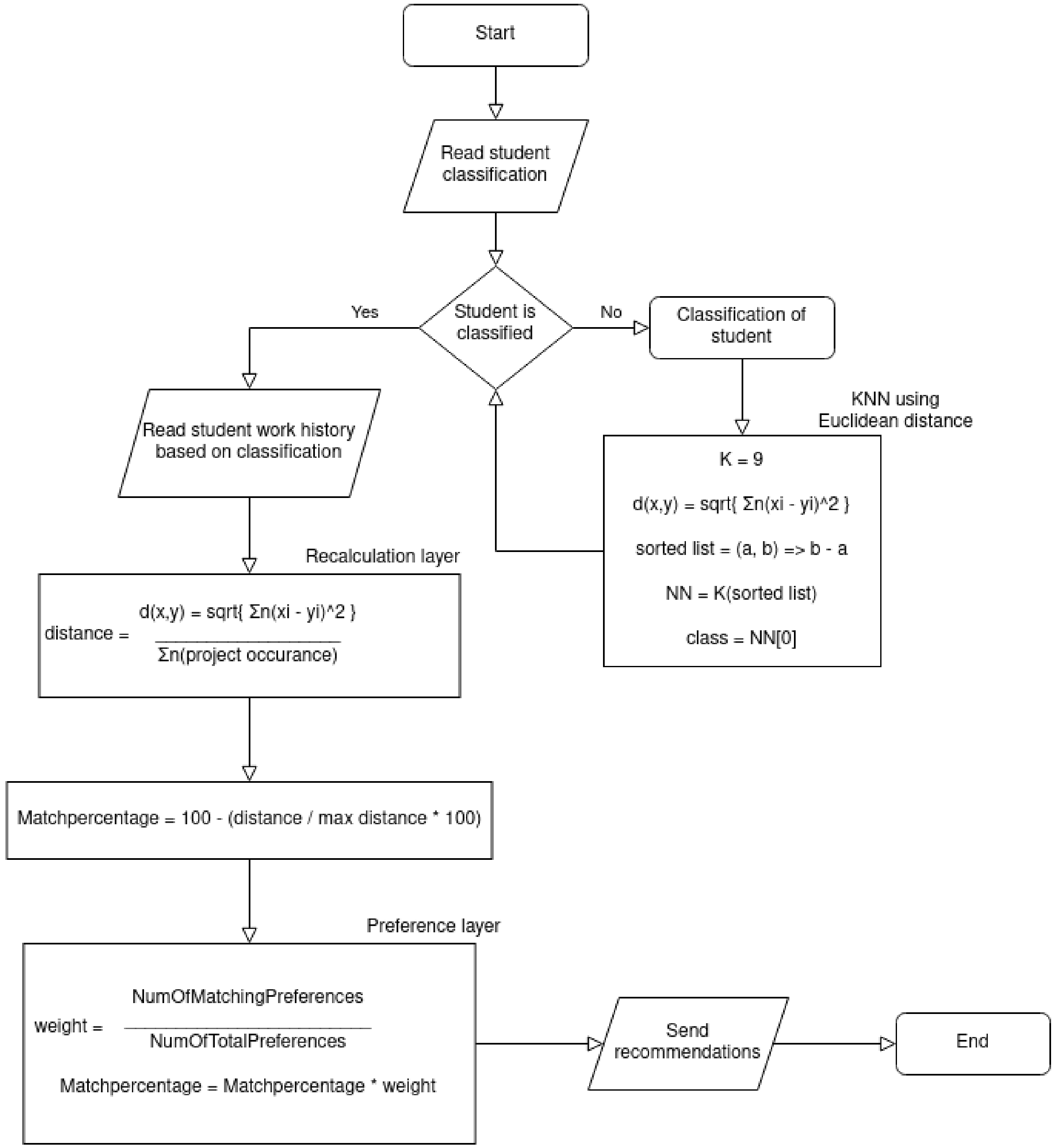

Figure 4 shows the final communication flow and calculations within the implemented recommendation service.

Test ID 1 - *Establishment of baseline accuracy and performance.*
This test was performed to establish a baseline of performance and accuracy. Performance is measured through the following calculation; s*et / nr*. Where s*et*=summed execution time and nr=number of runs. Accuracy is measured through comparison between the new student object and the student the project is found through. Only students that have been assigned the same class are taken into account.

Table 8 presents the performance of each execution and execution time derived from the baseline test.

											set
E-ID	1	2	3	4	5	6	7	8	9	10	
etpr	0.037 s	0.042 s	0.037 s	0.039 s	0.031 s	0.028 s	0.032 s	0.032 s	0.030 s	0.036 s	0.344 s

Table 9 shows a recommendation list produced for Student41 in the baseline test.

Assigned Role	Monitor evaluator					
ID	7	2	6	1	8	2
Name	Project7	Project2	Project6	Project1	Project8	Project2
Previous employees with the same class	Student38	Student38	Student37	Student37	Student30	Student30
Match	64.2%	64.2%	62.5%	62.5%	52.8%	52.8%

Result: The summed execution time of test 1 shown in table 8 was 0.344s. The Mean execution time was 0.344/10 = 0.0344s. The accuracy of the recommendation is shown in table 9. It directly reflects the similarity between the new student and the student the project is found through. Project2 is however shown two times with different percentages because two students with the same class have Project2 in their work history.

Test ID 2 - *Addition of recalculation layer*
This test was performed to compare performance and accuracy after adding a recalculation layer to the recommendation service where betpr=baseline execution time per run and etpr=execution time per run in test 2. The recalculation layer was added to prevent the same project from occurring multiple times. The second purpose of the layer was to produce a mean accuracy of the projects occurring multiple times.

Table 10 presents the performance of each execution and execution time derived from the baseline test and the compared execution time of each run in test 2.

											set
E-ID	1	2	3	4	5	6	7	8	9	10	
betpr	0.037 s	0.042 s	0.037 s	0.039 s	0.031 s	0.028 s	0.032 s	0.032 s	0.030 s	0.036 s	0.344 s
etpr	0.037 s	0.036 s	0.040 s	0.035 s	0.042 s	0.034 s	0.037 s	0.037 s	0.036 s	0.037 s	0.371 s

Table 11 shows a successful response of a recommendation list produced for Student41 compared to the result of the baseline match

Assigned Role	Monitor evaluator				
ID	7	6	1	8	2
Name	Project7	Project6	Project1	Project8	Project2
Previous employees with the same class	Student38	Student37	Student37	Student30	Student30 Student38
B-Match	64.2%	62.5%	62.5%	52.8%	52.8%
Match	64.2%	62.5%	62.5%	52.8%	58.5%

Result: The summed execution time of test 2 shown in table 10 was 0.371s. Mean execution time was 0.371/10 = 0.0371s which is 0,0027s slower than the baseline test. As table 11 shows, one of the occurrences of Project2 was removed. The mean accuracy of Project2 was 58.5%.

Test ID 3 - *Adding a preference layer to weight projects*
This iteration investigates performance and accuracy after adding a preference layer. The layer multiplies the accuracy from test 2 with the percentage of preferences that match the project's descriptive values. If for example only "distance work" and "paid position" match (two out of five preferences), the accuracy from test 2 is multiplied by 0.4 (40% of the preferences).

Table 12 shows the performance of each execution through execution time derived from the baseline test and the compared execution time of each run in test 2 and test 3.

											set
E-ID	1	2	3	4	5	6	7	8	9	10	
betpr	0.037 s	0.042 s	0.037 s	0.039 s	0.031 s	0.028 s	0.032 s	0.032 s	0.030 s	0.036 s	0.344 s
2etpr	0.037 s	0.036 s	0.040 s	0.035 s	0.042 s	0.034 s	0.037 s	0.037 s	0.036 s	0.037 s	0.371 s
etpr	0.046 s	0.043 s	0.038 s	0.051 s	0.043 s	0.034 s	0.052 s	0.030 s	0.041 s	0.034 s	0.412 s

Table 13 shows a successful response of a recommendation list produced for Student41 compared to the result of the baseline match and matches in test 2.

Assigned Role	Monitor evaluator				
ID	7	6	1	8	2
Name	Project7	Project6	Project1	Project8	Project2
Previous employees with the same class	Student38	Student37	Student37	Student30	Student30 Student38
B-Match	64.2%	62.5%	62.5%	52.8%	52.8%
2-Match	64.2%	62.5%	62.5%	52.8%	58.5%
Score	0.6	0.6	0.6	0.4	0.2
Match	38.5%	37.5%	37.5%	21.1%	11.7%

Result: The test showed a decrease in average performance; 0.0412s - 0.0371s = 0.0041s. While the accuracy increased, the match percentage showed an overall decrease.

Test ID 4 - *Filtering projects based on experience needed*
This test measures performance and accuracy after filtering out projects that require previous experience from the recommendation list.

Table 14 shows the performance of each execution through execution time derived from the baseline test and the compared execution time of each run in test 2, 3, and 4.

E-ID	1	2	3	4	5	6	7	8	9	10	set
betpr	0.037 s	0.042 s	0.037 s	0.039 s	0.031 s	0.028 s	0.032 s	0.032 s	0.030 s	0.036 s	0.344 s
2etpr	0.037 s	0.036 s	0.040 s	0.035 s	0.042 s	0.034 s	0.037 s	0.037 s	0.036 s	0.037 s	0.371 s
3etpr	0.046 s	0.043 s	0.038 s	0.051 s	0.043 s	0.034 s	0.052 s	0.030 s	0.041 s	0.034 s	0.412 s
etpr	0.042 s	0.036 s	0.034 s	0.051 s	0.041 s	0.034 s	0.049 s	0.039 s	0.050 s	0.033 s	0.409 s

Table 15 shows a successful response of a recommendation list produced for Student41.

Assigned Role	Monitor evaluator				
ID	7	6	1	8	2
Name	Project7	Project6	Project1	Project8	Project2
Previous employees with the same class	Student38	Student37	Student37	Student30	Student30 Student38
B-Match	64.2%	62.5%	62.5%	52.8%	52.8%
2-Match	64.2%	62.5%	62.5%	52.8%	58.5%
3-Match	38.5%	37.5%	37.5%	21.1%	11.7%
Score	0.6	0.6	0.6	N/A	0.2
Match	38.5%	37.5%	37.5%	N/A	11.7%

Result: The test showed an increase in average performance; 0.0412s - 0.0409s = 0.0003s. Adding a filter on the final preference value; *experience needed*, resulting in the removal of Project8 from the recommendation list. Project8 which had a match of 21.1% based on the personality of the student was filtered out since Student41 lacked the experience needed.

The evaluation activity of this study, following the description provided by Peffers et al [39], aimed at comparing the produced artifacts with the defined objectives for a solution. The comparisons focused on the functions, accuracy, and performance of the produced artifact. It also focused on how adequately the artifact supports a solution to the problem. The architecture has been designed and implemented using a layered architecture pattern which signifies its support for higher levels of flexibility, maintainability, and scalability. The artifact successfully communicates over HTTP-REST protocol and supports request-response which enables the artifact to be utilized by external systems and could thereby function as a part of a larger microservice ecosystem.

The tests and the results presented in the demonstration show that the first iteration of the artifact performed match calculations based on the similarities between new student objects and already classified students and their work history. However, the result of the first iteration contained duplicate projects. The results produced from the test with id two indicate that a recalculation was successful in that the duplicate project was removed and instead used to produce a mean accuracy based on occurrence. The addition of the recalculation layer resulted in a decrease in performance. This could however be considered as a false negative due to its relatively low difference from the baseline performance time. Another aspect to consider for the performance measurement is network latency. Since each test was performed utilizing the exposed API-endpoint in the network layer of the system and called from a separate subnetwork, network latency could be a factor. The third test utilized the addition of the preference layer and showed a slight increase in response time and a decrease in match percentage but an overall increase in accuracy due to the addition of weight calculation. The fourth and final test showed the effect of adding a hard filter on the project description value of previous experience. As an addition to using this preference value, set on each company project, as a weight factor, it was used to filter the recommendation list. If the student object lacked the boolean value the project was removed from the list resulting in Student41 not being recommended Project8, despite a match percentage of 21.1% based on the student's personality. This result could indicate that the use of previous experience as a hard filter could result in the company missing out on a potentially well-suited candidate if a project with an otherwise high match percentage to the student's personality included this value.

4. Discussion

A transformation of recruitment, from analog to digital has resulted in wider outreach and higher numbers of applicants per listed position. Automation through the implementation of recommendation systems is now used for matchmaking of applicants and jobs. This is an effective approach to the traditional recruitment process where work history and previous experience are promoted. Moving away from the traditional approaches of recruitment and towards an educational context, using recommendation systems built for those purposes could mean that suitable candidates are discarded. Therefore the purpose of this research was to investigate recommendation systems within educational contexts, successful implementations of recommendation system architecture patterns, and alternatives to previous experience when evaluating candidates. This purpose led to the following three questions;

Primary research question: How to design a recommendation system for recruitment within an educational context?

Second research question: What architectural patterns are successfully used in similar contexts?

Third research question: How can a recommendation system evaluate a candidate without using previous experience?

As presented in the literature review, studies showed that the MVC architecture pattern has a history of being widely used within web-based systems. However, studies within the context of recommendation systems show that aspects such as flexibility, maintainability, and scalability require complementary approaches. layered- and microservice architecture patterns have been conceptualized as a solution to meet these requirements. The findings in the review suggest that web-based systems of a larger scale can benefit from the modularity of a microservice architecture where each service is implemented using a layered architecture. Furthermore, the literature review showed that recommendation subsystems are often built as separate services and follow the structure of a layered architecture. These findings in combination with the context of the system being utilized by an educational platform led to the implementation shown in figure 3. The implementation presented in 3.2, shows that recommendations of suitable company projects can be produced based on a person's soft skills. These results indicate that the use of soft skills could prove useful in other recommendation systems. By using hybrid filtering, the system suggests projects that people with similar personalities (content-based filtering) have previously worked on (collaborative filtering). To compare personalities, this study uses the roles and skills suggested by Dr. M. Belbin. However further studies could show that comparisons of other soft skills could prove as effective. It should also be stated that the attributes carry the same weight. This means that they are being equally valued, which would likely not be the case when implementing the system for production. Through testing of the iterations of the recommendation service architecture, this study shows that the requirement of previous experience within a recommendation system could have negative impacts on the result of matchmaking and the potential loss of otherwise suitable candidates.

As stated in the discussion section of the literature review, a non-systematic approach was taken. It could be argued that conducting a systematic review would have provided this study with a wider range of resources which could have had a positive impact on the validity and reliability of this study. The study could also have benefited from more nuanced data. For example, focusing on resources presenting successful implementations of recommendation systems following certain architecture patterns might have yielded a false positive. By examinations of resources that present results of a negative nature this study could have gained a higher level of validity and reliability. Regarding the validity and reliability of the implementation method, this study limits the comparison of methods to design and creation and design science research methodology which could impact the implementation method as a whole. It should also be stated that, for the chosen method, the implementation and testing of alternative architecture patterns have been left out of this study. If included, this might have yielded more reliable and less interpretable results. Furthermore, comparisons of other forms of distance calculations are not included in this study and might yield different results in regards to the classification and matchmaking process. This however was not the focus of this study and could therefore be regarded as less relevant. In regards to the tests, the mocked data could also prove faulty as it might not accurately reflect real-world data. Furthermore, the amount of data is a limiting factor that affects the reliability and validity of the test results in a negative manner. With a higher volume of data, a different variance would most likely have been observed from which more generalizable patterns could have been extracted and examined.

Regarding how the data is formulated it could be argued that the interpretation of the roles described by Belbin is subjective and that other researchers could have extracted the attributes differently even if the same role description were to be analyzed. The question of weighting is also a factor that will impact how the attributes are used within the matchmaking calculations. This study utilizes the student's preferences as a weight factor for the overall match percentage. This should however be regarded as insufficient and the study would benefit from the addition of weighting each preference attribute as well as weighting the individual personality attributes. Finally, the fact that the study is not performed by domain experts should be kept in mind when interpreting the results of this study. The implemented recommendation service matches a student with a set of projects that similar students have previously worked at. It could however prove useful in providing employers with suitable candidates based on what projects other students have viewed. It could then be suitable to weight the relevance of the project in relation to how long similar students have looked at them, and not solely whether they have or not. Similar content-based filtering used to suggest projects to students could be useful in the reversed order, to recommend students to companies looking for employees.

5. Conclusion

How to design a recommendation system for recruitment within an educational context?

In an educational context, recommendation systems should not filter out students lacking relevant experience as this might result in the loss of otherwise well suited candidates. The recommendations could be successfully produced using hybrid filtering with student/company data for content-based filtering and student/company feedback for collaborative filtering. Furthermore the results in section 3.0 suggest more advanced data processing could provide the recommendations with higher specificity. The test done in this study, providing a higher specificity, does take longer. Whether there is a statistically significant correlation between execution time and the level of processing has not been established during this study.

What architectural patterns are successfully used in similar contexts?

The recommendation system delivering this type of matchmaking could benefit from being constructed with a layered architecture, as a service in a microservice architecture. These benefits include scalability, maintainability, and flexibility to the system. As presented in the results of the literature review these architecture patterns are successfully implemented in several of the analysed information resources. This result is further strengthened by the practical implementation and demonstration presented in this study.

How can a recommendation system evaluate a candidate without using previous experience?

As a substitute to previous experience, the recommendation system can match students with company projects using soft skills. These skills can be personality traits or other cognitive measurements. Examples include Belbin's personality traits, the big five personality traits, and cognitive profiles established through collaborative filtering. This type of data can be collected through self evaluation or external assessment.

6. Further research

The rapid evolution of technology means that our findings might not be as relevant in the near future. Because similar studies could provide different results, it is suggested that research regarding the construction and possibilities of recommendation systems in an educational context is done continuously. Future research could benefit from a more detailed and nuanced approach to valuing and weighting of attributes. Such findings could show what factors matter to what degree, to different parties. The investigation of new classes and/or clusters through unsupervised clustering algorithms, such as K-means, could provide more useful classes/clusters than those used in this study. Such studies should compare newfound material with the roles used in this study and present the most suitable data. Future studies would benefit from external testing towards potential users (both project owners and students) and note how the system is perceived. Research involving integration testing using this system as a microservice should be done to learn how well the system functions in a microservice architecture. Another aspect, touched upon briefly within the section of theoretical foundation, is bias within recruitment which should be further investigated to gather information about how this can be mitigated to a greater degree within AI driven recommendation systems.

References

[1] Human resources edu. Retrieved 2 May 2021 from: https://www.humanresourcesedu.org/what-is-human-resources/

[2] V. Laurim, S. Arpaci, B. Prommegger, H. Krcmar, Computer, Whom Should I Hire? – Acceptance Criteria for Artificial Intelligence in the Recruitment Process. *Proceedings of the 54th Hawaii International Conference on System Sciences,* 2021

[3] J. Wright, Dr. D. Atkinson, The impact of artificial intelligence within the recruitment industry: Defining a new way of recruiting, Carmichael Fisher, 2019

[4] K. Alshaikh, N. Bahurmuz, O. Torabah, S. Alzahrani, Z. Alshingiti, M. Meccawy, Using Recommender Systems for Matching Students with Suitable Specialization: An Exploratory Study at King Abdulaziz University. iJet, *International Journal of Emerging Technologies In Learning*

[5] J.S. Black, P. Van Esch, AI-enabled recruiting: What is it and how should a manager use it? *ScienceDirect*, Business Horizons 63, pp. 215-226, 2020

[6] K.V. Deshpande, S. Pan, J.R. Foulds, Mitigating Demographic Bias in AI-based Resume Filtering, *Session 5: Fairness in User Modeling, Adaptation and Personalization (FairUMAP 2020)*. UMAP '20 Adjunct, Genoa, Italy, July 14-17, 2020

[7] B. Gayle, X. Peng, W. Han, M. Kannan, S. Detmar, Development of an Instrument to Study the Use of Recommendation Systems, *AMCIS* Proceedings. 34, 2003

[8] R. Xu, D. Wunsch, Clustering, *IEEE Press Series on Computational intelligence*. WILEY A John Wiley & Sons, Inc., Publication, 2009.

[9] Z. R. Maruf and A. D. Laksito, The Comparison of Distance Measurement for Optimizing KNN Collaborative Filtering Recommender System. *2020 3rd International Conference on Information and Communications Technology (ICOIACT).* Yogyakarta, Indonesia, pp. 89-93, 2020

[10] C. Avci, B. Tekinerdogan, I.N. Athanasiadis, Software architectures for big data: a systematic literature review. *Big Data Analytics*, 2020

[11] Dr. R. Miryala, Dr. R. Aluvala, Trends, Challenges & Innovations in Management *Zenon Academic Publishing,* 2015

[12] G M. Jaradat, Internship training in computer science: Exploring student satisfaction levels. *Department of Computer Science,* Jerash University, 26150-311, Jerash, Jordan.

[13] TheLadders, Eye-Tracking Study, *Ladders* 2018. Retrieved 15 March 2021 from: https://www.theladders.com/static/images/basicSite/pdfs/TheLadders-EyeTracking-StudyC2.pdf

[14] L, Rosenfeld, P Morville, J Arango, Information Architecture - For the web and beyond, *Fourth Edition*. Sebastopol CA/United states of america: O'REILLY, 2015

[15] I. Sommerville, Software Engineering Tenth Edition, *Global Edition*. Hallbergmoos Germany, Pearson, 2018

[16] L.E. Janlert, Tänkande och beräkning - En inledning till datavetenskap och kognitionsvetenskap, Dimograf/Poland: Studentlitteratur AB, 2015

[17] DR. J Sullivan, Why You Can't Get A Job … Recruiting Explained By the Numbers. *ERE Recruiting Intelligence*

[18] E Albert. AI in talent acquisition: a review of AI-applications used in recruitment and selection. *On Another Note*

[19] J Ochmann, S Zilker, S Laumer, Job Seekers' Artificial Intelligence-related Black Box Concerns. *IT Professionals*, SIGMIS-CPR '20, June 19–21, Nuremberg, Germany, 2020

[20] P. Verma, S. Sharma, Artificial Intelligence based Recommendation System 2020 *2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2020

[21] P. Boström, M. Filipsson, Comparison of User Based and Item Based Collaborative Filtering Recommendation Services. *School of Computer Science and Communication (CSC), Master of Science in Engineering - Computer Science and Technology*, 2017

[22] Y Lu, S Helou, D Gillet, A Recommender System for Job Seeking and Recruiting Website. *Proceedings of the 22nd international conference on World Wide Web companion*

[23] C Tailor, D Michie, D.J Spiegelhalter, Machine Learning, Neural and Statistical Classification. *ResearchGate*

[24] R.F. de Mello, M.A Ponti, Machine Learning, A practical approach to the statistical learning theory". *Springer International Publishing AG*, part of Springer Nature, 2018

[25] M. Bustos-López, R. Vásquez-Ramírez, G. Alor-Hernández, An Architecture for Developing Educational Recommender Systems, *Instituto Tecnológico de Orizaba, Division of Research and Postgraduate Studies, Veracruz, Mexico*, 2015

[26] S.C. Geyik, Q. Guo, B. Hu, C. Ozcaglar, K Thakkar, X Wu, K Kenthapadi, Talent Search and Recommendation Systems at LinkedIn: Practical Challenges and Lessons Learned *LinkedIn Corporation, USA,* 2018

[27] L. Gutiérrez, V. Flores, B. Keith, A. Quelopana, Using the Belbin method and models for predicting the academic performance of engineering students. *Department of Computing & Systems Engineering*, Universidad Católica delNorte, Antofagasta, Chile, 2018

[28] O.P. John, S. Srivastava, The Big Five Trait Taxonomy: History, Measurement, and Theoretical Perspectives. *University of California, Berkeley*. GOULD 1981, p.158

[29] J. Rowley, F. Slack, Conducting a literature review. *Management Research News*, Vol. 27 No. 6, pp. 31-39, 2004

[30] G. Basu, Literature Review vs Systematic Review. *San Jose State University*. Retrieved 10 May 2021 from: https://libguides.sjsu.edu/LitRevVSSysRev/definitions

[31] N. Viennot, M. Lecuyer, J. Bell, R. Geambasu, J. Nieh, Synapse: A Microservices Architecture for Heterogeneous-Database Web Applications. *Columbia University,* 2015

[32] Y. Liu, J. Zhao A Task-oriented English Education Platform Powered by ICT&AI., *10th International Conference on Information Technology in Medicine and Education (ITME),* 2019

[33] L.V. Nguyen, M. Hong, J.J. Hung, B. Sohn. Cognitive Similarity-Based Collaborative Filtering Recommendation System, *applied science*, 2020

[34] C. Benfares, Y. El Bouzekri El Idrissi, A. Abouabdellah, Recommendation Semantic of Services in Smart City. *BDCA 17: Proceedings of the 2nd international Conference on Big Data*, Cloud and ApplicationsMarch 2017 Article No.: 52, 2017

[35] L.V. Nguyen, J.J. Hung, M. Hwang. OurPlaces: Cross-Cultural Crowdsourcing Platform for Location Recommendation Services. *International Journal of Geo-Information*, 2020

[36] R.A. Hamid, A.S. Albahri, J.K. Alwan, Z.T. Al-qaysi, O.S. Albahri, A.A Zaidan, A. Alnoor, A.H. Almoodi, B.B. Zaidan, How smart is e-tourism? A systematic review of smart tourism recommendation system applying data management, *Computer Science Review Elsevier*, 2021

[37] V. Vaishnavi, W. Kuechler, Design Research in Information Systems, *Association for Information Systems*, 2013

[38] D Bisandu, Design science research methodology in Computer Science and Information Systems *International Journal of Information Technology,* 2016

[39] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. Chatterjee, A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems, 24:3, 45-77*, 2007

[40] Oracle. What is Java technology and why do I need it?. Retrieved 5 May 2021 from: https://java.com/en/download/help/whatis_java.html.

[41] Python. Retrieved 5 May 2021 from: https://docs.python.org/3/

[42] Golang. Retrieved 5 May 2021 from: https://golang.org/

Appendix A - Class Archetypes Attribute Matrix

ROLES/ATTRIBUTES	Resource Investigator	Teamworker	Co-ordinator	Plant	Monitor Evaluator	Specialist	Shaper	Implementer	Completer finisher
Outgoing	7	6	6	6	3	2	3	2	2
Enthusiastic	8	5	6	6	3	2	5	3	3
Explores Opportunities	9	3	3	6	3	1	5	4	4
Develops Contacts	7	6	6	4	2	2	3	2	3
Co-operative	5	8	3	2	4	3	2	4	4
Perceptive	4	7	3	1	4	2	3	3	4
Diplomatic	3	9	3	2	3	2	2	2	5
Listens and averts friction	3	8	5	2	3	2	2	3	3
Mature	5	4	7	3	4	3	3	5	5
Confident	5	4	9	6	2	5	5	4	3
Identifies Talent	6	3	8	3	3	3	3	2	3
Clafifies Goals	4	3	8	2	3	2	4	2	2
Creative	3	3	5	9	2	4	5	3	3
Imaginative	3	2	5	8	2	3	3	2	4
Free-thinking	2	3	6	9	5	4	5	2	4
Generates ideas	5	4	5	7	2	5	5	3	5
Sober	4	5	4	2	7	4	3	4	5
Strategic	3	4	6	2	9	2	3	3	6
Discerning	3	5	3	3	8	3	2	3	4
Sees all options	6	3	4	2	8	3	2	3	5
Single minded	4	3	2	5	3	8	4	4	3
Self-starting	2	3	5	5	4	7	5	5	4
Dedicated	3	4	3	6	4	8	6	5	3
Provide specialist knowledge	4	3	2	5	5	9	3	4	4
Challenging	3	2	5	6	5	5	7	3	2
Dynamic	2	3	2	4	3	2	7	2	3
Thrives on pressure	2	2	3	5	3	3	8	3	3
Drive to overcome obstacles	3	3	3	4	2	5	8	5	4
Practical	6	5	6	2	6	5	4	7	6
Reliable	4	4	3	1	5	5	5	8	6
Efficient	6	6	4	2	5	5	4	9	6
Turns ideas into actions	3	5	6	3	4	6	5	7	5
Painstaking	4	5	2	2	6	6	3	6	9
Conscientious and anxious	3	4	4	1	3	3	2	5	8
Searches out errors	4	4	2	2	6	5	3	4	7
Polishes and perfects	4	5	5	1	5	6	2	5	9

Appendix B - Student test entity


```
{
  "data": {
        "id": "41",
        "name": "student41",
        "class_name": "",
        "outgoing": "2",
        "enthusiastic": "8",
        "explores_opportunities": "2",
        "develops_contacts": "1",
        "co_operative": "6",
        "perceptive": "2",
        "diplomatic": "4",
        "listens_and_averts_friction": "6",
        "mature": "7",
        "confident": "5",
        "identifies_talent": "2",
        "clarifies_goals": "7",
        "creative": "7",
        "imaginative": "1",
        "free_thinking": "6",
        "generates_ideas_and_solves_difficult_problems": "8",
        "sober": "5",
        "strategic": "3",
        "discerning": "8",
        "sees_all_options_and_judges_accurately": "8",
        "single_minded": "1",
        "self_starting": "6",
        "dedicated": "1",
        "provides_specialist_knowledge_and_skills": "4",
        "challenging": "6",
        "dynamic": "3",
        "thrives_on_pressure": "6",
        "has_the_drive_and_courage_to_overcome_obstacles": "2",
        "practical": "5",
        "reliable": "5",
        "efficient": "4",
        "turns_ideas_into_actions_and_organises_work_that_needs_to_be_done":
"5",
        "painstaking": "4",
        "conscientious_and_anxious": "6",
        "searches_out_errors": "7",
        "polishes_and_perfects": "2",
        "paid_position": "1",
        "distance": "0",
        "full_time": "1",
        "part_time": "0",
        "experience_needed": "0"
    }  }
```

Appendix C - Classified students & work history

Student ID	Student Name	Assigned class	Work History
10	student10	resource_investigator	Project1
11	student11	completer_finisher	Project2
12	student12	teamworker	Project3
13	student13	shaper	Project4
14	student14	teamworker	Project5
15	student15	completer_finisher	Project6
16	student16	resource_investigator	Project7
17	student17	completer_finisher	Project8
18	student18	shaper	Project9
19	student19	teamworker	Project1
20	student20	resource_investigator	Project2
21	student21	plant	Project3
22	student22	resource_investigator	Project4
23	student23	teamworker	Project5
24	student24	shaper	Project6
25	student25	specialist	Project7
26	student26	shaper	Project8
27	student27	implementer	Project9
28	student28	teamworker	Project1
29	student29	resource_investigator	Project2
30	student30	monitor_evaluator	Project2, Project8
31	student31	teamworker	Project4, Project9
32	student32	shaper	Project1, Project5
33	student33	specialist	Project2, Project6
34	student34	shaper	Project3, Project7
35	student35	teamworker	Project4, Project8
36	student36	completer_finisher	Project5, Project9
37	student37	monitor_evaluator	Project1, Project6
38	student38	monitor_evaluator	Project2, Project7
39	student39	completer_finisher	Project3, Project8
40	student40	co_ordinator	Project4, Project9