RULE BASED DISTRIBUTED DATA-SERVICE MODEL

by
Ece Arpacıoğlu

Submitted to Graduate School of Natural and Applied Sciences
in Partial Fulfillment of the Requirements
for the Degree of Master of Science in
Computer Engineering

Yeditepe University
2020

RULE BASED DISTRIBUTED DATA-SERVICE MODEL

APPROVED BY:

Prof. Dr. Şebnem Baydere .....
(Thesis Supervisor)
(Yeditepe University)

Assist. Prof. Dr. Onur Demir .....
(Yeditepe University)

Assist. Prof. Dr. Pınar Sarısaray Bölük .....
(Bahçeşehir University)

DATE OF APPROVAL: ..../..../2020

ACKNOWLEDGEMENTS

I would like to thank my Professor Şebnem Baydere for her support and help with immense gratitude. Pursuing my thesis under her supervision has been an experience which broadens the mind and presents an unlimited source of learning.

I would like to thank Research Assistant Kemal Çağrı Serdaroğlu for his assistance.

Finally, I would like to thank my family, my friends, my cousins and my Grad School family for their endless love and support, which encouraged me to overcome many difficulties faced during my thesis study.

ABSTRACT

RULE BASED DISTRIBUTED DATA-SERVICE MODEL

This thesis proposes a rule-based data management system for cloud-based heterogenous data service architectures. Management of a scalable data service architecture requires an overarching control mechanism that communicates with a large number of web services quickly and accurately in a reliable manner. The system has to be worked with less time and better efficiency during data update of different services to reach these conditions. Hence, the proposed rule-based system communicates with the web services regularly and processes the designated rules sequentially. One of the common communication methods used to manage the requests in a service architecture is the queuing model. In this thesis, a multilayer service queue model is proposed for the mechanism to work better.

In this study, the service management system consists of three main parts; user management, rule management, and rule processing. These parts contain heterogeneous data services needing high security. For this reason, system safety is provided with a two-level protection mechanism which are authentication and authorization.

A special application; Graduate School Board Decision Managements System (GSBDMS) is designed implemented for proof of concept. The limits of the rule-based approach are tested with this application. A comparative performance analysis of the modules is given with proper performance metrics. The proposed queuing model is compared with FIFO for its suitability as the communication mechanism. The performance results reveal that the proposed model is better than FIFO in terms of scalability and response time.

ÖZET

KURAL TABANLI DAĞITIK VERİ-SERVİS MODELİ

Bu tez çalışmasında, bulut tabanlı heterojen veri hizmeti mimarisi için kural tabanlı bir veri yönetim sistemi önerilmiştir. Günümüz veri yönetim sistemlerinde, güvenilir bir sistem elde edebilmek için, çok sayıda web servisinin ve bu servislerle hızlı ve doğru şekilde iletişim kuran kapsamlı bir kontrol mekanizmasının olması önemlilik arz etmektedir. Bu şartlara ulaşmak için, sistemin farklı servislerin veri güncellemesi sırasında daha iyi verimlilikle ve daha kısa sürede çalışması gerekir. Bu nedenle, önerilen kurala dayalı sistem, web servisleriyle düzenli olarak iletişim kurar ve belirtilen kuralları sırayla işler. Bu mekanizmaları yönetmek için kullanılan yaygın haberleşme yöntemlerinden biri de kuyruklama modelidir. Bu tezde, mekanizmanın daha iyi çalışabilmesi için çok katmanlı bir kuyruklama modeli önerilmiştir.

Bu çalışmada önerilen veri yönetim sistemi üç ana bölümden oluşmaktadır; kullanıcı yönetimi, kural yönetimi ve kural işlemedir. Bu bölümler yüksek güvenlik ihtiyacı duyan heterojen veri servislerini içerir. Sistem koruması için, sistem güvenliği iki seviyeli bir koruma mekanizması ile sağlanmaktadır. Bu seviyeler kimlik doğrulama ve yetkilendirmedir.

Önerilen sistemin testi için örnek uygulama olarak Enstitü Yönetim Kurulu Kararları Sistemi (EYKKS) tasarlanmış ve gerçeklenmiştir. Kural tabanlı yaklaşımın sınırları bu uygulama ile test edilmiştir. Bu uygulamanın modüllerinin karşılaştırmalı performans analizi, uygun performans ölçütleriyle verilmiştir. Önerilen kuyruklama modeli, ölçeklenebilirlik ve tepkime süresi açısından FIFO modeli ile karşılaştırılmıştır. Bu karşılaştırma sonuçlarına göre önerilen modelin FIFO'dan daha başarılı olduğu gösterilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

DPSA	Dynamic Priority Scheduling Algorithm
FCFS	First-Come First-Served
MSc	Master of Science
NLP	Natural Language Processing
PhD	Philosophy of Doctorate
RR	Round Robin
SJF	Shortest Job First
SJN	Shortest-Job-Next

1. INTRODUCTION

Technological improvements in the open system solutions facilitate less human operations on data service platforms as automated updates are prone to less errors or defects than manually performed operations. These systems provide further advantages such as uninterrupted runtime, data storage, remote access controlling and so on. Therefore, they are favored against the manually-controlled systems. In addition, possible human-related problems are also considered by system owners. Although an investment cost is required for the design of an automation system in any kind, these systems have various advantageous for producing systematic and robust solutions in the long run. With the help of rapidly grown technological developments, human-less systems can also be designed for distributed heterogeneous information systems.

These technologies are beneficial for data hiding and data processing management. In data hiding, caching data into different platforms is important. As compared to past, increments of needing these platforms have made it difficult to administrate the distributed information systems. Furthermore, data update is necessary to add new information, new features over the old ones for the new platforms. This may result in some alterations like change in a database, webpage update, update into common areas. It should be ensured that these operations are realized in a certain order without any error.

Furthermore, data processing management involves managing data on a digital platform. Since data and data diversity are growing, necessity of data management is important. Because of these situations, the topics which include accuracy of data, simplified reporting process, secure and safe storage, better decision making and increasing productivity have to be taken into account. Therefore, they are used in every field such as archaeology, mathematics, chemistry, education etc.

Taking growing data into consideration, human error rises in data management. Therefore, as an example, one update on the web page may be missed because of many other additional works. Cloud-based heterogeneous data services, web service-based connectivity technologies and a variety of software libraries are available for the development of applications to prevent human interaction errors. These systems include security and version control to be traceable and make it fast and safe. However, the complexity of solutions

primarily requires the development of rule-based structures and management architectures for them.

The main motivation of this thesis is to ensure that information which is stored in different systems can be easily classified and updated when required, based on predefined rules, without any delay and error. Therefore, a rule-based event processing model has been developed for a heterogeneous data service architecture and the success of the model has been proved with a sample application automatically processed the rules. For rule processing, Natural Language Processing (NLP) is used. NLP usage made the process of data splitting effortlessly.

In brief, there is a wide range of application-specific service-based customized automation solutions. A general-purpose system solution that can create and operate rules through a management system is insufficient in some areas. These areas will be explained in background chapter.

Whenever rules are implemented in a distributed information system, queueing models play an indispensable role. These models help inserting, deleting, updating and organizing the data. By applying queueing models, the queues are used for rearranging rules with jobs and subjobs. They are used for different kinds of systems. However, due to occasional insufficiency of speed, a special queueing model which is proposed in this thesis is required.

Briefly, the problem statement is that rule based distributed structure used in the data management systems is insufficient with queueing models in the literature. Our solution is that a queueing model is developed and proposed to be used with data-services in a rule based model heterogeneous system.

The contributions of the thesis are listed below:

- The automated system with NLP for the rule processing keeps away from manual operation.
- Combining NLP, rule based system and queueing model enhances time efficiency in heterogeneous system.
- Improving the accuracy of choosing the suitable subjob for the queue, also contributes to the overall performance of the organization.

- Using data services working concurrently along with queueing model in rule based system enhances process handling.

Cost-efficiency is another gain for other significant and crucial tasks.

The rest of the thesis is organized as follows,

In chapter 2, we explain today's natural language processing, rule-based systems, queueing models, scheduling algorithms, and their insufficient sides. The general solution, how to rearrange these insufficient sides, belongs to this chapter. The differences of the models are compared in terms of some tangible properties.

Chapter 3 includes four sections to explain the whole system. The system architecture design is the first part. The sections of the proposed architecture will be defined as an aspect of design. Each section will be demonstrated by using flowcharts and state diagrams. Second section, the rule-based system design is explained. The progress of the rule-based system with using a sequence diagram and use case diagram is given in this section. The third section is about database design and important parts of the system design at the background. Proposed queueing model approach is explained here. Finally, in the fourth section, the whole system and mechanism of queueing model are indicated with use case scenarios.

Test cases and results show efficiency of the proposed work in chapter 4. The proposed algorithm in terms of performance has been tested. Evaluation of performance has been announced comprehensively according to the outputs of the program. By looking results of analysis, evaluation of efficiency has been observed.

Finally, chapter 5 includes conclusions and future research directions. The conclusion chapter which covers future usage of this methods has been discussed. Usability of the proposed queueing model and incoming plan are represented. According to outputs of this model efficiency, proposed queueing model results in better performance and it provides maintainability in the future.

2. BACKGROUND

With the rising and growing information technologies, new algorithms arise for complex information systems. In these systems, different data types are used in various environments and devices. Systems that are developed for the management of these architectures have to be fault-tolerant and improvable. In heterogeneous information systems, rule based systems are developed to provide these features. Rule based systems work with natural language processing (NLP) and queuing models for keeping organized, optimizing costs. NLP is used for extracting information from text-based document and determining the rules. Queuing models are needed with scheduling algorithms for coordinating the rule based systems and optimizing costs.

This chapter comprises the history of modern NLP, rule based systems, queueing models and scheduling algorithms by comparing with proposed system. NLP is worked for extracting data from text and speech. This plays an important role to determine and form rules in rule based systems. In computer science, a rule based system is used to store and manipulate knowledge to interpret information in a useful way. In the rule based system, main purposes of scheduling algorithms are to minimize resource starvation and ensure fairness amongst the parties utilizing the resources. There are many different queueing models with scheduling algorithms. In this chapter, many of them will be announced.

2.1. NATURAL LANGUAGE PROCESSING

NLP is an interdisciplinary field which an area of research and application that shows how computers can be used to understand and shape natural language text or speech to analyse and perform the desired tasks [1-3]. This processing helps to extract meaning from text or spoken languages. The processing contains some of levels of analysis below.

- Morphological - componential analysis of words, including prefixes, suffixes and roots
- Lexical - word level analysis including lexical meaning and part of speech assignment

- Syntactic - analysis of words in a sentence in order to uncover the grammatical structure of the sentence
- Semantic - determining the possible meanings of a sentence, including disambiguation of words in context
- Discourse - interpreting structure and meaning conveyed by texts larger than a sentence
- Pragmatic - understanding the purposeful use of language in situations, particularly those aspects of language which require world knowledge [4].

There are two assignments using the all levels. First one is taking and analysing queries from the text. Second one is understanding and getting information underlying the queries. They are operated rigorously in different languages. Especially, operations of queries are more difficult in Turkish because of verbal multiword expressions in the corpus of the language. For instance, in Turkish, the words are derived by the suffixes at the end of the word. As Oflazer's extensive study, it is possible to derive up to thirty thousand root words with the help of various suffixes [5].

Some studies handle various type of NLP in Turkish. The following works can be considered as examples.

- Turkish text classification with machine learning and transfer learning: in NLP area, two different Turkish datasets and word vectors on them were created for text classification problem. Word vectors were transferred on the second set of data created with data collected from various news sites with transfer learning. On this data set, text classification process was done with Naïve Bayes and Support Vector Machines, one of the machine learning algorithms. The effects of word vectors used in the study on transfer accuracy and transfer rate and the performance of machine learning methods were analysed in detail. At the end of the study, it was seen that the Support Vector Machine model performed more successfully, and the performance value was improved with the transfer learning method used [6].
- A named entity recognition dataset for Turkish: in named entity recognition topic, for the Turkish language, a data set consisting of news texts marked with the names of assets is presented. The markings cover the names of individuals, places and institutions, which are the main asset names. In addition, a rule-based asset name

recognition system is evaluated on the final version of this data set and relevant evaluation results are presented for reference in further studies [7]

- Text-based Human-Computer interaction in Turkish: in this study in the light of the Turkish grammar rules, a dialogue system for the understanding and interpretation of text expressions has been implemented [8].
- Taking advantage of Turkish characteristic features to achieve authorship attribution problems for Turkish: this work consists of two parts. The first one is the study of Turkish author determination studies in terms of formal features. This short review is the first study to examine Turkish author determination studies in terms of formal features and aims to achieve successful features for Turkish. The second part consists of experiments that make use of the characteristic features of Turkish. The first experiment is designed on the frequency of the verbs in the texts based on the ability to easily derive Turkish words using the Support Vector Machines as learning algorithm [9]

The common feature of these examples is that all of them make sense out of text.

2.2. RULE BASED SYSTEMS

Rule based systems are defined as systems that apply human-made rules to store, sort and manipulate data. In computer science, a rule-based system is a set of "if-then" statements that uses a set of assertions, to which rules on how to act upon those assertions are created. Generally, rule based systems are used for heterogenous information systems. Therefore, heterogenous information systems are studied too when rule based systems are analysed. Studies in [10-11] are examples for heterogeneous information systems. In these systems, external systems using several communication protocols (databases, internet services, embedded systems) with distributed applications were developed. In contrast to fundamental structure similarities in these studies, this project offers more flexible and advanced model.

According to another study, the design of object oriented rule based management consist of 2 stages; rule editing procedures and a rule application model [12]. This system aimed to provide five main features: interchangeability, reusability, modular structure, flexibility and extensibility. Proposed rule based model contained these features too. In addition to these, creating rule stages offered various fields like text-based flexible solutions. Rule based

multimedia application structures [13], using web services [14-15], decision-making mechanisms [16], work activities management [17] can be considered as examples to these previously mentioned text-based flexible solutions.

Generally, rule based structures consist of two main basic stages that are creating rules and applying rules. An example architecture of these stages containing a rule based advisory system for personalization in web-based applications were introduced [18]. The aim of this study was to create an adaptive web application by capturing the user's web usage behaviour in three parts: data collection, model building and giving advice. The information of the user was kept while browsing the web with log files, and a correlation analysis was made between the user information and a rule based structure was developed in these associations. In this way, recommendations were created by targeting pages that the user navigates frequently without scanning the user's entire history. Due to the advantages of the rule-based solution, a rule-based structure was used in the study.

As a result, rule-based structures, rules-based communication of automation network services, and data editing have been studied at different structures and levels. One of the aims of this study is to demonstrate a more flexible and adaptable rule management model with a pilot application and to apply it to heterogeneous services.

2.3. QUEUEING MODELS AND SCHEDULING ALGORITHMS

In many cloud computing systems, the properties like flexibility, scalability and cost-effectiveness are needed for rapid development. In the business areas, computing is divided into consumers, service providers and resource providers. The service providers want to reduce response time and to provide more service and benefits for consumers. Depending on the work performed in the system, the jobs and their multiple processes are used. They need to be faster. It can be done with queues and queueing models in queueing theory with scheduling algorithms.

Queueing Theory is a field of mathematical study dedicated to the understanding and modelling of waiting lines. The purpose of queueing theory is to construct models so that the service waiting time of a client can be predicted, given the system characteristics and the current state of the system. Queueing theory which is founded by Danish scientist Agner

Krarup Erlang in 1917 has become one of the most important elements of the science and the technology, recently. Thanks to the studies of many valuable scientists such as Palm (1943), Takacs (1956, 1957, 1962), Bhat (1965, 1968), Çinlar (1967), Whitt (1972), Gnedenko and Kovalenko (1989) and Atkinson (1995, 2000, 2009), the theory has been enriched by presenting important results and various application areas [19].

The queuing systems without waiting line have been analysed extensively. In this kind of systems, since some of arriving customers left without taking any service, a very important problem called the analysis of "stream of overflows" appeared. The stream of overflows in queuing systems without waiting line was first studied by Palm (1993). Palm (1943) proved that in GI/M/n/0 queuing system, the stream of overflows is a renewal process and found the Laplace-Stieljes transform of the Inter-Overflow time distribution and obtained the loss probability by using difference equations. The models related to queuing systems without waiting line in the literature can be classified into two groups in general:

- M/M/n/0 queuing model: Since there is no waiting line in the system, a customer arriving in the system when all servers are busy leaves without taking any service. This model is analysed by means of Markov process since the interarrival times and the service times have exponential distribution.

- GI/M/n/0 and M/G/n/0 queueing models: Since there is no waiting line in both systems, a customer arriving in the system when all servers are busy leaves without taking any service. However, interarrival times are independent of each other and have an arbitrary distribution in the former, whereas in the latter, the service times are independent of each other and have an arbitrary distribution. Since these models cannot be analysed by Markov process, methods such as supplementary variable, embedded Markov chain, and semi-Markov process were developed. The fundamental problem in this kind of models is the calculation of loss probability and the minimization of this probability [19].

A/B/n/m/d notation given by Kendall (1953), facilitates the definition of the models in the analysis of the queuing systems. A represents the distribution function of interarrival times, B represents the distribution function of the service time, n represents the number of servers, m represents the number of customers waiting in line, and finally d represents the service

discipline. Specially, the letter M stands for the exponential distribution whereas G represents an arbitrary distribution; GI indicates that interarrival times are independent of each other and have an arbitrary distribution function.

Queueing models provide the analyst with a powerful tool for designing and evaluating the performance of queueing systems. In the queueing models, queues are processed according to scheduling algorithms.

A scheduling algorithm is a set of rules that determines the task to be executed in a particular moment. Although there are several scheduling algorithms that have been proposed in the literature, the design of those algorithms is challenged by need for supporting different levels of services, fairness, and implementation complexity and so on [20].

Some popular process scheduling algorithms are listed as: First-Come, First-Served (FCFS) Scheduling, Shortest-Job-Next (SJN) Scheduling, Static Priority Scheduling, Round Robin (RR) Scheduling, Multiple-Level Queues Scheduling. These algorithms use the queues to arrange and execute the processes.

In the first place, processes are executed by an order of arrival to queue in FCFS Scheduling. According to the process which has the shortest execution time, the processes line up into queue in SJN Scheduling. Then, all processes are performed one by one. In the Static Priority Scheduling, each process is assigned by a priority. Process with the highest priority is executed first and then the other processes are performed in order. Into RR Scheduling, each process execution time is divided into time period that is determined with the quantum (a fix time). If it is not completed, it continues after the other processes. Context switching is used to save states of pre-empted process.

Finally, the Multiple-Level Queues Scheduling are mostly used in the big heterogeneous systems. A lot of queues and many basic scheduling algorithms are used in these systems. Multiple queues are maintained for processes with similar characteristics. Each queue has its own scheduling algorithms. If it is needed, the priorities are assigned to each queue. A dynamic priority scheduling algorithm (DPSA) was served based on the static priority scheduling algorithm (SPSA) in 2011 [21]. In this algorithm, there are three queues based on the priority. The processes are placed to queues in order of priority. The priorities are

specified depend on the task's source and the system design. In the top queue, process that has the highest priority is firstly executed. In the other queues, if there is a process which has waited for a threshold time A_k, it is sent to the top queue. In this study, the algorithm is compared with FCFS and SPSA. The performance analysis shows that the advantage of DPSA. In OM algorithm for multi-level queue, scheduling can be an example for this algorithm type [22]. A similar study is made with Shortest Job First algorithm and multiple queues in it. The processes into queues are executed with Shortest Job First algorithm.

In another study, multilevel feedback queue scheduling is implemented and executed [23]. The algorithm has three queues. First queue takes the shortest processes, and also has 10 quantum within the burst time 10 for execution of each process. The variable q_c is created to calculate the time quantum of the second and third queue. If the burst time of the first queue is greater than 10, the processes enter the second queue. In the second queue the processes are performed depending on SJF (Shortest job first) scheduling algorithm. If the burst time exceeds the time quantum, the processes are sent to the third queue. In this study, comparisons are made based on the static and dynamic quantum of the algorithm. With dynamic quantum, the algorithm has better results. Many other studies which are pursued with using different scheduling algorithms and quantum calculations are also available.

In brief, queueing models and scheduling algorithms are detailed and viewed. In our study, new queueing model is proposed examining queueing models and scheduling algorithms in the past.

3. ANALYSIS AND DESIGN

This chapter starts with explaining system architecture design and analysis. After that, each part of the system is clarified one by one. Security and connections are described firstly. Thereafter, NLP from text-based document, rule based system design, proposed queueing model, database design and lastly example of rule processing are detailed sequentially. Therefore, how to create rule based solution is described clearly.

3.1. SYSTEM ARCHITECTURE DESIGN AND ANALYSIS

General system architecture of the structure and connections of model between different services are explained in this chapter. Users, application, database, connected services and connection ways of connected services are represented in Figure 3.1. In this context, the main parts are users, web services, application that is designed tool using web server and authentication server. In this study, users are plenipotentiary user, grad school employee that is named normal user. Furthermore, web services are WordPress, google drive, database, also office programs over google drive.

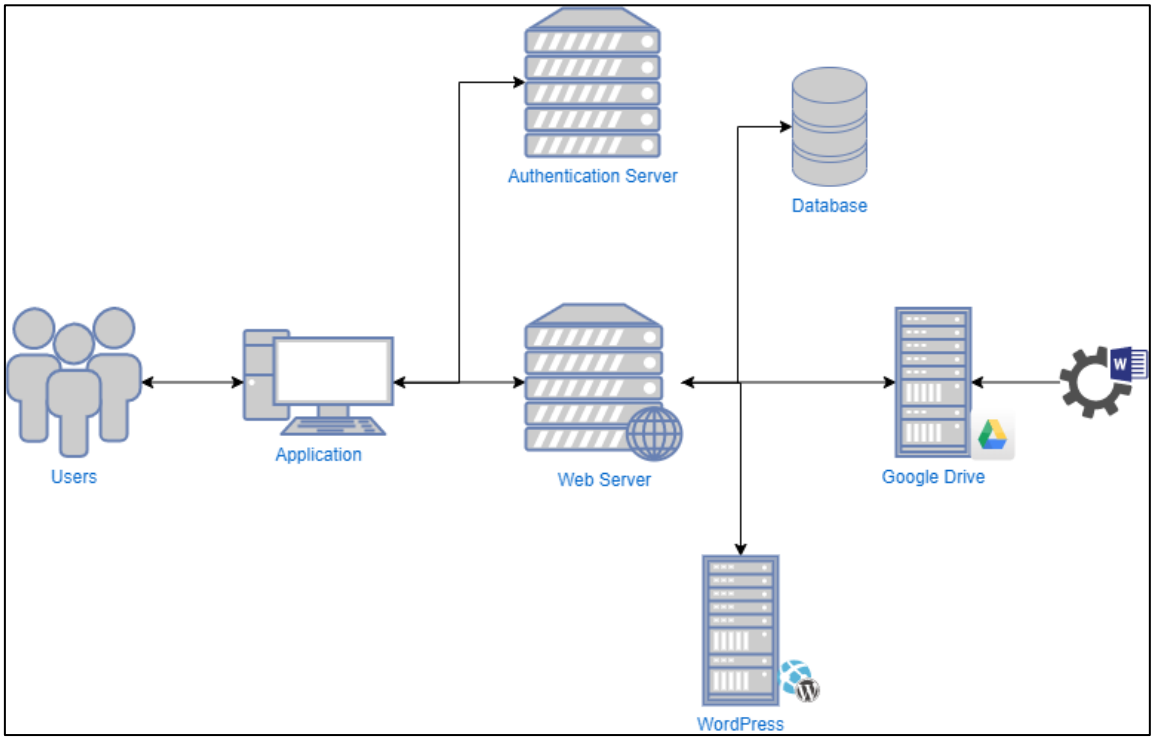


Figure 3.1. General system architecture

In the first place of process of system, a user joins the system with passing the system security part. After that, user can see the management parts of the system according to user role. The Use Case diagram of competences of users is displayed by users in Figure 3.2. As it is seen, there are 2 user types and 5 competences. The first and main plenipotentiary user can control the whole system. Otherwise, the normal user can just access rules defined in the system before board management field.

Competences contain 4 management parts, 1 applying part. Respecting management parts, user security data is collected with user manual access for protection in user management. Furthermore, the fields which a user can reach (competence) into the system are determined at the user role management stage. Rules are added, updated, deleted in rule management. At board management stage, text-based document that is board document in special application designed for this study is uploaded to system with some information of board that are name of document, document date, type of document, operation type, permission type for users, process method, text-based document taking from Google Drive. The document consists of paragraphs that represents the board decisions containing different rules. Therefore, the uploaded document is processed with determined rules in this part. Finally, rules that are defined in rule management can be used by a normal user with accessing rule part.

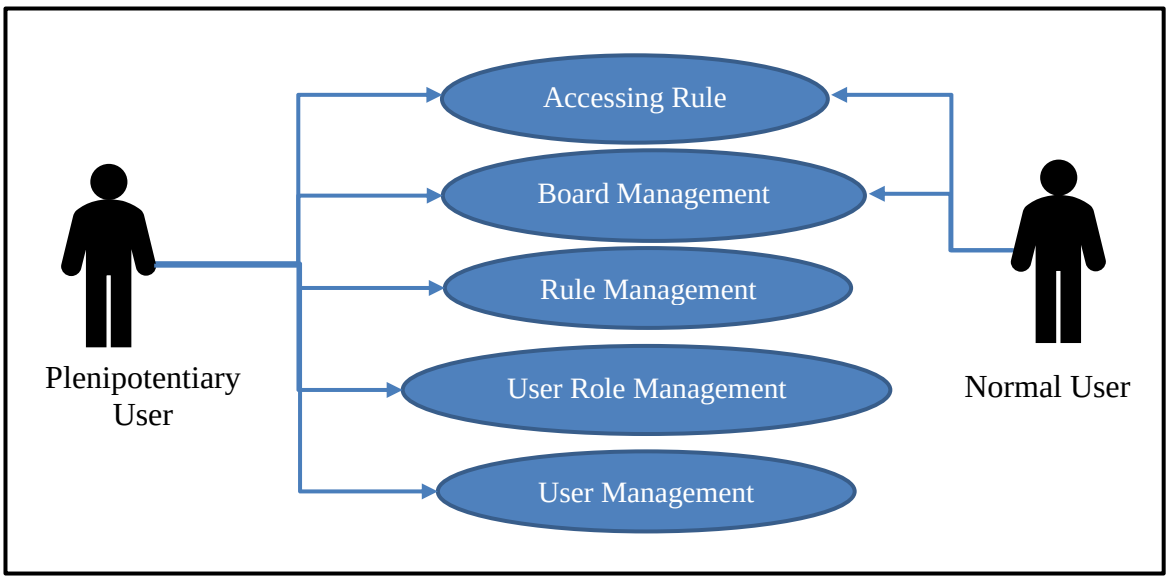


Figure 3.2. Use Case diagram of management system

In this system architecture, after user authentication, uploading document is done to the system, then each paragraph is processed into the document. From each paragraph, a rule is handled, and each rule activates a job. A job contains subjobs and the job finishes after its subjobs are done. These subjobs are connected to the services. Therefore, queueing model is used to manage the subjobs and proposed queueing model is tested with these subjobs in this study.

In this architecture, the dataflow and updates are provided to the needed documents on information system via web services and APIs at the bottom layer. This flow occurs sequentially via rule based event handling. In addition, authorization server provides security between user and centre program while data exchanges. If the user wants to communicate via webservices, he/she needs to verify himself with his user account in authentication server firstly. If his verification is successful, authentication server sends a token opening the gateway for sending request to webservice of software. If token is valid, the transaction is established, and the user can enter the system. The user login flowchart is represented in Figure 3.3. This figure explains the process of login procedure. In login procedure, the user has to attempt the login to program with correct username and password; otherwise, the user encounters the error message. In document processing, sequence diagram represents connections such as Google Drive, database server with needed stages for applying any rule in Figure 3.4.

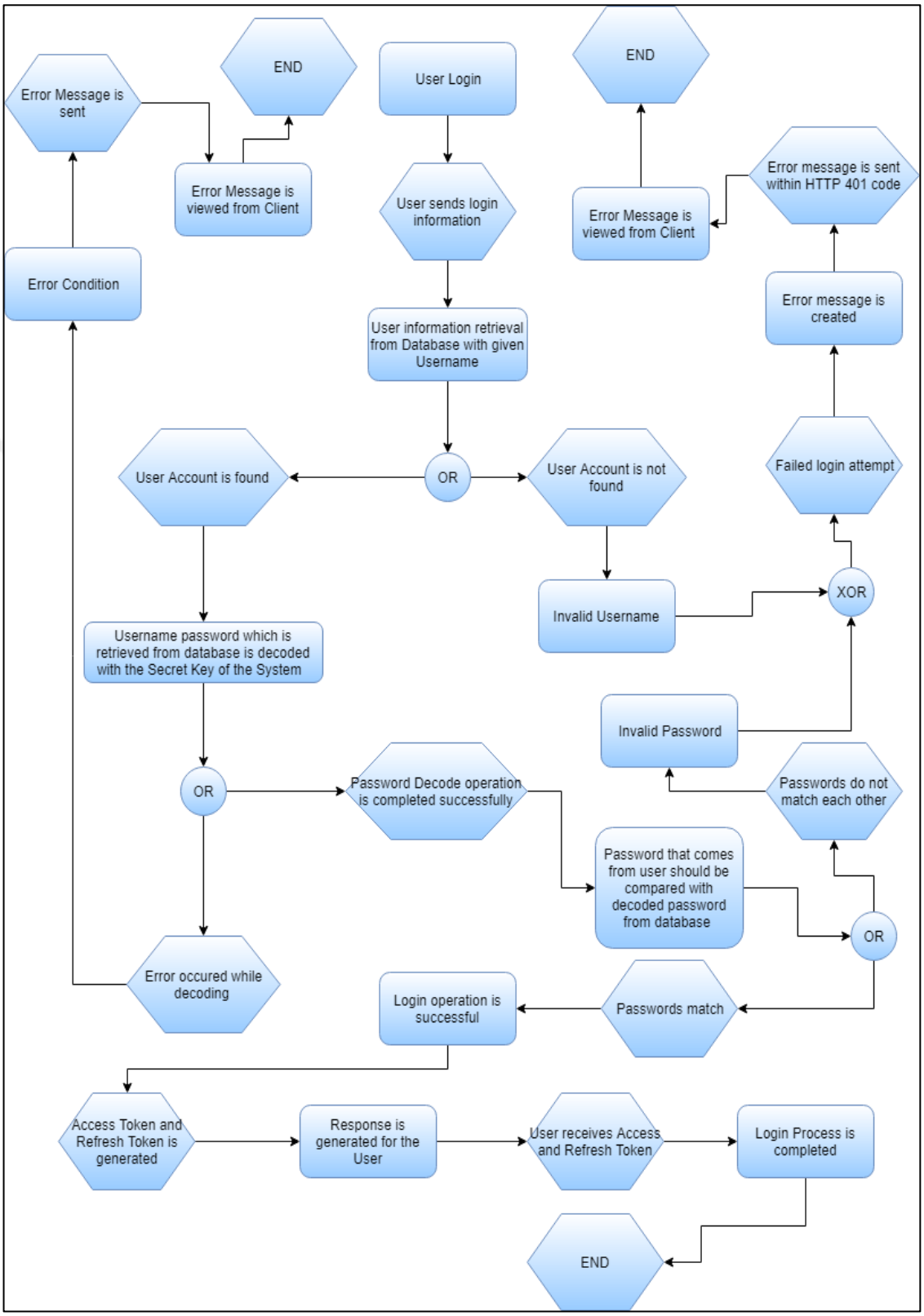


Figure 3.3. User login flowchart

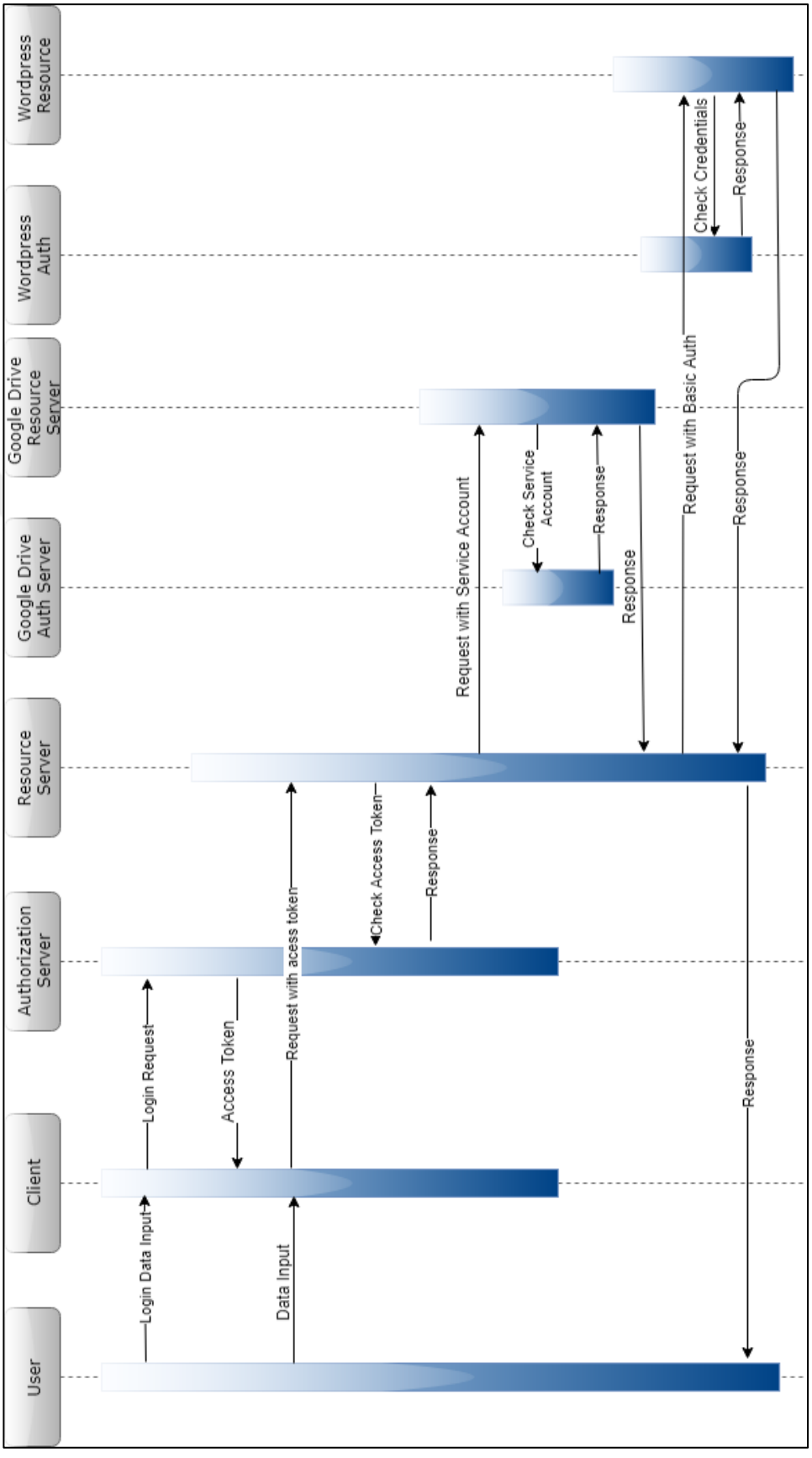


Figure 3.4. Sequence diagram

3.2. NATURAL LANGUAGE PROCESSING FROM TEXT-BASED DOCUMENT

In this section, the paragraphs from the text-based document are explained through board document with NLP in special application designed for this study.

In the board management of system, a lot of rules are created from text-based document. For this creation, text-based board documents are uploaded to the system, then the documents pass some stages that is shown in Figure 3.5.

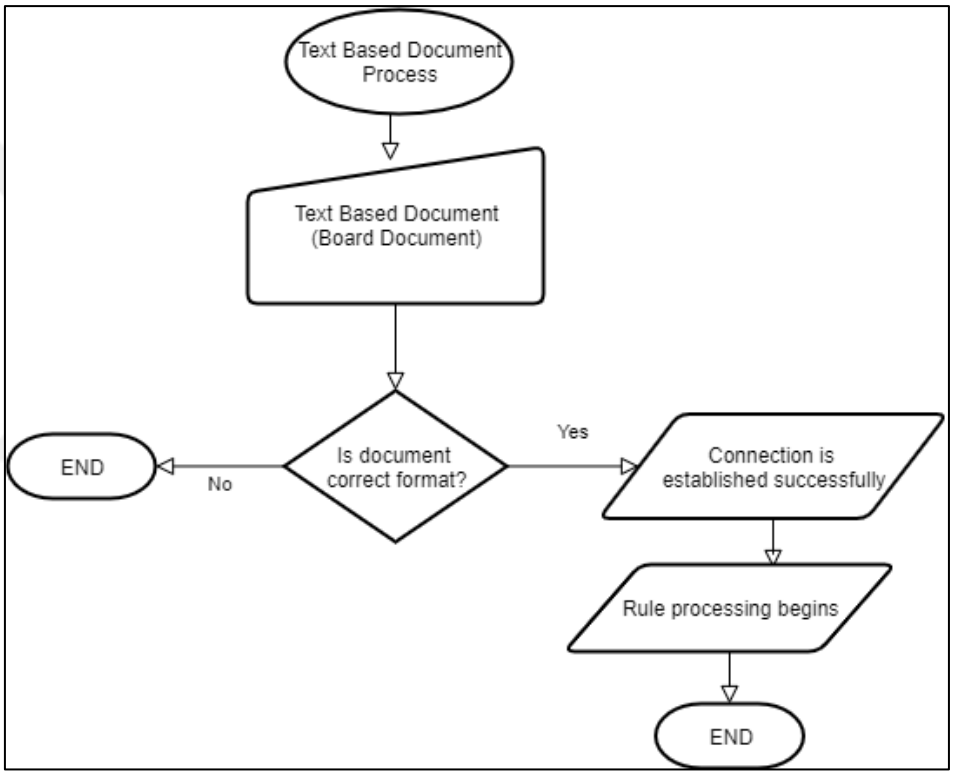


Figure 3.5. Text-based document processing

This stage begins with uploading the board document. When a board document is uploaded to the system, board decisions in the board document are running one by one with operation processing. The next stage is board decision progress. In this board processing, firstly document control is done. Then, board is divided into two parts: general decision titles part and decisions part. Afterwards, each part is controlled whether it is in the correct format or not. Each decision that is represented as a rule in our architecture is shown in two types; regular type and irregular type. The regular type means that the decision/rule is described in certain format. In this system, there are two described rule: changing curriculum, opening

program. Conversely, irregular type does not have certain format. When we encounter the irregular type, the decision is processed if irregular format is described before. Otherwise, the irregular type is not defined before and the decision waits that the description is designed. If any error occurs in these stages so far, the document is taken to waiting stage in the system. The system continues with processing of decisions after the system creates a new irregular type. In the processing of decisions, the decisions are obtained after the board document with regular type decisions is taken from Google Drive folder.

The decisions that are separated into two parts in the board is explained before. In general decision titles part, which decisions into the document are found with basic linear search algorithm. Each title is scanned to find described rule into the system. When the title does not contain rule name, the title is skipped. The other titles are searched. After searching the all titles are done, decisions are handled one by one.

The decision sentence gets through states of natural language processing. Therefore, the sentence can be decomposed into tiny parts. The sentence goes through some states are indicated for determining regular or irregular type in Figure 3.6. After these states are completed, the rule can be applied.

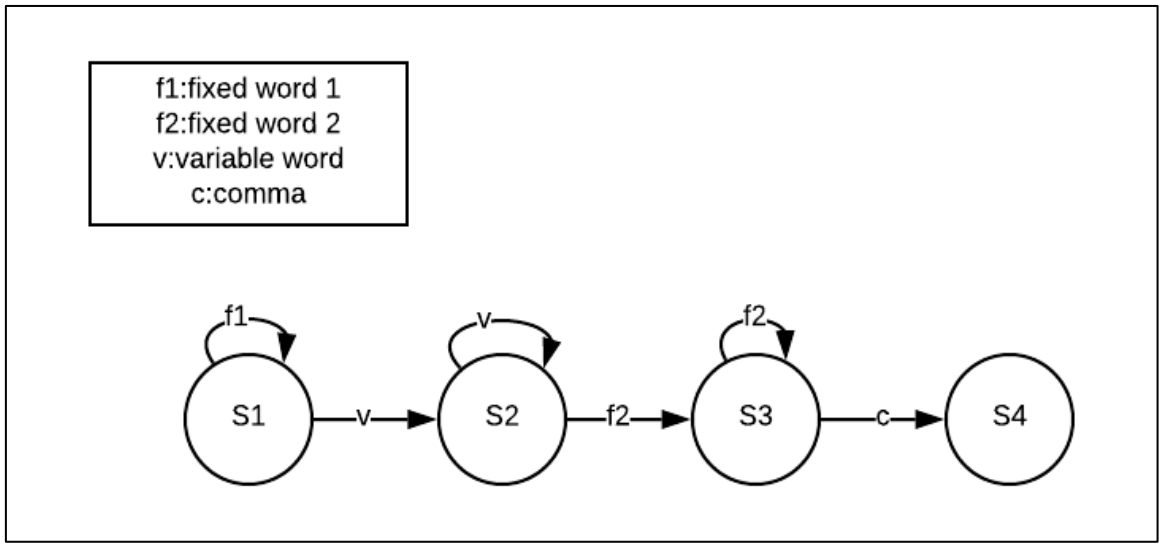


Figure 3.6. State diagram for regular type decision

State S1 that is initial state receives some fixed words which are determined before. After a variable word (that is determined before again) comes, it is jumped to state S2. This points that the variable words are received by State S2. After the state S2, a fixed word comes, and

it is jumped to state S3. Lastly, the comma comes into the sentence and the processing finishes with jumping to final state S4.

If the states are done for one decision, the regular type is done, and some words are taken from the decision. Some words are important for processing rules. Therefore, the words steer the program and determine the rule properties. The following sentences are examples to title of decision, and decision in Table 3.1.

Table 3.1. Example sentences

Sentence Type	**Sentence**
title of decision	"Bilgisayar Mühendisliği Lisansüstü Programlarının müfredat değişikliği talebinin görüşülmesi,"
decision	"Enstitümüz Bilgisayar Mühendisliği Yüksek Lisans Programlarında bulunan Bölüm Seçmeli (Departmental Elective) ders gruplarından birinin Serbest Seçmeli (Free Elective) olarak değiştirilmesine ve yeni düzenlemenin Senatoya arzına,"

From title of decision, 'müfredat değişikliği' is taken and matched with changing curriculum rule. In addition, changing curriculum done with is understood with department name 'Bilgisayar Mühendisliği' phrase. After the title operation is completed, the decision is passed through Natural Language Processing. Firstly, 'Enstitümüz' is taken for fixed word. After that, 'Bilgisayar Mühendisliği' and 'Yüksek Lisans' are taken as variable words to understand that curriculum is changed on which department and program. These variable words are used in rule. 'Programlarında bulunan Bölüm Seçmeli (Departmental Elective) ders gruplarından birinin Serbest Seçmeli (Free Elective) olarak değiştirilmesine ve yeni düzenlemenin' are passed as variable words too. However, these words are not used into the system. Thenceforth, 'Senatoya arzına' is taken as fixed word. Lastly, ',' comma is taken to jump final state.

The NLP is finished and then the rules pass many processes with applications and different kinds of their several connections. The proposed architecture offers an interface for these applications. They are WordPress for the website editing, Google Drive for uploading or downloading documents, Office for some alterations of documents and lastly database to

hold the whole system's important information. This web service process is showed by Figure 3.7.

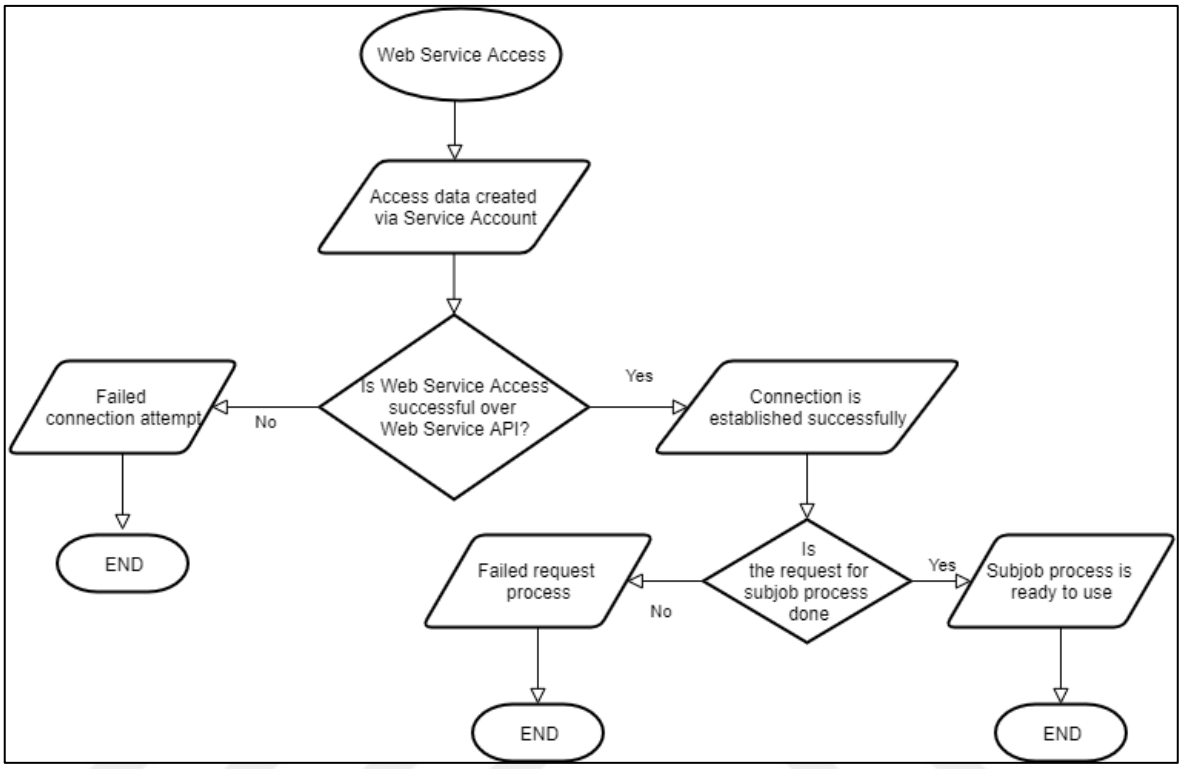


Figure 3.7. Web Service flowchart

To access one web service into the system, access data is created via Service Account firstly. After that, Web Service Access is done, and controlled. When it is not concluded correctly, connection attempt is failed and ended. Otherwise, connection is established successfully. Then, checking the request for subjob process is not done, the request process is failed and ended. Otherwise, subjob process is ready to use and the accessing process is ended successfully.

3.3. RULE BASED SYSTEM DESIGN

A rule based system uses a simplistic model based on a set of IF/THEN statements to implement an expert system [24]. In this system, many rules can store, sort and manipulate data into rule management and board management. In our model, rule based system consists of three parts. This structure is represented in Figure 3.8.

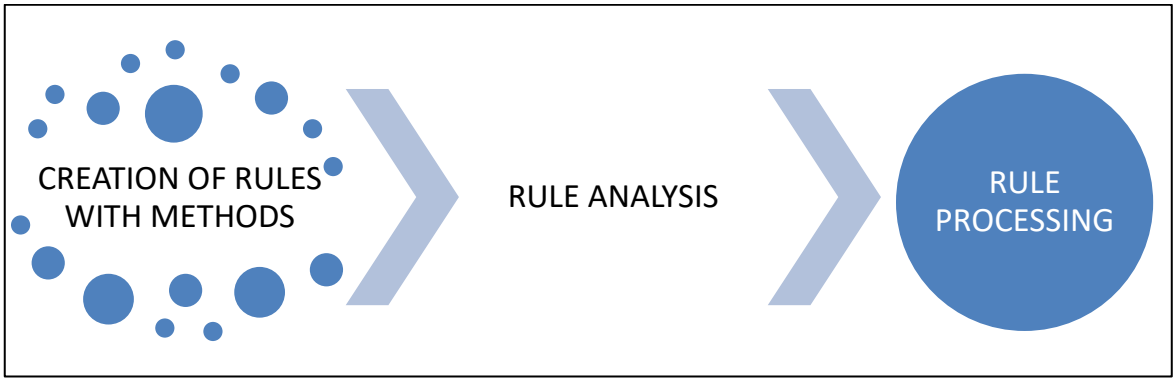


Figure 3.8. Progress of rule based system

In first part, the rules with some methods are created. Afterwards, analysis of rules are done, and then rules are handled in rule processing.

Rules are constructed by the combination of various methods in this work. After that, rules are coordinated with proposed queueing model. Therefore, rules are handled with processing of decisions that academic unit is chosen as an example in “Opening Program”. It is indicated how to compose dynamic and general system with identified rule. In example implementation, “Opening Program”, opening new program rule triggers off from decision and its processing is under debate. Into that rule, there are various steps such as placing curriculum document into Google Drive, updating web pages via WordPress, creating student follow ups via Office etc. The system creates and handles these updates in “Opening Program” rule automatically and proposed algorithm helps it.

The rule processing consists of many connections with programs in heterogeneous system. These connections actualize with programs’ web services. The system is communicated with these web services in queueing model. Proposed queueing model is detailed in next proposed approach section.

3.4. PROPOSED APPROACH

In this study, proposed approach is indicated into expression 3.1 according to Kendall’s notation. This means that n server queue with deterministic arrivals, exponentially distributed service times and m number of waiting positions.

D/M/n/m (3.1)

The proposed approach has 3 main parts: taking rules from board document to the queue, distributing the rules to service queues and processing the subrules with removing from service queues. Each rule connects to a job. Each job consists of many subjobs (subrules that is mentioned). After the jobs are broken into many processes, subjobs settle in the service queues. In addition, subjobs are performed according as dependency state, heuristic time, priority value that is given by user. These cases are shown below into expression (3.2, 3.3).

f: function to convert the job to subjobs
J: job
S: subjob cluster
SU: subjob
n: job number/subjob cluster number
m: total subjob number
β: heuristic time
α: user priority value
δ: dependency state

$$\boldsymbol{f}(J_{\mathrm{n}}, \delta_{\mathrm{n}}, \beta_{\mathrm{n}}, \alpha_{\mathrm{n}}) = S_{\mathrm{n}} \quad (3.2)$$

$$S_{\mathrm{n}} = \{SU_{\mathrm{n1}} \dots SU_{\mathrm{nm}}\} \quad (3.3)$$

In this part, properties of jobs and subjobs are indicated in Figure 3.9. Each job has a identification number (named as no), type that is opening program or changing curriculum for this system, amount of its subjobs, user_priority when there is precedence, its completed heuristic time, note for extra data for processing.

Each subjob has an identification number (named as no), job no that is connected with, subjob no when it is dependent to another subjob, subjob_completed that contains whether dependent subjob is completed, service_no that displays which service operate, user_priority when there is precedence, its completed heuristic time, attribute that contains which service part will operate, note for extra data for processing. The completed heuristic time is calculated with using A* algorithm.

In short, each job has many subjobs, but each subjob connects to just one job. Each subjob expresses different service parts. Service parts can be transmitting document to Google Drive, modifying the Office document or writing a paragraph to webpage into WordPress. In addition, the subjobs can be interdependent as we mentioned. Therefore, Office part may need to be processed after the Google Drive part.

```
class Jobx {
        int no;
        String type;
        int no_subjobs;
        int user_priority;
        int time;
        String note;

        Jobx next;
        Jobx prev;
}

class Subjobx {
        int no;
        int connected_jobno;
        int connected_subjobno;
        boolean connected_subjob_completed;
        int serviceno;
        int user_priority;
        int time;
        int attribute;
        String note;

        Subjobx next;
        Subjobx prev;
        }
```

Figure 3.9. Job-Subjob pseudocode

In brief, connections and structure of general structure shown below in the Figure 3.10. Each subjob is settled as a process in the queues. The number of queues is determined with the number of services. There is a queue for each service. In the proposed architecture, there are three queues, each queue is handled with new scheduling algorithm. Each subjob is distributed to queues according to types of services that they use, Shortest-Job-Next (SJN) scheduling and Priority scheduling algorithms. Each subjob chooses its service's queue. After this choosing, priority field indicates our contribution into the algorithm in this phase. While putting the subjobs into queues, they are compared according to priority with the other subjobs in the queue. When a subjob that has the highest priority comes, it is placed in front of the other subjobs in the queue. When it has lower priority than top subjob in the queue, the comparisons continue until the system finds a subjob that has the lower priority than it. After that, it is placed in front of found subjob. When the all subjobs are bigger than this, it is placed at the end of the queue. When a subjob that is in the queue has same priority with

the subjob, their completion times are compared. The subjob that has the shortest time is placed in front of the other.

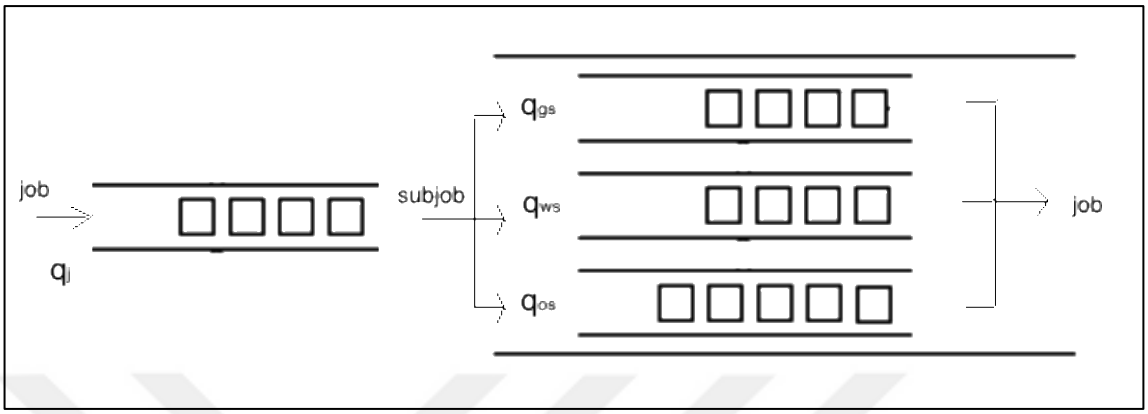


Figure 3.10. Connections between jobs, subjobs and queues

The pseudocode of these stages is shown below in the Algorithm 3.1. Also, the state diagram is in Figure 3.11.

Algorithm 3.1. Placement of subjobs in queues algorithm

```
FUNCTION PlaceSubJobtoQueue (Subjob subjob)
        generalQueuesubjob is set to generalQueue's first element
        WHILE (generalQueuesubjob is empty)
                IF (subjob.priority equals generalQueuesubjob. priority)
                        IF (subjob.time is greater than or equal generalQueuesubjob. time)
                                Push(generalQueuesubjob, subjob)
                        ELSEIF (generalQueuesubjob.time is greater than generalQueuesubjob. time)
                                Push(subjob, generalQueuesubjob)
                        ENDIF
                ELSEIF (generalQueuesubjob.priority is greater than subjob. priority)
                        Push(generalQueuesubjob, subjob)
                ELSE
                        generalQueuesubjob is set to generalQueuesubjob's next element
                ENDIF
        ENDWHILE
ENDFUNCTION
```

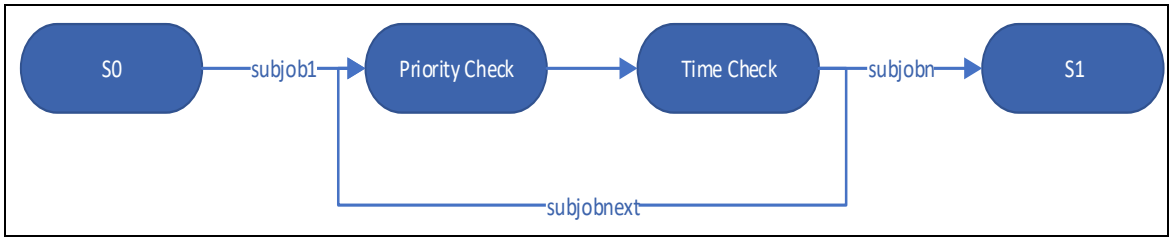


Figure 3.11. State diagram for placing subjob to queue

After this settlement into the queue, the rules are processed with each subjob. But, all subjobs connected to a job need to be finished to complete the job.

3.5. DATABASE DESIGN

In database design, database does not connect user and centre software directly into system architecture in Figure 3.1. This situation is needed for software reliability. The stranger software does not allow to enter the system.

Centre software provides the authentication with authorization server after some procedures are done through user database table. In addition, procedures can be followed with controlling board decision in the system. Accessing to tables that keep the all information about management is done over the program and this access is established with management fields.

Database tables organize the components such as user, userrole, permission, rolepermission, module, modulecomponent, component, componentsetting, method, activitylogs, persons and board. The Entity Relationship diagram is indicated in Figure 3.12. for database.

In this database, while some tables are directly related with the user and his/her properties, some tables are connected to the boards. Furthermore, the connection between the board and the user is established with some tables. These tables and their relationships are detailed below.

Each user has identity, title, name, surname, work, username, password, e-mail, notes and a user role. Each user role can have more than one permission according to permission table. Persons table represents the academicians for this special architecture (in web page update)

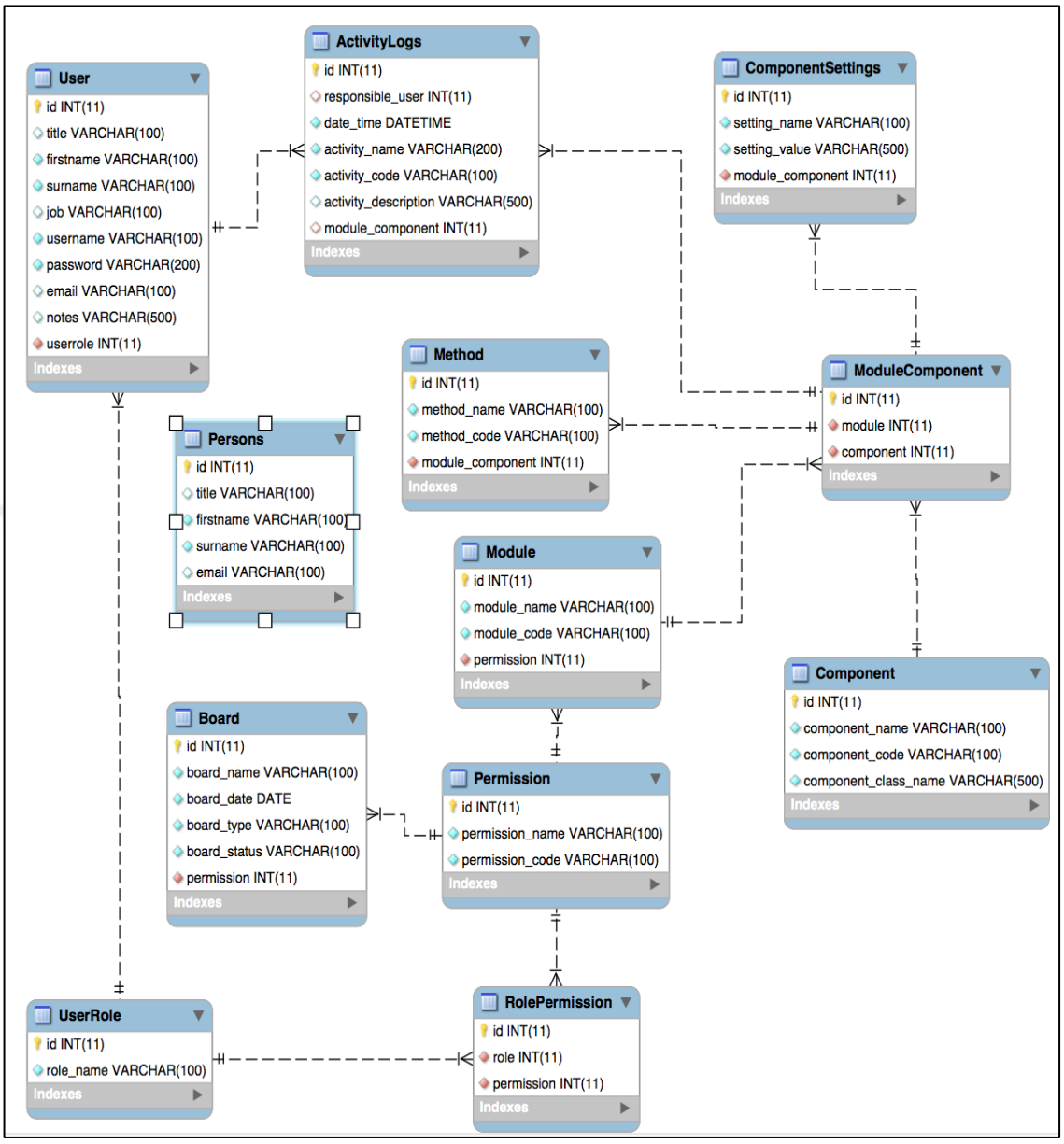


Figure 3.12. Entity Relationship Diagram

The board table is not connected to the other tables, so it is independent. The table helps to show board situation. For example, once board is uploaded to the system, it can be in processing, in done or in waiting status. Therefore, board has id, board_name, board_date, board_type, board_status.

Board processing starts by uploading the system with some properties: name, date, type (Graduate School Executive Board, Administration Executive Board), operation (Done, Processing, Waiting), process (nothing, fifo, new process), document (Executive Board Decisions from Google Drive).

Each different type of board has various decisions, and these are processing in rules. Operation property displays board situation. When board is done to process, there is nothing to do. This option is created for future updates. Changes are needed in board and after the changes are done, board can be processed. Waiting option represents that the board needs to wait for changing into the board and processing. In this option, the board processing is done after it waits certain time. If processing option is selected, board decisions are processed. Process feature determines how board decisions process. Document option helps to choose the board document.

3.6. EXAMPLE OF THE RULE PROCESSING

In this section, “Opening a program” rule is processed as an example for explaining rule processing clearer. “Opening a program” has 3 important stages. These stages are analysed with rule management system. Then, it is seen that there are 4 important parts in this rule: downloading the required documents from Google Drive web service, processing the documents with Office web service, uploading the processed documents to Google Drive web service, updating the webpage with WordPress web service. These parts are needed for different stages.

This mechanism is realized by proposed queuing model in heterogeneous network structure. These rules need to be processed step by step with some parts according to dependencies. These parts are numbered and shown in the Figure 3.13. x, y and z are dependent to each other. y and z are started after x is completed. a is independent, so it can start when it is time for processing in queue. Furthermore, r is dependent to p. This means that r is processed after p.

This example is detailed in 3.7.1. section into use case scenarios.

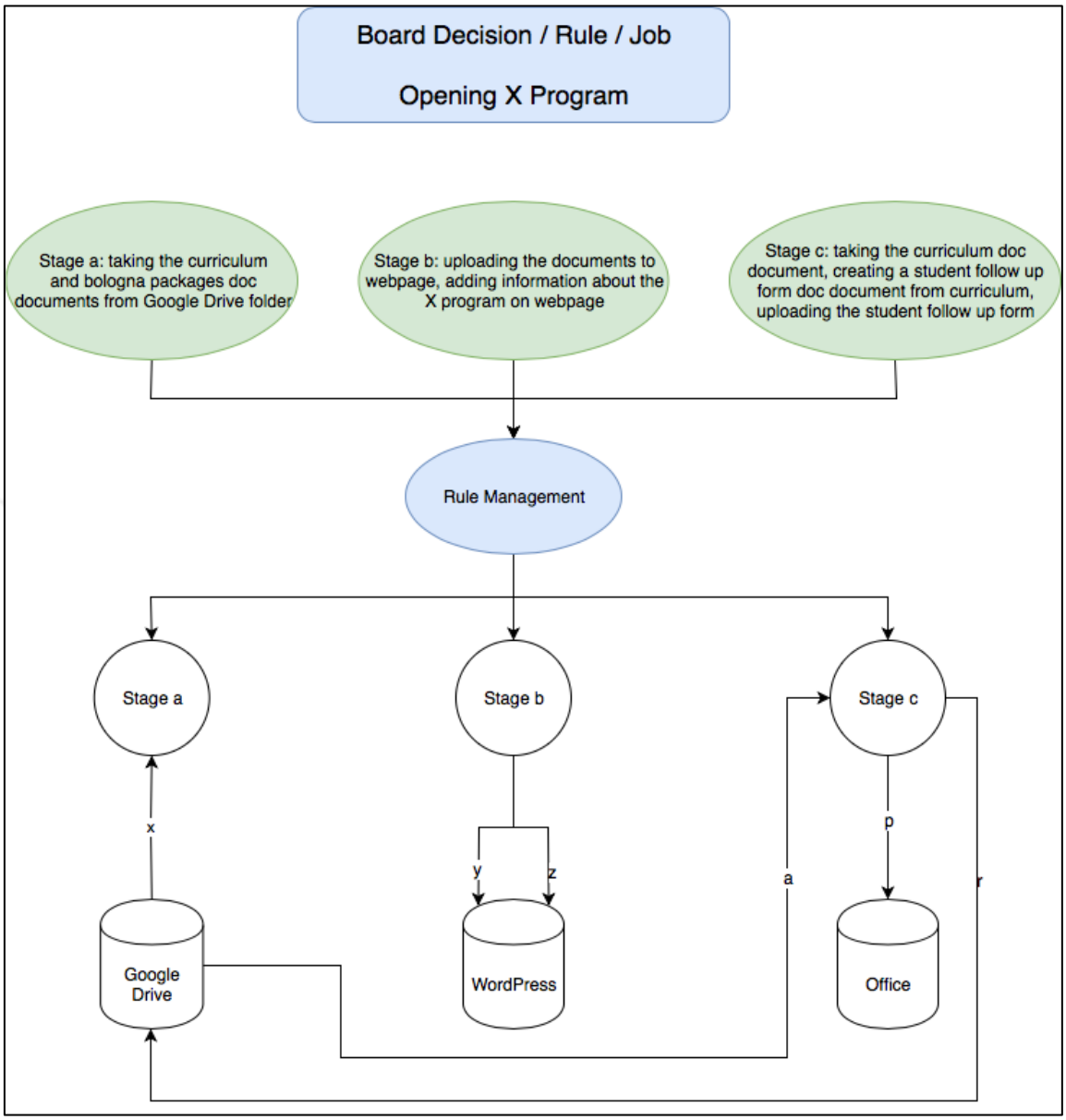


Figure 3.13. Processing of "Opening Program" board decision

3.7. USE CASE SCENARIOS

In this section, possibilities of rule processing scenarios are expressed. First two scenarios contain base rules in the system: opening program and changing curriculum. Others consist of different possibilities with combining the subjobs. In Table 3.2. there are all jobs that are rules and subjobs list with fundamental features.

Table 3.2. Fundamental features of jobs and subjobs

Category	Number	Name	Used Web Service	Number of Subjobs	Completed Heuristic Time (unit)	Type
Job	J1	Opening Program	-	6	22	-
Job	J2	Changing Curriculum	-	3	11	-
Subjob	S1	taking the curriculum and bologna packages doc document	Google Drive	-	3	Download
Subjob	S2	uploading the documents to webpage	WordPress	-	4	Upload
Subjob	S3	adding information about the program on webpage	WordPress	-	4	Change
Subjob	S4	taking the curriculum doc document	Google Drive	-	3	Download
Subjob	S5	creating a student follow up form doc document from curriculum	Office	-	5	Change
Subjob	S6	uploading the student follow up form	Google Drive	-	3	Upload
Subjob	S7	taking the curriculum and curriculum change form doc documents	Google Drive	-	3	Download
Subjob	S8	creating new curriculum with old curriculum and curriculum change form doc documents	Office		5	Change
Subjob	S9	uploading new curriculum	Google Drive	-	3	Upload

3.7.1. Opening Program

This rule represents opening a program in grad school academic structure. It is assumed that opening a biomedical engineering in MSc program is required. In order to supply this request, the rule contains 7 subjobs and 3 connections with its webservices. Subjobs are placed into queues with the some features detailed in Table 3.3. Their dependencies (as in Connected Subjob Number column), their connected jobs, completion times, queues where subjobs are placed into (as in Queue Number column), and priorities are displayed as below. All subjobs are connected to ‘opening program’ job because there is only one job for this scenario. In addition to this, priorities of the subjobs are same. Furthermore, the subjobs are placed into queues in order to visualize this scenario as shown in Figure 3.14. In the figure, $q_{gs,}$ q_{ws}, q_{os} represent Google Drive Service Queue, WordPress Service Queue and Office Service Queue, respectively.

Table 3.3. Specialities for 'Opening Program' rule

Subjob Number	Connected Job Number	Connected Subjob Number	Completion Time (unit)	Queue Number	Priority Number
S1	1	-	3	1	1
S2	1	S1	4	2	1
S3	1	S1	4	2	1
S4	1	-	3	1	1
S5	1	S4	5	3	1
S6	1	S5	3	1	1

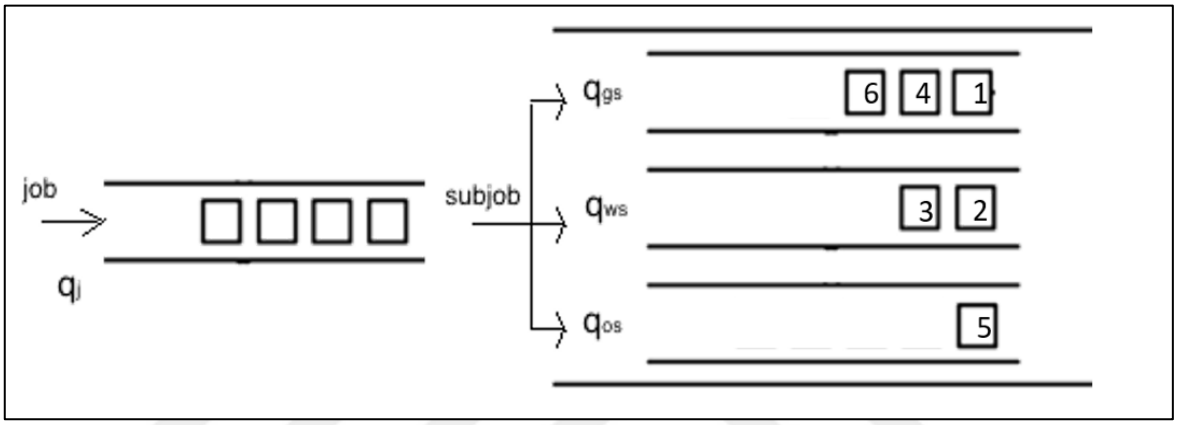


Figure 3.14. Subjobs of ‘Opening Program’ in queues

Working schema is detailed below in an order (Figure 3.15.):

- Firstly, subjob 1 works. However, the other subjobs 2 and 5 placed on top of the other queues wait the $q_{gs,}$, since they are dependent to subjobs 1 and 4.
- While these subjobs wait, 1 unit step is passed. Therefore, subjob 4 begins to work and subjob 3 starts to wait. Because subjob 3 has a dependency to subjob 1. In other words, it waits to subjob 1.
- Subjob 6 starts to wait because it has dependency, after 1 unit is passed.
- After 1 unit is passed, subjob 1 is completed. From that time, the subjobs 2 and 3 can start to work.
- After 1 unit is passed, subjob 4 is finished. Therefore, subjob 5 starts to process and subjob 6 waits.
- Afterwards, 3 units are passed, subjob 2 and 3 finish.
- After 1 unit is passed, subjob 5 finishes and subjob 6 starts to work.
- Lastly, 3 units are passed and all subjobs are completed. Therefore, the job (rule) is completed.

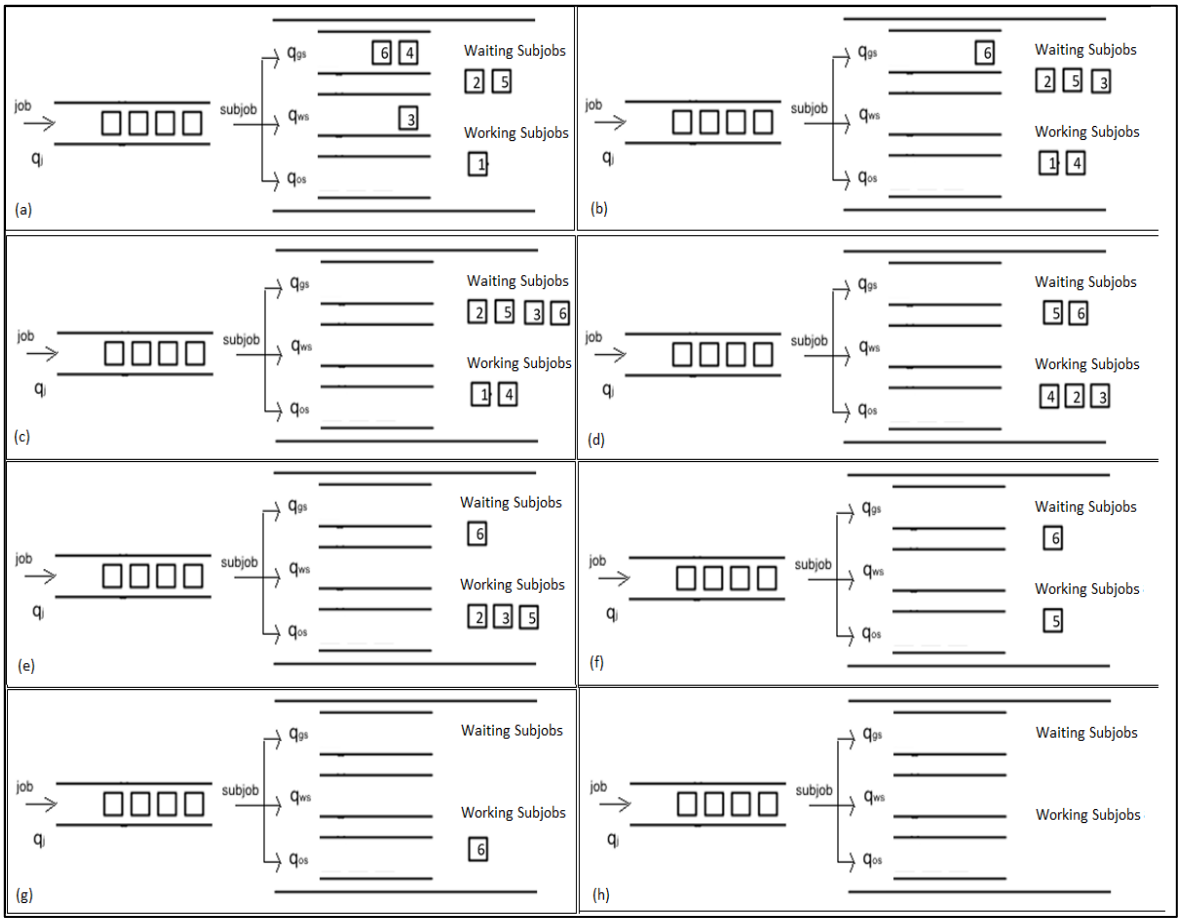


Figure 3.15. Steps of 'Opening Program', (a) Step 1, (b) Step 2, (c) Step 3, (d) Step 4, (e) Step 5, (f) Step 6, (g) Step 7, (h) Step 8

In brief, job is splitted to subjobs. Then, subjobs are placed into queues. According to their dependencies, while some subjobs work simultaneously, the others work, respectively. In addition, when all subjobs are processed in FIFO, total completion time is 22 units which is sum of the whole completion time of all subjobs. Otherwise, all subjobs are processed in our proposed model, so total completion time is 12 units. According to results of this scenario, proposed model has better performance than FIFO.

3.7.2. Changing Curriculum

This rule represents changing a curriculum of a program in grad school academic structure. For instance, curriculum of computer engineering in MSc is wanted to be changed. This rule contains 3 subjobs and 2 connections with web services. After the subjobs are detailed, subjobs are placed to queues with features in Table 3.4. Their dependences, priorities,

connected job, completion times are displayed below. All subjobs are connected to ‘changing curriculum’ job because there is one job for this scenario. In addition, priorities of subjobs are same. Furthermore, in Figure 3.16. the subjobs are placed to queues with visual way.

Table 3.4. Specialities for 'Changing Curriculum' rule

Subjob Number	Connected Job Number	Connected Subjob Number	Completed Heuristic Time (unit)	Queue Number	Priority Number
S7	1	-	3	1	1
S8	1	S1	5	3	1
S9	1	S2	3	1	1

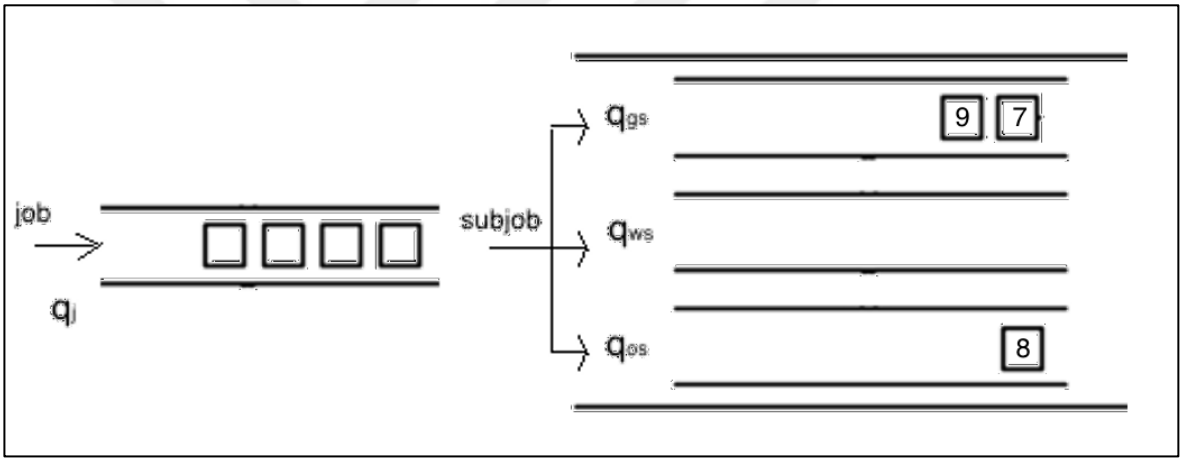


Figure 3.16. Subjobs of ‘Changing Curriculum’ in queues

Firstly, each subjob is dependent to another subjob in this scenario. Working schema is explained below in an order (Figure 3.17.):

- Subjob 7 starts to work at first step, but subjob 8 needs to wait to complete the subjob 7.
- After 1 unit, subjob 9 start to wait processing.
- Subjob 7 finishes after 2 units. While subjob 8 starts to process, subjob 9 still waits to subjob 8.
- When 5 units are passed, subjob 8 finishes. Also, subjob 9 starts to work.
- Thereafter 3 units are passed, all subjobs are completed.

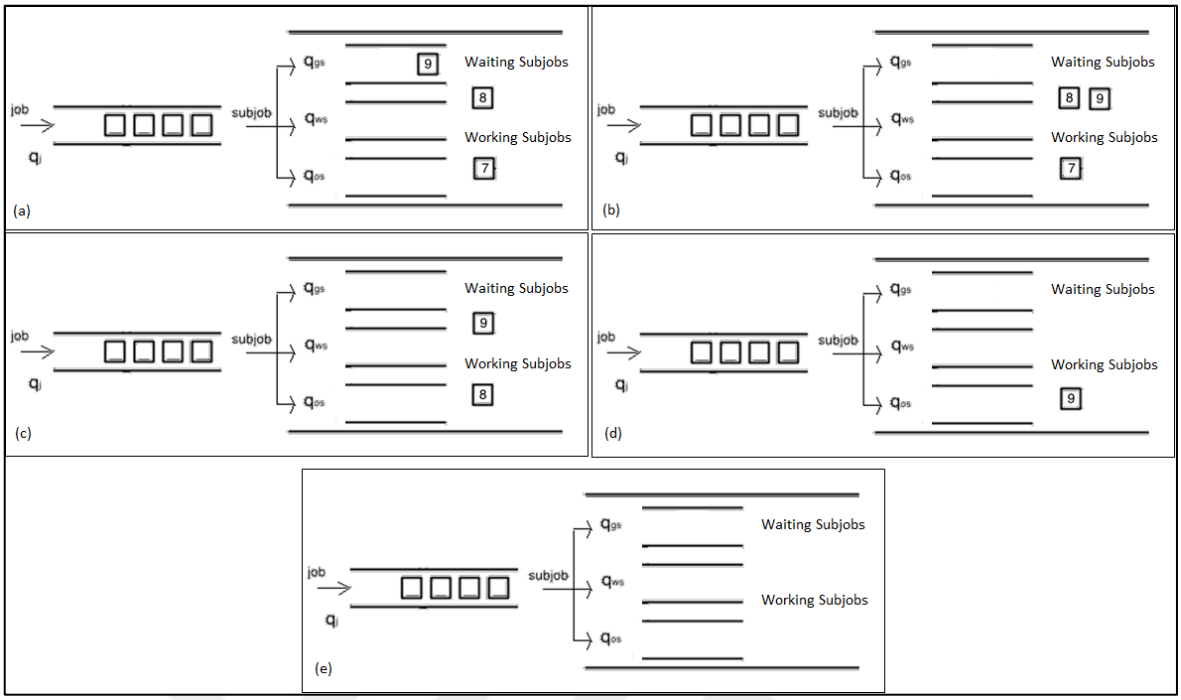


Figure 3.17. Steps of 'Changing Curriculum', (a) Step 1, (b) Step 2, (c) Step 3, (d) Step 4, (e) Step 5

Briefly, all subjobs take 11 units. When all subjobs are processed in FIFO, total completion time is 11 units. The completion times are equal to each other because of their dependencies.

3.7.3. Opening Program and Changing Curriculum

In this scenario, there are 2 jobs and 9 their subjobs. The jobs are opening program and changing curriculum that are used in previous examples.

Table 3.5. Specialities for 'Opening Program' and 'Changing Curriculum' rule

Subjob Number	Connected Job Number	Connected Subjob Number	Completed Heuristic Time (unit)	Queue Number	Priority Number
S1	1	-	3	1	2
S2	1	S1	4	2	2
S3	1	S1	4	2	2
S4	1	-	3	1	2
S5	1	S4	5	3	2
S6	1	S5	3	1	2
S7	2	-	3	1	2
S8	2	S7	5	3	2
S9	2	S8	3	1	2

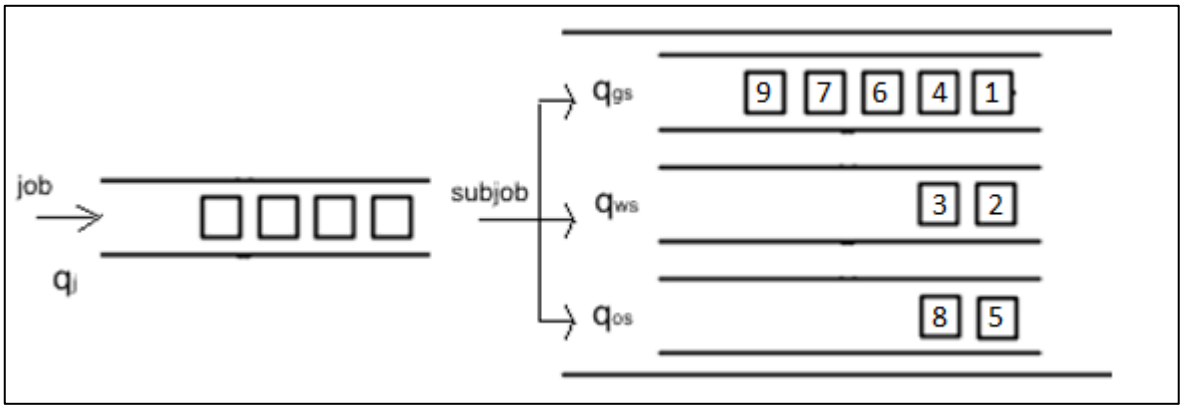


Figure 3.18. Subjobs of 'Opening Program' and 'Changing Curriculum' in queues

Working schema is detailed below in an order (Figure 3.19.):

- Step 1:Subjob 1 starts to process, subjob 2 begins to wait for completion of subjob 1, and subjob 5 begins to wait for completion of subjob 4.
- After 1 unit, subjob 4 starts, subjob 3 and 8 begin to wait.
- After 1 unit again, subjob 6 begins to wait, subjob 1 completes, and subjob 2 begins to work.
- Subjob 4 finishes and subjobs 3,5,7 begin to process after 1 unit.
- Subjob 9 starts to wait after 1 unit.
- 2 units are passed and subjob 2 and 7 finishes and subjob 8 begins to work.
- 1 unit later, subjob 3 is finished.
- After 1 unit, subjob 5 is finished and subjob 6 starts to process.
- After 3 units, subjob 6 and 8 finish, and subjob 9 starts.
- Lastly, after 3 units, subjob 9 finishes and the processing of subjobs is completed.

The total completed time of this processing is 15 units in proposed model. Otherwise, all subjobs are processed sequentially in FIFO and the total completed time of this processing is 33 units. That the proposed model is better than FIFO is proved again.

In 3.7.3.1. and 3.7.3.2., settlements to queues are done depending on priority and completed heuristic time of new subjobs. Formerly, the mechanism of processing of subjobs are clarified. In these sections, the settlements are detailed.

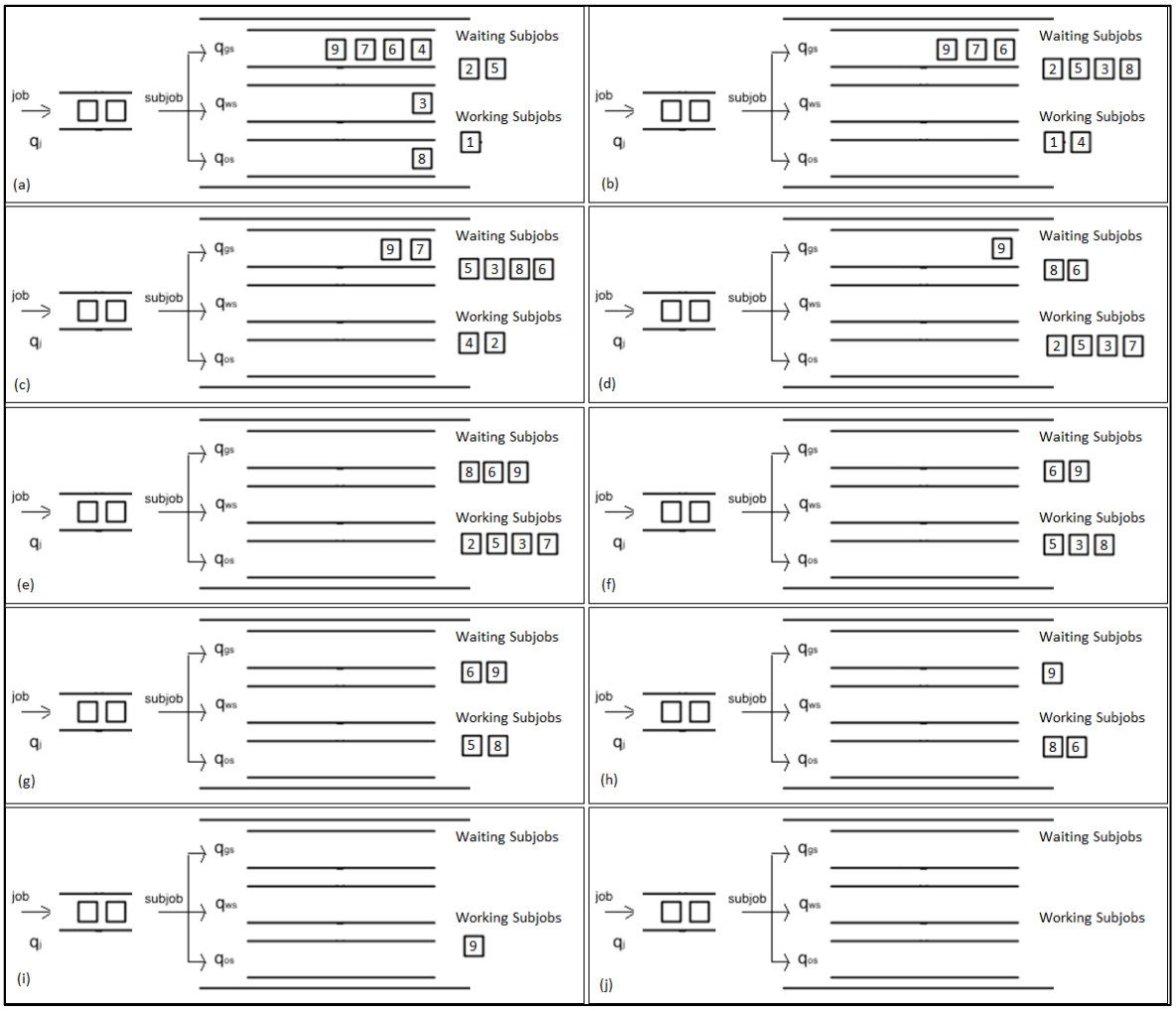


Figure 3.19. Steps of 'Opening Program' and 'Changing Curriculum', (a) Step 1, (b) Step 2, (c) Step 3, (d) Step 4, (e) Step 5, (f) Step 6, (g) Step 7, (h) Step 8, (i) Step 9, (j) Step 10

3.7.3.1. Opening Program and Changing Curriculum with a New Rule 1

In another scenario, an extra variation of the job is added to 3.7.3. section. The connections and features of this variation are explained in Table 3.6 and Table 3.7. In addition, all subjobs are rearranged in Figure 3.20. Until now, there is no change of priority values and completed heuristic times for all subjobs. In Table 3.6. new subjobs are added with the connected queue numbers. In this example, Table 3.7. shows that completed heuristic times are changed when the priority values are same.

Table 3.6. Connections with new rule 1

Subjob Number	Connected Job Number	Connected Subjob Number	Queue Number
S1	1	-	1
S2	1	S1	2
S3	1	S1	2
S4	1	-	1
S5	1	S4	3
S6	1	S5	1
S7	2	-	1
S8	2	S7	3
S9	2	S8	1
S10	3	-	1
S11	3	-	1
S12	3	-	2
S13	3	-	2

Table 3.7. Features of new rule 1

Category	Number	Name	Used Web Service	Number of Subjobs	Priority	Completed Heuristic Time (unit)	Type
Job	J3	Arrangement	-	4	2	14	-
Subjob	S10	changing curriculum name on Google Drive	Google Drive	-	2	2	Change
Subjob	S11	changing parent folder of curriculum	Google Drive	-	2	4	Change2
Subjob	S12	changing data in curriculum from Board	Office	-	2	6	Change2
Subjob	S13	create new doc office	Office	-	2	4	Create2

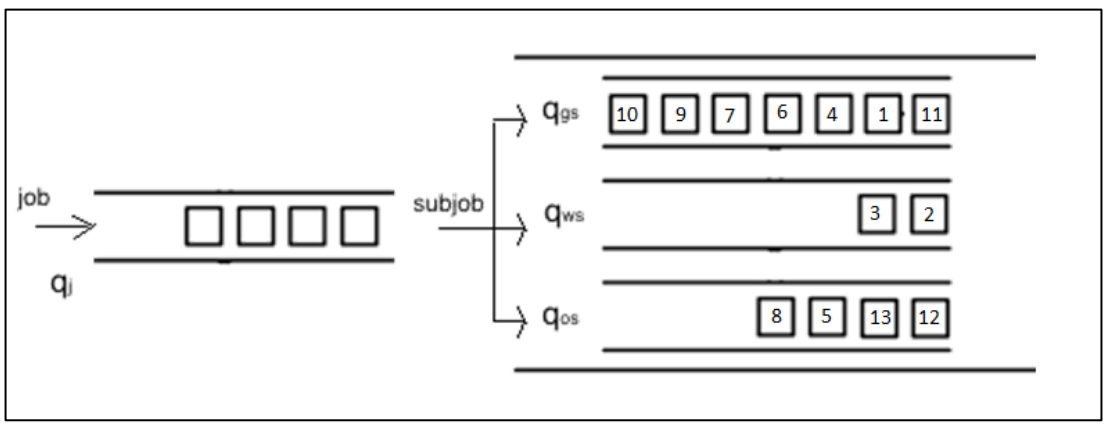


Figure 3.20. Subjobs with new subjobs of new rule 1 in queues

Figure 3.20. displays that all subjobs placements. In this placement, subjob 10 and 11 is settled to queue 1 when subjob 12 and 13 are settled to queue 3 because of their web service types. In addition, because subjob 11 and subjob 12 have the shortest completed times, they are settled the top of the queues. Otherwise, subjob 10 has the biggest completed time. Therefore, it is settled as the last element of the queue. Subjob 13 has bigger completed time than 12. Furthermore, it has the lowest completed time according to other subjobs except subjob 12. Because of them, it is settled after the subjob 12 in the queue.

3.7.3.2. Opening Program and Changing Curriculum with a New Rule 2

In another scenario, an extra variation of the job is added to 3.7.3. section. The connections are explained in Table 3.6 before. Furthermore, features of extra one is clarified in Table 3.8. In addition, all subjobs are rearranged in Figure 3.18. In this example, Table 3.8. shows that priority values are changed when completed heuristic times are same. In brief, opposite side of section 3.7.3.1 is offered int this scenario.

Table 3.8. Features of new rule 2

Category	Number	Name	Used Web Service	Number of Subjobs	Priority	Completed Heuristic Time (unit)	Type
Job	J3	Arrangement	-	4	1	14	-
Subjob	S10	changing curriculum name on Google Drive	Google Drive	-	2	3	Change
Subjob	S11	changing parent folder of curriculum	Google Drive	-	3	3	Change2
Subjob	S12	changing data in curriculum from Board	Office	-	3	3	Change2
Subjob	S13	create new doc office	Office	-	1	3	Create2

Figure 3.21. displays that all subjobs placements. In this placement, subjob 10 and 11 is settled to queue 1 when subjob 12 and 13 are settled to queue 3 because of their web service types. In addition, because subjob 10, 11, 12 have the lowest priority value in the queue, they are settled the bottom of the queues. Furthermore, subjob 10 has bigger priority value than

subjob 11. Therefore, subjob 10 is placed in front of subjob 11. On the other hand, subjob 13 has the biggest priority value. Therefore, it is settled as the first element of the third queue.

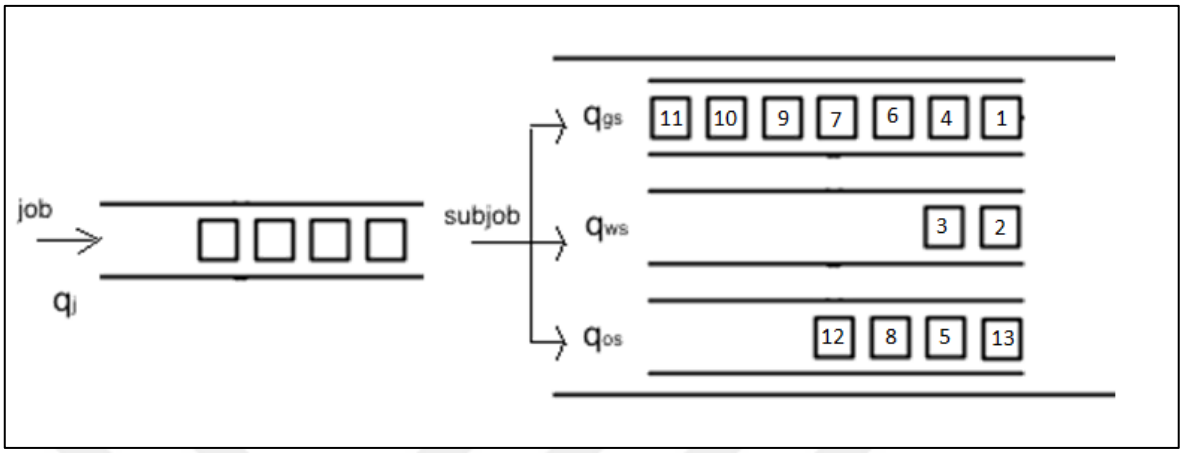


Figure 3.21. Subjobs with new subjobs of new rule 2 in queues

When the new rule 3 is composed like Table 3.9., the processing order is same with the new rule 2. Because of this, the proposed algorithm looks the priority firstly. When the priority values are different from each other, the algorithm does not look the completed heuristic time.

Table 3.9. Features of new rule 3

Category	**Number**	**Name**	**Used Web Service**	**Number of Subjobs**	**Priority**	**Completed Heuristic Time (unit)**	**Type**
Job	J3	Arrangement	-	4	1	14	-
Subjob	S10	changing curriculum name on Google Drive	Google Drive	-	2	3	Change
Subjob	S11	changing parent folder of curriculum	Google Drive	-	3	4	Change2
Subjob	S12	changing data in curriculum from Board	Office	-	3	2	Change2
Subjob	S13	create new doc office	Office	-	1	3	Create2

4. TESTS AND RESULTS

Until now, system background is explained, the architecture of system is analysed. In this stage, the tests of the system are done, and their results are evaluated. Therefore, this part contains two main pieces; first one is real test environment and second one is performance tests results analysis of the queueing models that are the proposed queue model and FIFO queue model.

Real test environment is explained in the first part, then test results are detailed. The performance tests organized depending on repetition of jobs and dependency to each other.

4.1. REAL TEST ENVIRONMENT

The environment is built on two servers with two computers in the same network. These servers are used for user authentication, communication with web services, and tool for user interface.

The system begins to run with interaction with user interface. When the user attempts to enter the system, the login window is encountered. The login window helps the authorization service with user management system. The login window is indicated in Appendix A (Figure A.1.). Furthermore, if there is an error with wrong username and/or password, the login fail popup appears in Appendix A (Figure A.2.).

When the user reaches our system, general view appears as in Figure A.3. In this page, there are three management parts which are User Management, Rule Management, and Board Management. The first two parts were already explained in detail previously. Board Management is added for queuing models test design. In board, a decision is connected by a rule with a job that consist of subjobs, these subjobs need a connection with many web services. These subjobs sometimes works independent from the others but sometimes dependent to each other, and thus they need the rule processing system. The rule processing system provides to organise the subjobs.

User management system contains three little sections that are managing users, userroles, permissions. Rule management system consists of managing rules and their components. This section helps to manage board part.

Board management part is indicated in Figure A.4. In this system, there are three operation: add new board, edit board, delete board. How to use these sections are detailed in Appendix B.

In this section, we explain how to process the board. When the board uploaded the system with tool, the board decisions are decomposed to create jobs separately. The job has many subjobs that are used webservices to communicate different technologies that are Google Drive and Office in this test environment.

Completion time is calculated with different test environments to do performance analysis.

In test environments, there are changing test items consisting of numbers of subjobs, number of jobs, and queueing models that are the main important items for comparing. In queueing models, FIFO and proposed model are compared to each other depends to other items. There are many types of subjobs; downloading document from Google Drive, uploading document to Google Drive, and making changes into document with Office program. Range of number of jobs changes from one to 100. According to number of jobs, completion time is calculated for group of jobs. The test is repeated four times to calculate completion times, the mean of the calculations is indicated with the graphics.

4.2. ANALYSIS OF TEST RESULTS

The tests contain dependent and independent jobs and the number of jobs. In the graphics of test results, the blue line shows values for FIFO, otherwise the orange line shows values for proposed one. First test environment is established with one job repeating and containing a subjob. The job involves downloading document from Google Drive subjob. More than one job is described in the test environment and the results of completion time are taken for each model. Same test is done for making changes into document with Office subjob. The results are reviewed with Figure 4.1. and Figure 4.2.

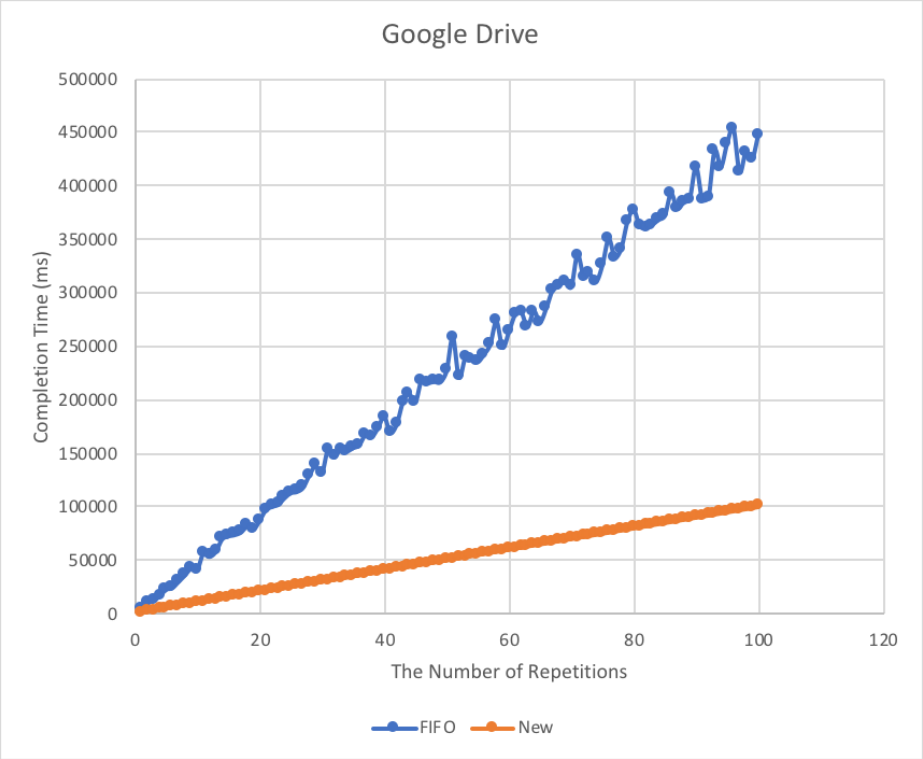


Figure 4.1. Completion time statistics for Google Drive subjob

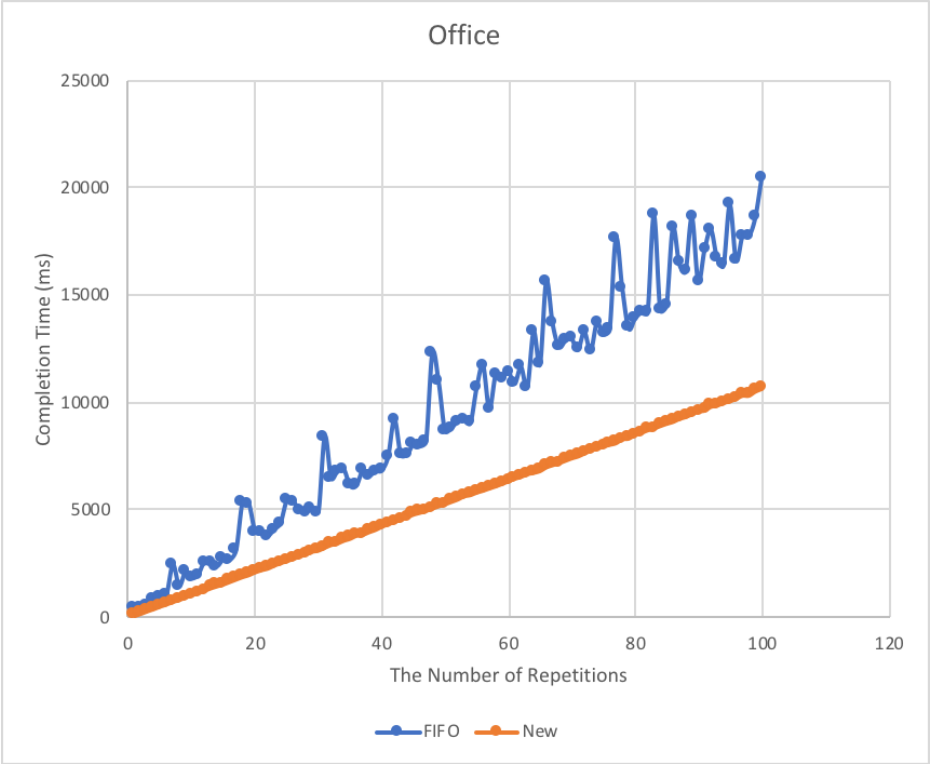


Figure 4.2. Completion time statistics for Office subjob

The test results show that the proposed model works with better performance than FIFO. Statistics point to positive increasing graph. This means that graph is linear. Therefore, the performance of the proposed model is four times faster than FIFO model according to Google Drive subjob. Also, the proposed model performance is two times faster than FIFO model with Office subjob. Office subjob comes out to be slower than Google Drive subjob.

The second test environment consists of one job with two subjobs. This job contains downloading document from Google Drive subjob and making changes into documents with Office subjob. Independent Dual graph shows that results of the job that have independent subjobs in Figure 4.3. Conversely, in the dependent dual graph subjobs are dependent to each other. This means that in this dependency, Office subjob needs to be done after Google Drive subjob. The jobs are defined and run from one to 100 times. The results of completion times are taken for FIFO and New algorithms. The results are shown with Figure 4.3. and Figure 4.4.

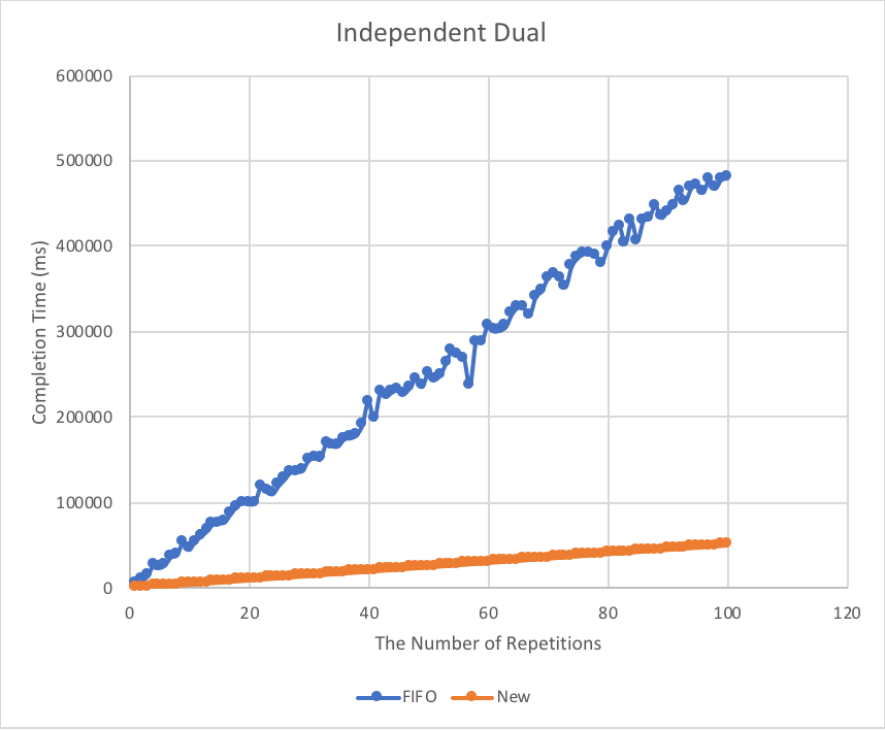


Figure 4.3. Completion time statistics for Independent Dual

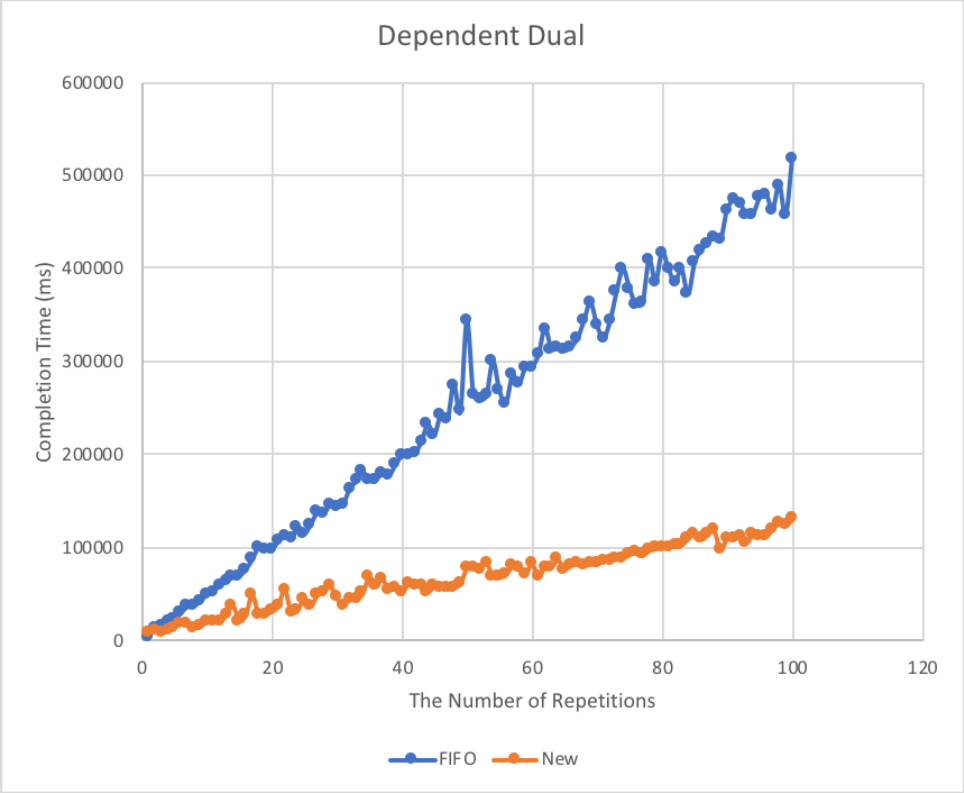


Figure 4.4. Completion time statistics for Dependent Dual

The figures shows that the proposed model is better than FIFO. The independent dual graph indicates proposed method is ten times faster than FIFO. Also, in the dependent dual graph, it shows that proposed one is approximately five times faster than FIFO approximately. In addition to these findings, there is no change in completion time of the FIFO subjob. Because of that, dependency is no importance for the FIFO, FIFO always operates subjobs one by one. However, the proposed method operates some subjobs at the same time, so the algorithm becomes speedier. In addition, because of dependency, the proposed model works relatively slowly in dependent dual with respect to independent dual.

The other and last test environment contains one job with three subjobs which are downloading document from Google Drive subjob, making changes into document with Office program subjob and uploading document to Google Drive subjob. There is a dependency field like second environment. The main different between second and third environment is that there is one more subjob only. The results are graphed in Figure 4.5. and Figure 4.6.

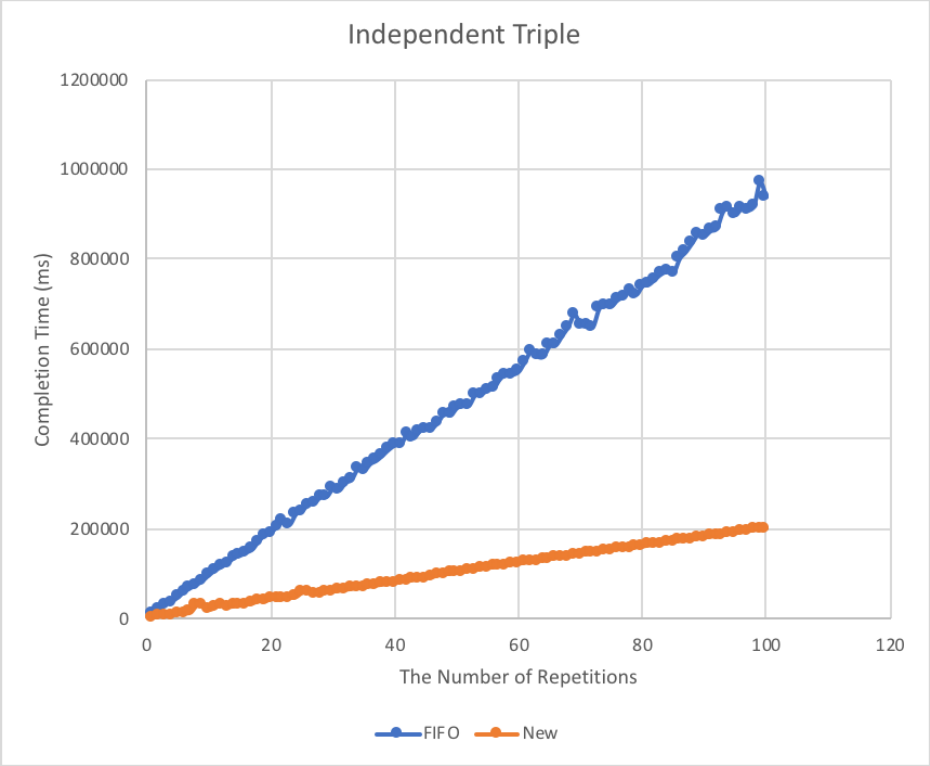


Figure 4.5. Completion time statistics for Independent Triple

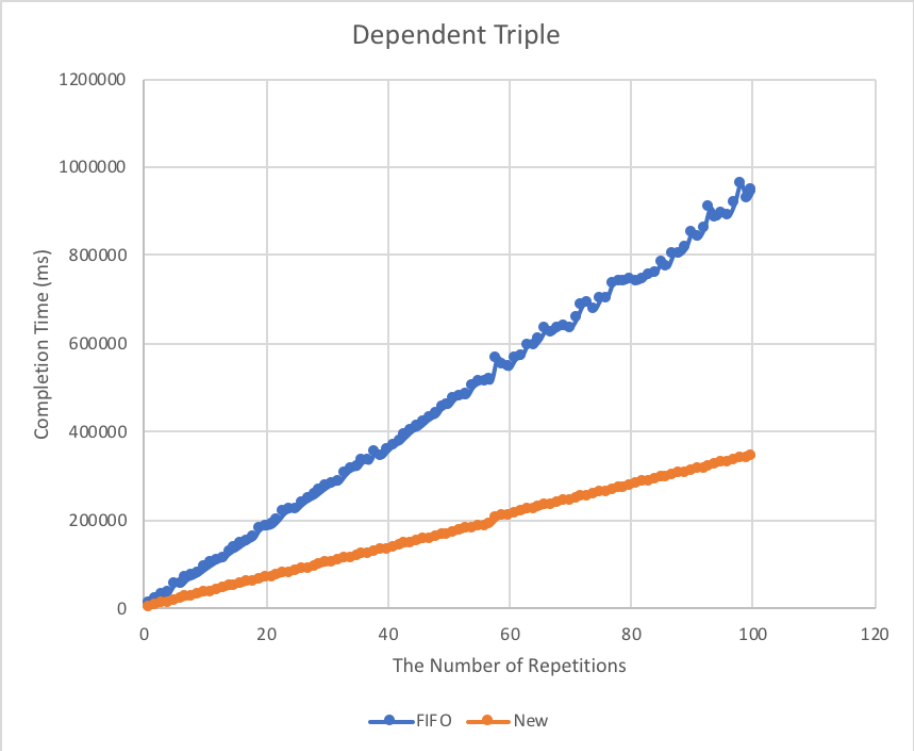


Figure 4.6. Completion time statistics for Independent Triple

Both of two figures show that proposed method is better than FIFO again. The independent triple graph indicates proposed one is five times faster than FIFO. Also, the dependent triple graph shows that proposed one is approximately two and half times faster than FIFO. This shows that completion time lines of FIFO and proposed one gets close to each other when number of subjobs increases as seen in second and third environments. In other words, completion time of proposed method increases more when number of subjobs increases. In the future work, it can be observed by collecting more data and comparing proposed algorithm with other models.

5. CONCLUSION

In this study, we proposed a new queuing model for heterogenous systems in rule based systems. The background research in this thesis shows that rule based distributed data-service models are inadequate. Therefore, we designed a new queueing model for queueing model. This proposed model is tested in the Graduate School as a tool. This tool is designed for Grad School Board decisions. The decisions are broken into tiny little pieces and their processing is done with this model. The rule based system is used for converting the decisions into job and, separating the jobs into subjobs. These subjobs are applied into queuing models.

In this model, there is a special queue for each service in heterogeneous system. These queues are dependent to each other sometimes but they always work simultaneously. This procedure gains more time and advantage of doing more operations. These advantages were proved in the tests results. Furthermore, this proposed model is needed to be tested with more test parameter. These tests can be tested many times with simulation tools such as MATLAB.

For a future study, this work can be extended by:

- In NLP part, new developed parts can be added.
- More types of rule can be included for rule based system.
- Number of heterogeneous systems can be increased.
- New features can be added on proposed queueing model.
- Type of test and test environment can be diversified to test and verify proposed model limits.

REFERENCES

1. Li K, Dewar RG, and Pooley RJ. R. Requirements capture in natural language problem statements, Heriot-Watt Technical Report HW-MACS-TR-0023, 2004.

2. Chowdhury GG. Natural language processing. *Annual Review of Information Science and Technology*. 2003; 37(1):51-89.

3. Sintoris K, Vergidis K. Extracting business process models using natural language processing (nlp) techniques. *2017 IEEE 19th Conference on Business Informatics (CBI)*. IEEE. 2017; 135-139.

4. Liddy ED, Liddy JH. An NLP approach for improving access to statistical information for the masses. *Center for Natural Language Processing.* 2001; 6.

5. Oflazer K. Turkish and its challenges for language processing. *Language Resources and Evaluation.* 2014; 48(4):639-653.

6. Aydoğan M, Karcı A. Turkish text classification with machine learning and transfer learning. *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE. 2019; 1-6.

7. Küçük D, Küçük D, and Arıcı N. A named entity recognition dataset for Turkish. *2016 24th Signal Processing and Communication Application Conference (SIU)*. IEEE. 2016; 329-332.

8. Özyurt Ö, and Köse C. Text-based Human-Computer interaction in Turkish. *2007 IEEE 15th Signal Processing and Communications Applications*. IEEE. 2007; 1-4.

9. Saygılı NŞ, Amghar T, Levrat B, and Acarman T. Taking advantage of Turkish characteristic features to achieve authorship attribution problems for Turkish. *2017 25th Signal Processing and Communications Applications Conference (SIU).* IEEE. 2017; 1-4.

10. Hatzisymeon G, Nikos H, Dimitris A, and Vasilis S. An architecture for implementing application interoperation with heterogeneous systems. *IFIP International Conference on Distributed Applications and Interoperable Systems.* Springer, 2005; 194-205.

11. Gama K, Lionel T, and Didier D. Combining heterogeneous service technologies for building an Internet of Things middleware. *Computer Communications*. 2012; 35(4):405-417.

12. Tsai CJ, Shian-Shyong T, and Yu-Co W. A new architecture of object-oriented rule base management system. *In Technology of Object-Oriented Languages and Systems*. 1999;31:200-203.

13. Luc M, and Zaccarin A. New rule-based framework for post-processing merging in video sequence segmentation. *In Image Processing, 2000 International Conference on*. 2000;1:327-330.

14. Jung JY. Generating rule-based executable process models for service outsourcing. In Information Technology: New Generations, 2009. *ITNG'09. Sixth International Conference on*. 2009;114-118.

15. Yong Z, Zhao J, and Xing C. An extensible framework for internet booking application based on rule engine. I*n Web Information Systems and Applications Conference.* 2009;139-142.

16. Shamim A, Hameed H, and Maqbool US. A framework for generation of rules from decision tree and decision table. *In Information and Emerging Technologies (ICIET), 2010 International Conference on*, 2010;1-6.

17. Jie Y, and Qiu L. Rule-Based business data processing in service coordination systems. *In Broadband and Wireless Computing, Communication and Applications (BWCCA), 2014 Ninth International Conference on*. 2014;317-320.

18. Daniel M, and Tomai N. Association-rules-based recommender system for personalization in adaptive web-based applications. *In International Conference on Web Engineering*. 2010; 85-90.

19. İşgüder HO. Optimization Of Loss Probability In The GI/M/n/0 Queueing Model With Heterogeneous Servers [dissertation]. İzmir: Dokuz Eylül University; 2013.

20. Mansouri W, Ali KB, Zarai F, and Obaidat MS. Radio resource management for heterogeneous wireless networks: Schemes and simulation analysis. *Modeling and Simulation of Computer Networks and Systems*. 2015; 767-792.

21. Lee Z, Wang Y, Zhou W. A dynamic priority scheduling algorithm on service request scheduling in cloud computing. *In Proceedings of 2011 International Conference on Electronic and Mechanical Engineering and Information Technology*. 2011;9:4665-4669.

22. Hasija M, Kaushik A, Kumar P. OM Algorithm for multi-level queue scheduling. *International Conference on Machine Intelligence and Research Advancement.* IEEE,2013; 564-568.

23. Thombare M, Sukhwani R, Shah P, Chaudhari S, and Raundale P. Efficient implementation of multilevel feedback queue scheduling. *In 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. 2016;1950-1954.

24. Rule-Based System. [cited 2019 1 July]. Available from: https://www.sciencedirect.com/topics/engineering/rule-based-system.

APPENDIX A: USER LOGIN WINDOWS

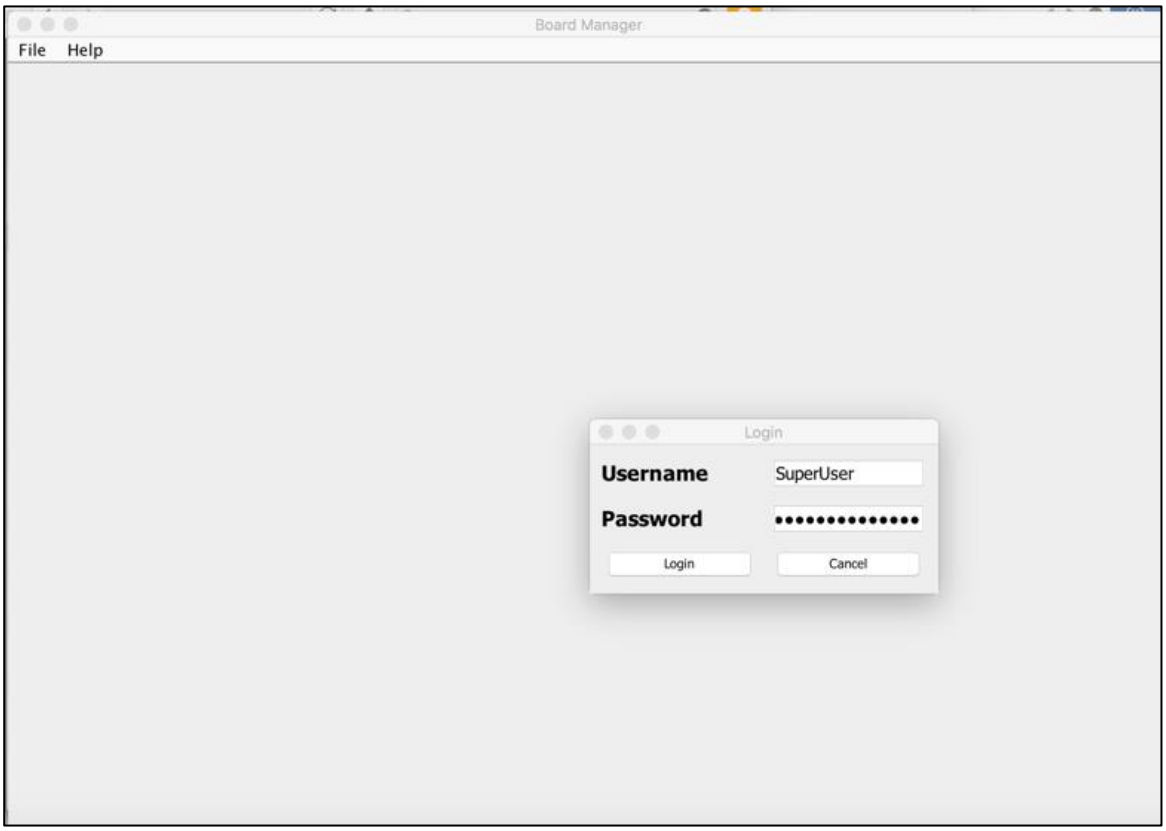


Figure A.1. The login window of the system

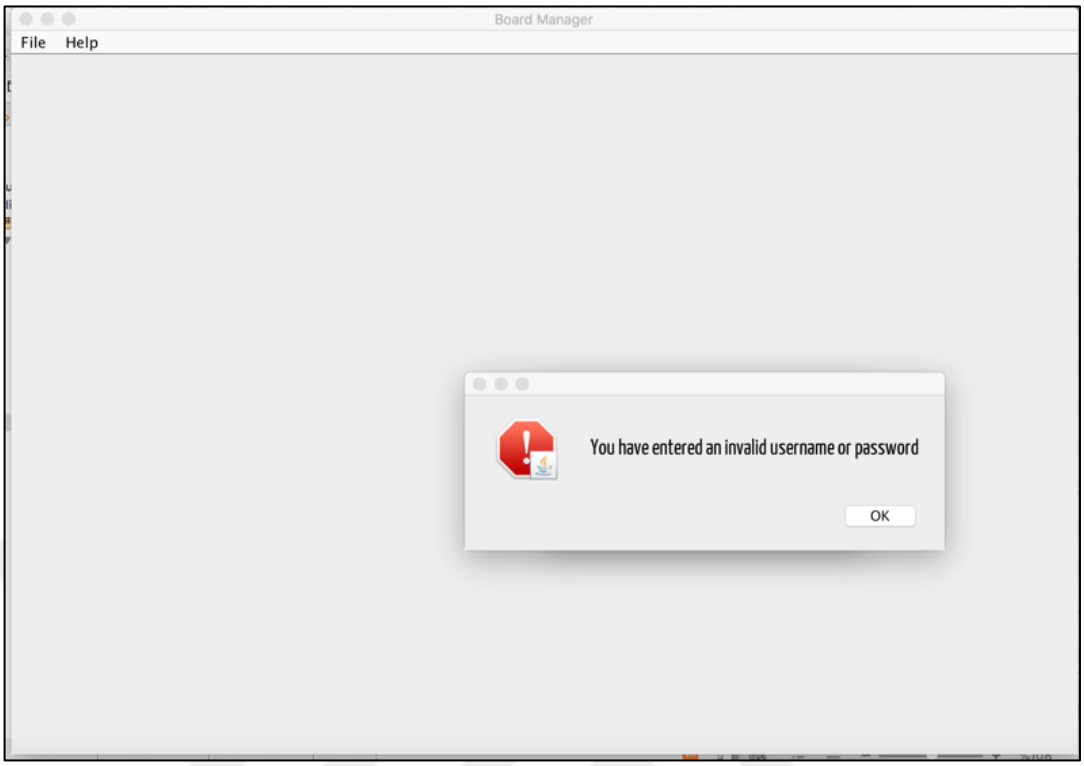


Figure A.2. Failure popup of the login window

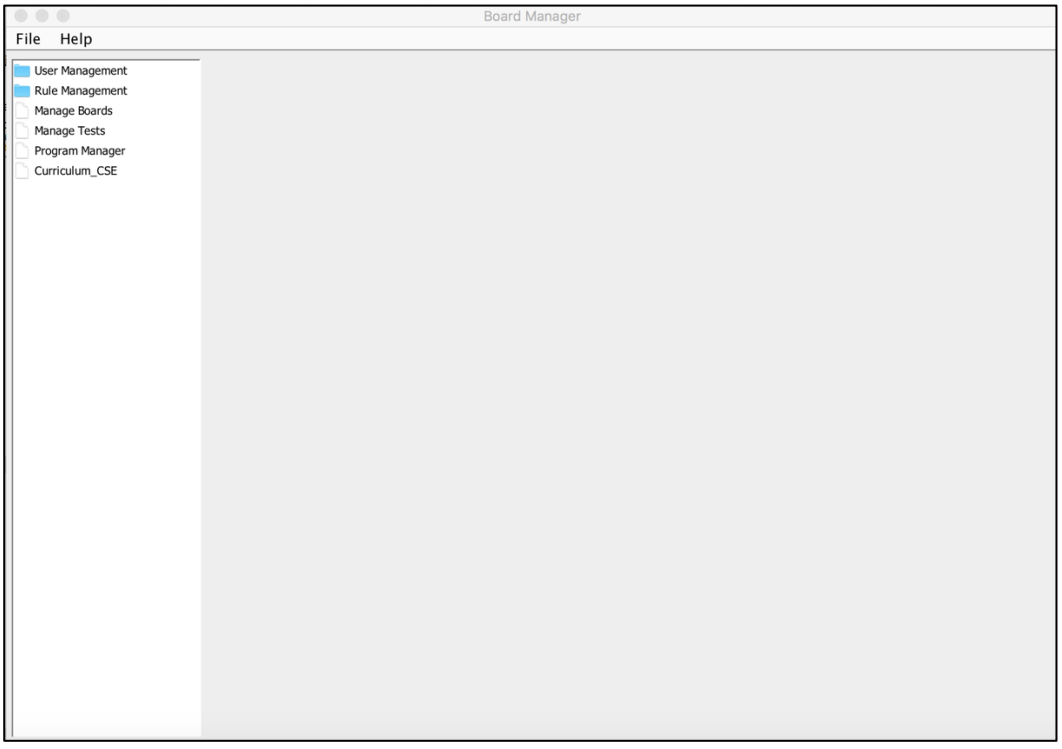


Figure A.3. General system

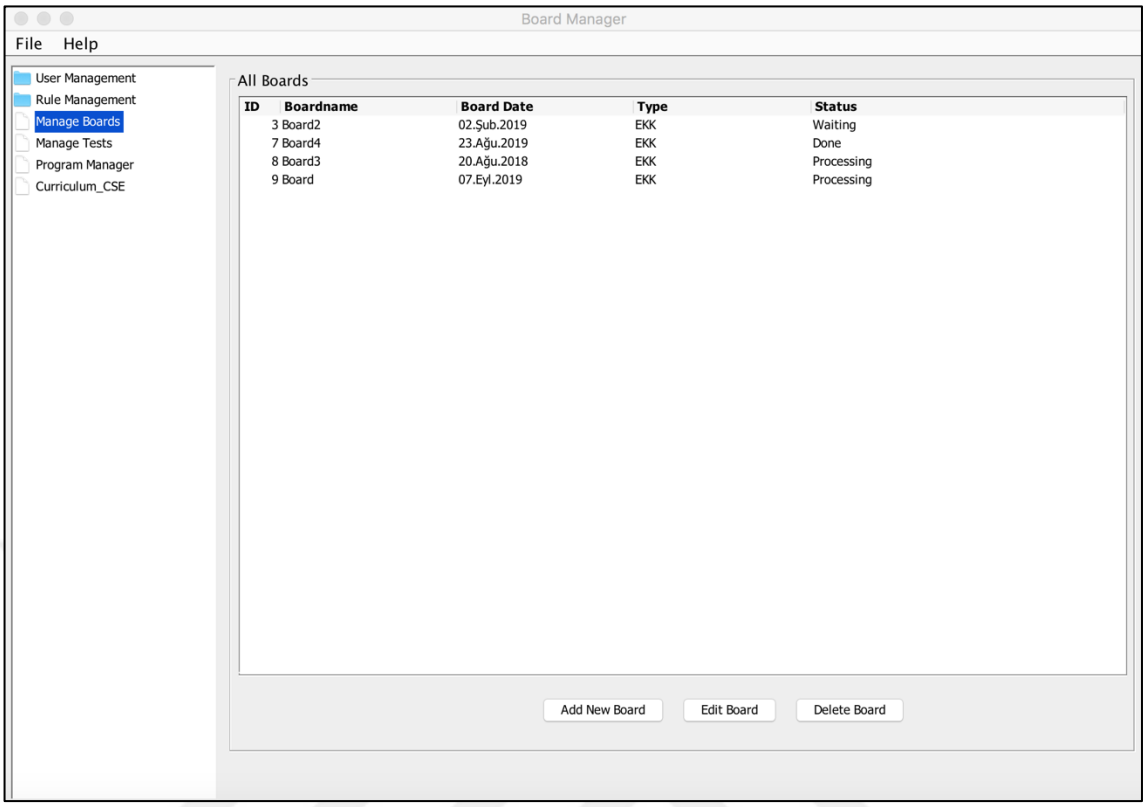


Figure A.4. Board management part

APPENDIX B: WINDOWS OF TOOL

The board management screen is shown as Figure B.1. When we add a new board, board properties are written to the window in Figure B.2. Name property describes the board. Date refers the board date. Type is the kinds of board that are taken different decisions. Operation expresses that the board will wait, process or do nothing. Permission can point which userrole to do this operation. Process refers the types of processing that are nothing(without queue), FIFO, new process (new queuing system). Lastly, document helps to take the board document from Google Drive.

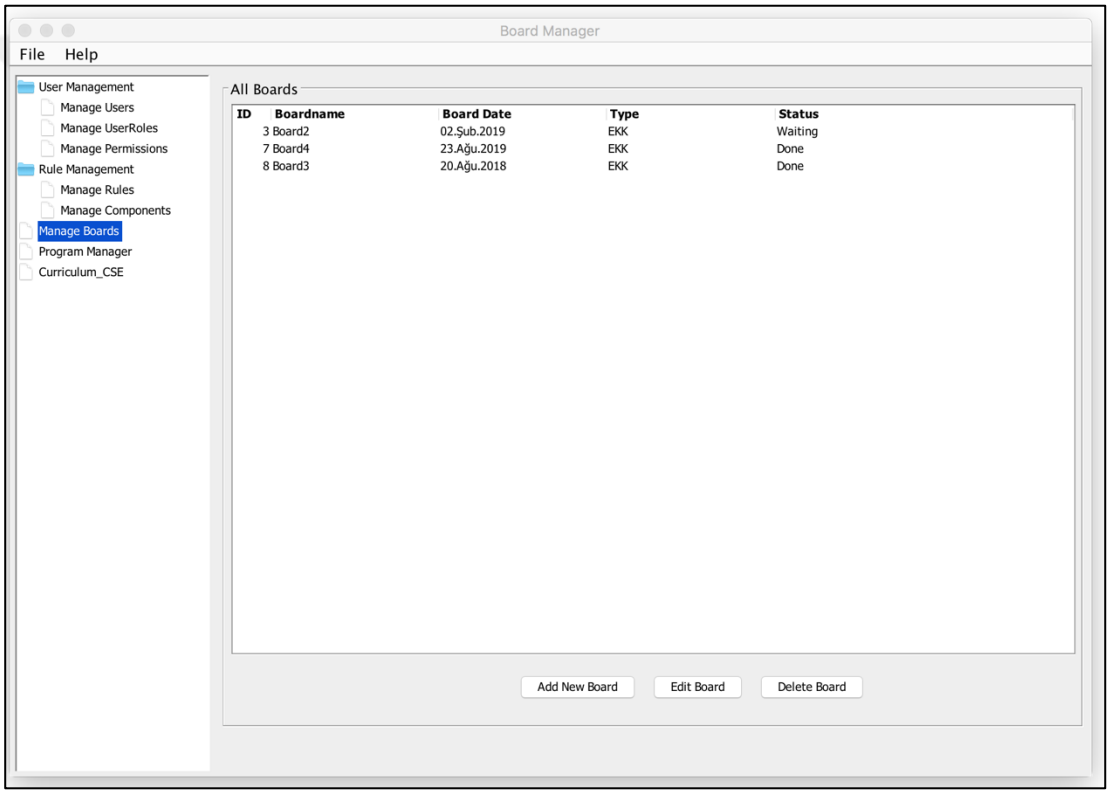


Figure B.1. Board management system

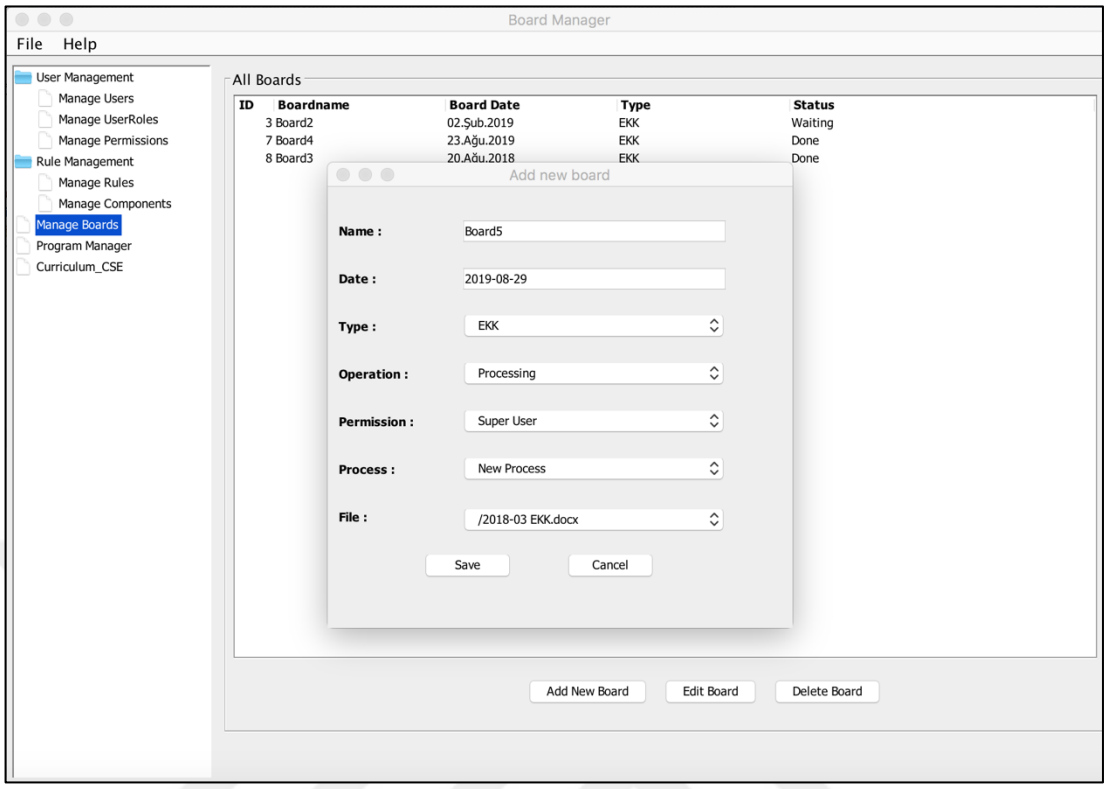
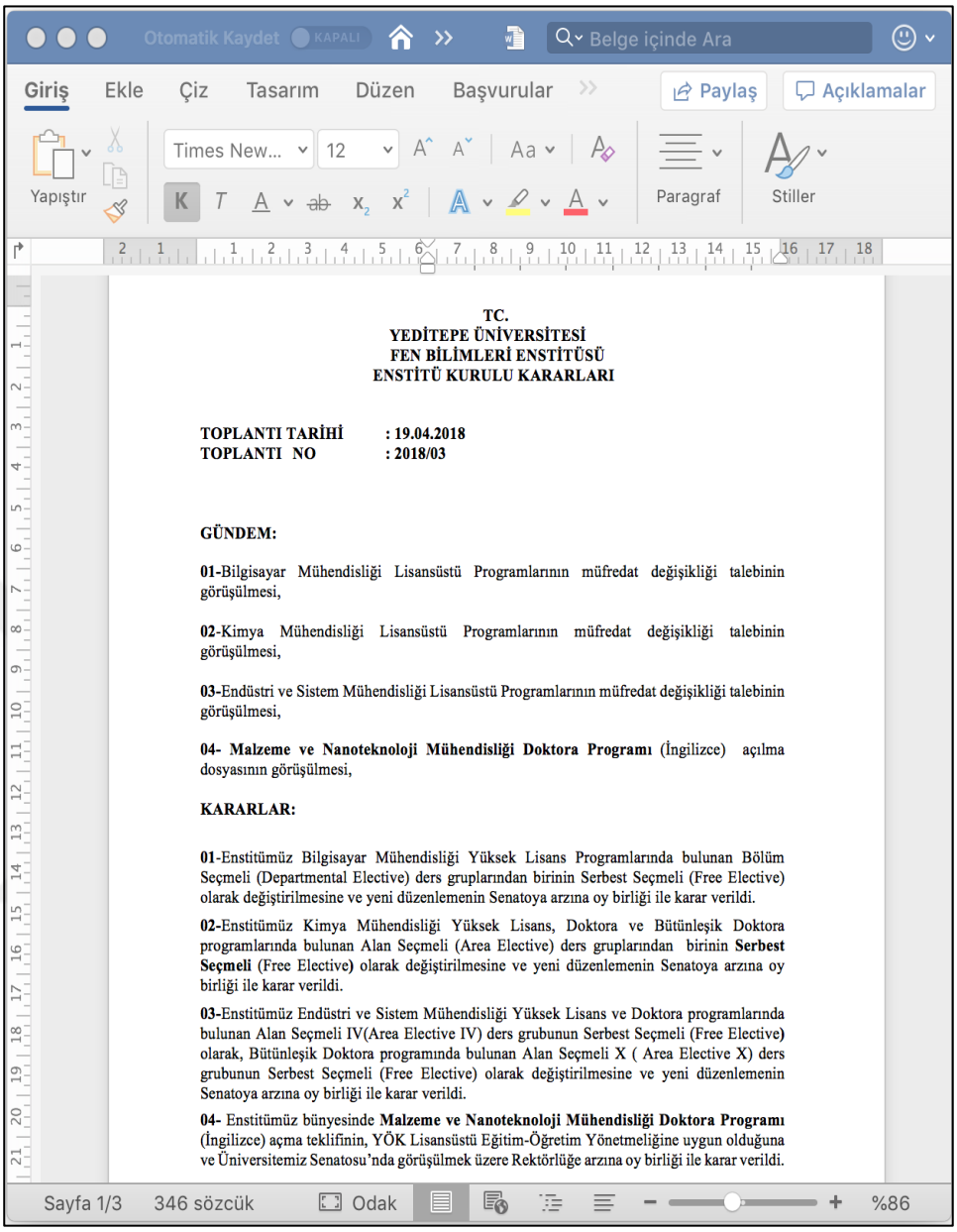


Figure B.2. Adding a new board

After save button is pressed with processing operation in add new board, all decisions are processed into board. We show an example board document in Figure B.3. In this board, there are “Change Curriculum” and “Open Program” processes. For these processes, Google Drive, Office web services are needed to be done, also curriculum change form document and curriculum document are used in program (Figure B.4. and Figure B.5.). When we apply the adding board, we observe the program in Figure B.6. So, we can investigate the results from the Google Drive folder. In these results, we can see the changed currculum document with Figure B.7.



TC.
YEDİTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENSTİTÜ KURULU KARARLARI

TOPLANTI TARİHİ : 19.04.2018
TOPLANTI NO : 2018/03

GÜNDEM:

01-Bilgisayar Mühendisliği Lisansüstü Programlarının müfredat değişikliği talebinin görüşülmesi,

02-Kimya Mühendisliği Lisansüstü Programlarının müfredat değişikliği talebinin görüşülmesi,

03-Endüstri ve Sistem Mühendisliği Lisansüstü Programlarının müfredat değişikliği talebinin görüşülmesi,

04- Malzeme ve Nanoteknoloji Mühendisliği Doktora Programı (İngilizce) açılma dosyasının görüşülmesi,

KARARLAR:

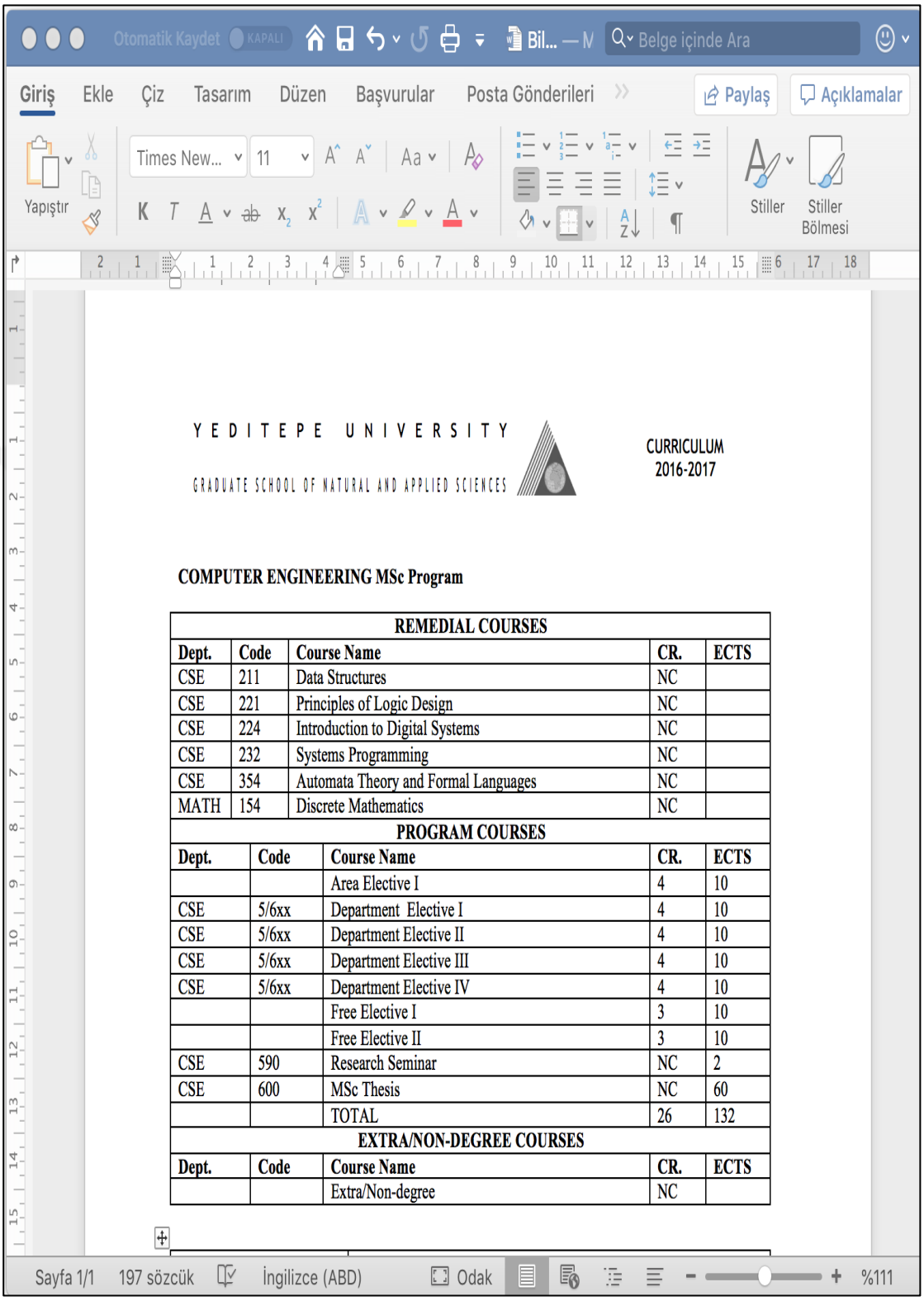
01-Enstitümüz Bilgisayar Mühendisliği Yüksek Lisans Programlarında bulunan Bölüm Seçmeli (Departmental Elective) ders gruplarından birinin Serbest Seçmeli (Free Elective) olarak değiştirilmesine ve yeni düzenlemenin Senatoya arzına oy birliği ile karar verildi.

02-Enstitümüz Kimya Mühendisliği Yüksek Lisans, Doktora ve Bütünleşik Doktora programlarında bulunan Alan Seçmeli (Area Elective) ders gruplarından birinin **Serbest Seçmeli** (Free Elective) olarak değiştirilmesine ve yeni düzenlemenin Senatoya arzına oy birliği ile karar verildi.

03-Enstitümüz Endüstri ve Sistem Mühendisliği Yüksek Lisans ve Doktora programlarında bulunan Alan Seçmeli IV(Area Elective IV) ders grubunun Serbest Seçmeli (Free Elective) olarak, Bütünleşik Doktora programında bulunan Alan Seçmeli X (Area Elective X) ders grubunun Serbest Seçmeli (Free Elective) olarak değiştirilmesine ve yeni düzenlemenin Senatoya arzına oy birliği ile karar verildi.

04- Enstitümüz bünyesinde **Malzeme ve Nanoteknoloji Mühendisliği Doktora Programı** (İngilizce) açma teklifinin, YÖK Lisansüstü Eğitim-Öğretim Yönetmeliğine uygun olduğuna ve Üniversitemiz Senatosu'nda görüşülmek üzere Rektörlüğe arzına oy birliği ile karar verildi.

Figure B.3. Board document



YEDITEPE UNIVERSITY

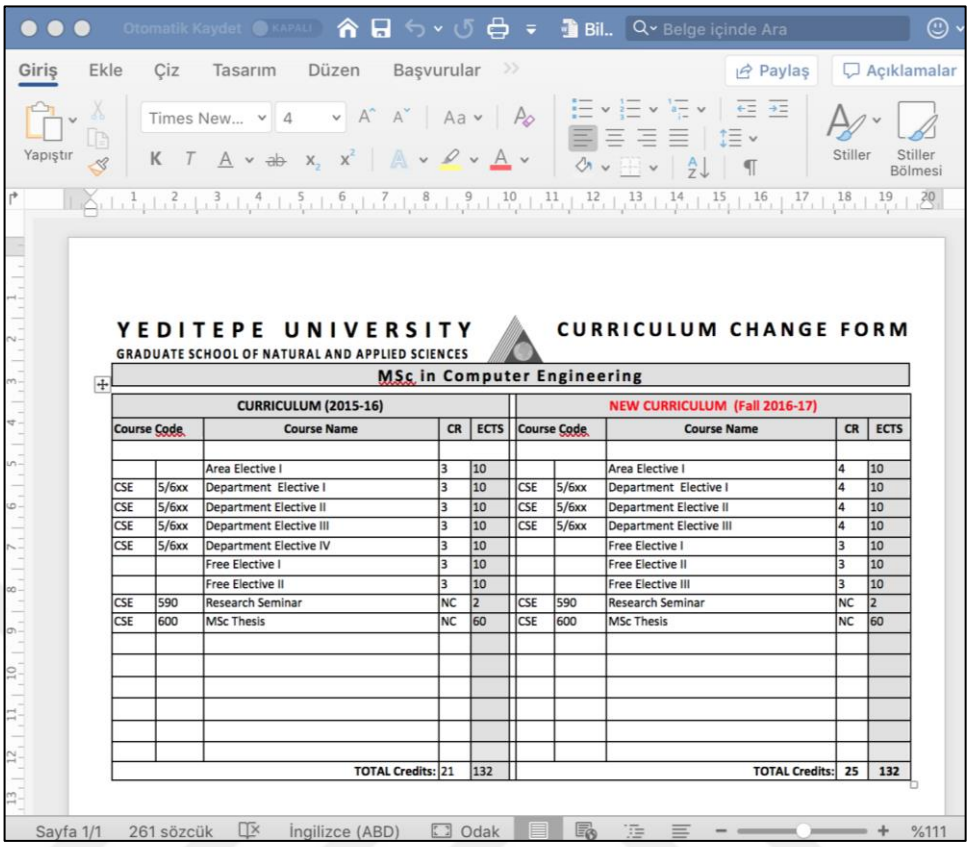
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CURRICULUM
2016-2017

COMPUTER ENGINEERING MSc Program

REMEDIAL COURSES				
Dept.	**Code**	**Course Name**	**CR.**	**ECTS**
CSE	211	Data Structures	NC	
CSE	221	Principles of Logic Design	NC	
CSE	224	Introduction to Digital Systems	NC	
CSE	232	Systems Programming	NC	
CSE	354	Automata Theory and Formal Languages	NC	
MATH	154	Discrete Mathematics	NC	
PROGRAM COURSES				
Dept.	**Code**	**Course Name**	**CR.**	**ECTS**
		Area Elective I	4	10
CSE	5/6xx	Department Elective I	4	10
CSE	5/6xx	Department Elective II	4	10
CSE	5/6xx	Department Elective III	4	10
CSE	5/6xx	Department Elective IV	4	10
		Free Elective I	3	10
		Free Elective II	3	10
CSE	590	Research Seminar	NC	2
CSE	600	MSc Thesis	NC	60
		TOTAL	26	132
EXTRA/NON-DEGREE COURSES				
Dept.	**Code**	**Course Name**	**CR.**	**ECTS**
		Extra/Non-degree	NC	

Figure B.4. Curriculum document



YEDITEPE UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CURRICULUM CHANGE FORM

MSc in Computer Engineering

CURRICULUM (2015-16)					NEW CURRICULUM (Fall 2016-17)				
Course Code		Course Name	CR	ECTS	Course Code		Course Name	CR	ECTS
		Area Elective I	3	10			Area Elective I	4	10
CSE	5/6xx	Department Elective I	3	10	CSE	5/6xx	Department Elective I	4	10
CSE	5/6xx	Department Elective II	3	10	CSE	5/6xx	Department Elective II	4	10
CSE	5/6xx	Department Elective III	3	10	CSE	5/6xx	Department Elective III	4	10
CSE	5/6xx	Department Elective IV	3	10			Free Elective I	3	10
		Free Elective I	3	10			Free Elective II	3	10
		Free Elective II	3	10			Free Elective III	3	10
CSE	590	Research Seminar	NC	2	CSE	590	Research Seminar	NC	2
CSE	600	MSc Thesis	NC	60	CSE	600	MSc Thesis	NC	60
		TOTAL Credits:	21	132			TOTAL Credits:	25	132

Figure B.5. Curriculum change form

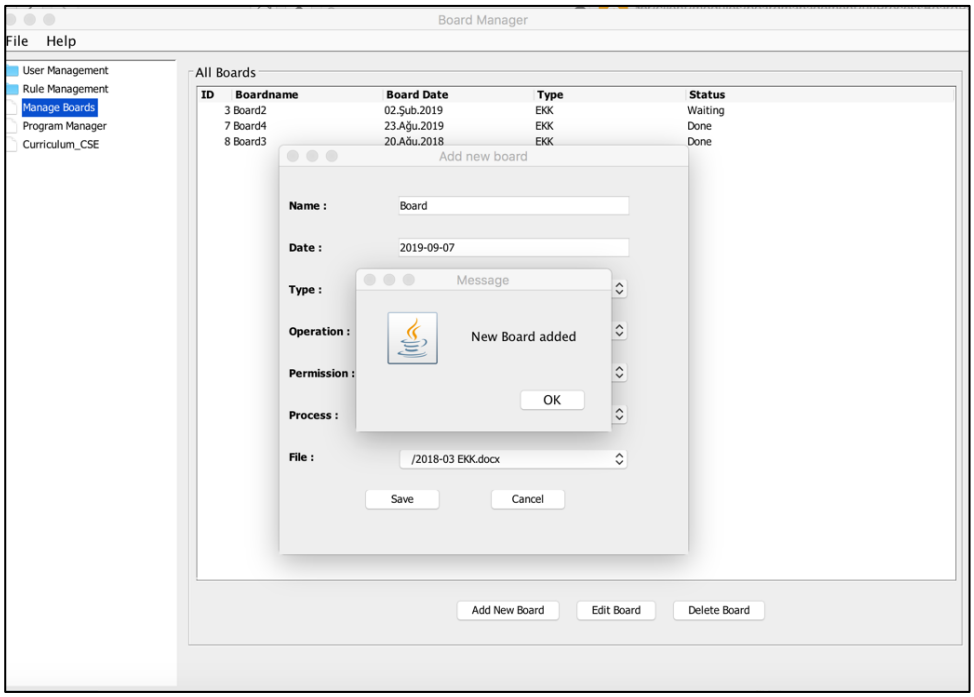
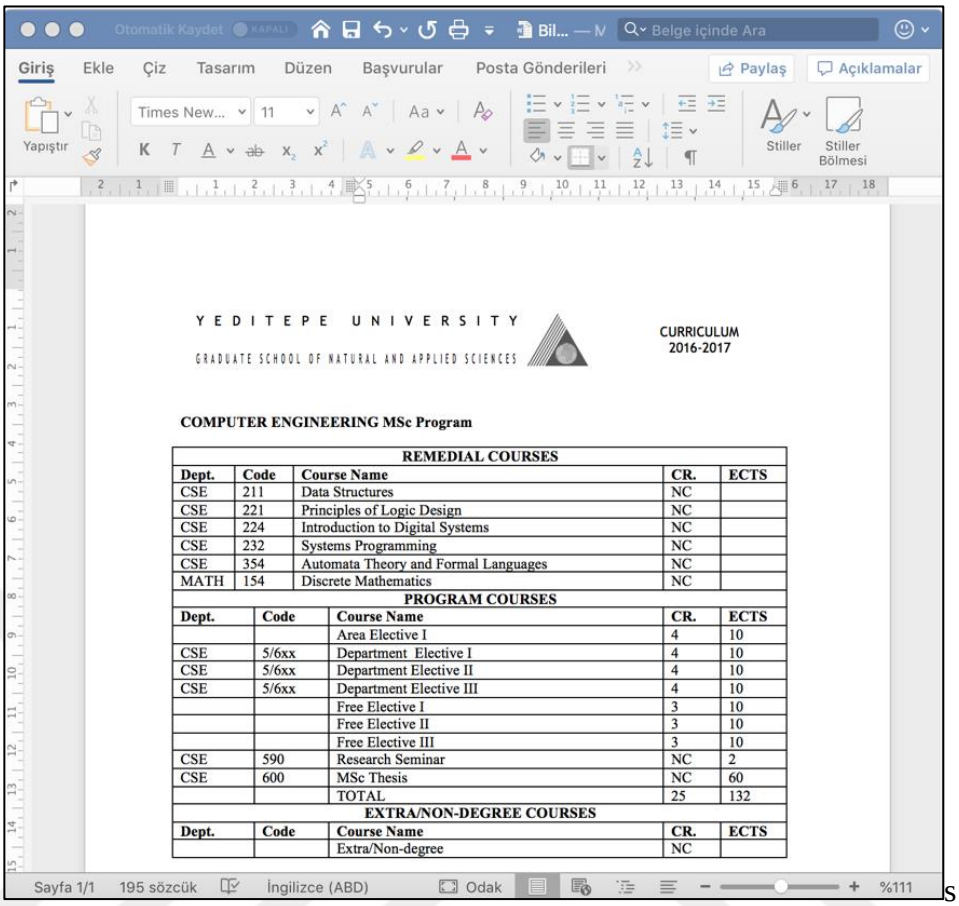


Figure B.6. New board is added in program



YEDITEPE UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CURRICULUM
2016-2017

COMPUTER ENGINEERING MSc Program

REMEDIAL COURSES				
Dept.	**Code**	**Course Name**	**CR.**	**ECTS**
CSE	211	Data Structures	NC	
CSE	221	Principles of Logic Design	NC	
CSE	224	Introduction to Digital Systems	NC	
CSE	232	Systems Programming	NC	
CSE	354	Automata Theory and Formal Languages	NC	
MATH	154	Discrete Mathematics	NC	
PROGRAM COURSES				
Dept.	**Code**	**Course Name**	**CR.**	**ECTS**
		Area Elective I	4	10
CSE	5/6xx	Department Elective I	4	10
CSE	5/6xx	Department Elective II	4	10
CSE	5/6xx	Department Elective III	4	10
		Free Elective I	3	10
		Free Elective II	3	10
		Free Elective III	3	10
CSE	590	Research Seminar	NC	2
CSE	600	MSc Thesis	NC	60
		TOTAL	25	132
EXTRA/NON-DEGREE COURSES				
Dept.	**Code**	**Course Name**	**CR.**	**ECTS**
		Extra/Non-degree	NC	

s

Figure B.7. Curriculum document after board processing