DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A RUN-TIME ENVIRONMENT FOR EXECUTION OF STATE MACHINES

by

Eminullah YAŞAR

August, 2019

İZMİR

A RUN-TIME ENVIRONMENT FOR EXECUTION OF STATE MACHINES

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering, Computer Engineering Program

by

Eminullah YAŞAR

August, 2019

İZMİR

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A RUN-TIME ENVIRONMENT FOR EXECUTION OF STATE MACHINES"** completed by **EMİNULLAH YAŞAR** under supervision of **PROF. DR. SÜLEYMAN SEVİNÇ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Süleyman SEVİNÇ

Supervisor

Prof. Dr. Nuri A Başoğlu

(Jury Member)

Asst. Prof. Dr. Faruk UTKU

(Jury Member)

Prof.Dr. Kadriye ERTEKİN

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I wish to thank Prof. Dr. Süleyman SEVİNÇ who is my advisor, Prof. Dr. Ali Rıza ŞİŞMAN who is medical consultant, Yiğit Ege MİMAN and my company and family for their supports.

Eminullah YAŞAR

A RUN-TIME ENVIRONMENT FOR EXECUTION OF STATE MACHINES

ABSTRACT

Aim of the thesis is to create an innovative database for creating and storing protocols for pre-diagnosis examination requests, to select the appropriate protocol from this database for a specific patient by the physician, to transform the selected protocol into a personalized finite state machine by the system and to execute all steps in the protocol by the system by interacting with patients, physicians and other stakeholders with the physician's start command. With this way, valuable time is gained to the physician, standardization and coordination is provided and unnecessary examination requests are excluded, while the necessary ones are included in the validation process.

To achieve this goal, a "computer language" model has been created to identify the pre-diagnostic protocols. This computer language is fully capable of expressing pre-diagnostic protocols and stages. A "user interface model" has been created so that pre-diagnosis protocol definitions can be retrieved from users and translated into computer language. This user interface model allows technically non-expert users to define pre-diagnostic protocols to the database. The system performs the execution of the protocol by producing a "Finite State Machine" with a timeout and timing functions, which can produce outputs as well as the inputs for a given patient and selected protocol. Many messages will be produced for the physician, patient or other stakeholders in the system. These messages can be in the form of text or audio-visual messages. By using the Google Cloud Messaging system, it was developed to transmit these messages to their recipients as a notification message after processing.

Keywords: Healthcare informatics, medical examinations, finite state machines, pre-diagnosis protocols

DURUM MAKİNALARININ İCRASI İÇİN ÇALIŞMA ZAMANI ORTAMI

ÖZ

Tezin amacı, ön-teşhise yönelik inceleme talepleri için protokoller oluşturulup saklanabilmesi için inovatif bir veritabanı oluşturulması, hekimin belirli bir hastası için bu veritabanından uygun protokolü seçmesi, seçilen bu protokolün sistem tarafından hastaya kişiselleştirilmiş bir sonlu durum makinasına dönüştürülmesi ve hekimin başla komutu ile birlikte, sistem tarafından protokoldeki tüm adımların hasta, hekim ve diğer paydaşlarla etkileşerek icra edilmesidir. Bu sayede hekime değerli zaman kazandırılmakta, standardizasyon ve koordinasyon sağlanmakta, gereksiz incelemeler dışarıda bırakılırken, gerekli olanlar verifikasyon sürecine dahil edilmektedir.

Bu amacı gerçekleştirebilmek için, ön-teşhis protokollerinin tanımlanmaları için bir "bilgisayar dili" modeli oluşturulmuştur. Bu bilgisayar dili ön-teşhis protokollerini ve aşamalarını tamamen ifade edebilecek güçtedir. Protokol tanımlamalarının kullanıcılardan alınıp, bilgisayar diline çevrilebilmesi için bir "kullanıcı arayüz modeli" oluşturulmuştur. Bu kullanıcı arayüz modeli teknik olarak uzman olmayan bir kullanıcıya ön-teşhis protokollerini veritabanına tanımlama imkanı vermektedir. Sistem tarafından belirli bir hasta ve seçilen protokol için, girdileri işleyebildiği kadar çıktıları da üretebilecek, zamanaşımı ve zamanlama fonksiyonlarına sahip olan bir "Sonlu Durum Makinası" üretilerek protokolün icrası gerçekleştirilir. Sistemde hekim, hasta veya diğer paydaşlara yönelik pek çok mesaj üretilecektir. Bu mesajlar metin şeklinde veya audio-visual mesajlar olabilir. Google Cloud Messaging sistemi kullanılarak, bu mesajların işlenmesinden sonra alıcılarına bir bildirim mesajı olarak iletilmesi geliştirilmiştir.

Anahtar kelimeler: Sağlık bilişimi, tıbbi muayeneler, sonlu durum makineleri, ön tanı protokolleri

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE

INTRODUCTION

As a result of the clinical examination of their patients in their daily practice, physicians form one or more pre-diagnoses (yet unconfirmed hypothesis) for their patients. The physician plans a series of medical examinations to confirm or reject these pre-diagnoses. These may include laboratory tests, radiological examinations, and other types of examinations. For the rejection or acceptance of the pre-diagnosis hypothesis, the necessary investigations are selected by the physician from various examination lists and followed up manually. For example, the TSH, FT4 and T3 tests used to observe thyroid function should be performed by staging. This staging is difficult for manual monitoring, so physicians have to order these tests together at the same time. As another example, OGTT tests used in the differential diagnosis(DDX) of type II diabetes should be performed by timing. This timing is left to the patient to be followed. It is very difficult to follow them manually and this leads to some clinical, financial and operational losses and inefficiencies. In existing practices, the selection, planning and realization of pre-diagnosis medical examinations are performed and coordinated by the physician. Some institutions have developed algorithms for pre-diagnosis validation and have published them as flow-chart diagrams. However, physicians must perform these algorithms themselves. That is very difficult process. For this reason, the aim of the thesis is to develop a system that will fully standardize and coordinate the selection, planning and staging of pre-diagnosis examination requests. The system was a real-time message-based distribution system that executes and coordinates algorithms that executed by physicians into a finite state machine. Thus, valuable time will be gained to the physician and important clinical, financial and operational benefits will be provided to health institutions.

In order to achieve this aim, answers were examined in previous studies for some questions such as “How many laboratory tests are really needed for patients?”, “How much of the laboratory tests are actually unnecessary?”, “What can be done to reduce unnecessary laboratory test requests with health informatics?” Answers of these

questions and more are examined in the literature review. In the beginning part of the task, a "computer language" model must be created to define the pre-diagnosis protocols. This computer language must be able to fully express the pre-diagnostic protocols and stages. Then, we need a "user interface model" for the protocol definitions to be retrieved from the users and translated into computer language. This user interface model allows technically non-expert users to define pre-diagnostic protocols to the database. Then, a transformation mechanism should be developed by the system to perform the execution of the protocol by producing a "Finite State Machine" with timeout and timing functions, which can produce outputs as well as the inputs of a given patient and selected protocol. Finally, many messages are produced to the physician, patient or other stakeholders in the system. These messages can be in the form of text or audio-visual messages. Using the Google Cloud Messaging system, a message module is needed to transmit these messages to their recipients as a notification message after processing. All of these implementations can be seen on chapter two. Result of these implementations and some ideas about future studies can be seen on chapter three.

1.1 Background

Nowadays, internet technologies are highly developed. This development has shown itself in the area of health and health technologies. As a result of these developments, special laboratory information management systems, hospital information management systems, picture archiving and communication systems and many health technologies have been developed. Especialy the laboratory ordering systems, it is possible to perform a test request for patients and follow up the test requests and test results. In these traditional systems, physicians must request laboratory tests one by one for each patient to assist in the diagnosis of the disease. This event increases workload of physicians, causes unnecessary test requests for patients, causes significant costs for hospitals and affect the quality of the patient's test results.

In some institutions, they have developed algorithms for pre-diagnosis validation and have expressed them as flow diagrams. However, these algorithms must be performed by physicians. Some of these algorithms are available at Mayo Clinic (Algorithms, 2018) and Arup Consult (Arup Algorithms, 2018). These algorithms have helped me with the way in which a new algorithm can be generated and the rules to follow.

Therefore, I propose to develop a system that transforms the established protocols for pre-diagnosis examinations into a personalized finite state machine and executes all steps in the protocol by interacting with patients, physicians and stakeholders with the physician's start command.

1.2 Statement of the Problem

As a result of the clinical examinations performed by physicians in their daily practice; physicians create one or more pre-diagnosis for the patient. The selection, planning and realization of medical examinations for pre-diagnosis are carried out by physicians over long drop-down lists. Tests which are unnecessary or need to be done in the next stages, cause significant costs for hospitals and affect the quality of the results.

Also during this procedure, the test can be requested without any need for the patient and the same test requests can be made from different physicians for the same patient. Medical examination spending are shown in the following Table 1.1 between 2002 and 2013 to TUIK data (Atasever, 2014).

Table 1.1 Medical spending between 2002 and 2013 in Turkey (Atasever, 2014)

	2002	2004	2006	2008	2009	2010	2011	2012	2013	2002-2013 Increase
TL	340	585	830	1,845	2,086	2,422	2,681	3,056	3,339	8,8
Prices in 2013, TL	947	1,197	1,433	2,652	2,822	3,018	3,138	3,285	3,339	2,5
USD	226	411	580	1,427	1,348	1,614	1,605	1,705	1,756	6,8
SGP USD	557	722	981	2,073	2,295	2,575	2,716	2,903	3,041	4,5
Share in Health Expenditures (%)	1.8	1.9	1.9	3.2	3.6	3.9	3.9	4.1	4.0	
Share in GDP (%)	0.10	0.10	0.11	0.19	0.22	0.22	0.21	0.22	0.21	

When the table 1.1 is examined, it was seen that the loss of 193.16 million $ for medical examinations in 2013

1.3 Methodology and Tools

Development language was selected Node.js 10.13.0, JavaScript, HTML5 and css3 for fast and secure data that is easily accessible, accessible from many platforms, providing users with an effective interface to provide flow. Development tool was selected Microsoft Visual Studio Enterprise 2017 and testing library was selected Mocha which compatible with Microsoft Visual Studio.

Choosing the database to use in the project; security, speed, installation, support and licensing fee, compliance with the outputs of the project, the convenience of storing pre-diagnosis protocols to be established and taking into account the criteria of the database that can best model real situations, non-relational MongoDB in the database category is selected. I decided to use free and open source version MongoDB Community Server 4.0.3 and Robomongo 1.2.1.0 to use efficiently.

1.4 Scope of the Study

Thanks to this thesis, physicians will be able to easily identify pre-diagnosis protocols they use in their daily life. The suitability of the algorithms will be checked by the system. Each step of protocol drawing has validation functions. Suitable algorithms are converted Finite State Machine (FSM) by the system. Users can see validity date for algorithm.

Defined pre-diagnosis protocols will be listed to enable the user to select the appropriate protocol. User will be able to make changes to this protocol if it was not used. If it was used before, another version of algorithm stored in database.

This project is a TUBITAK project that achieved with sponsorship of Labenko Bilisim Inc. and all rights of the system belongs to Labenko Bilisim Inc. The transformation of algorithms to Finite State Machine (FSM) and the execution of the operations will be done. The system will automatically make all examination requests determined by a single request and the system will make relevant decisions and inform the users or stakeholders where necessary.

1.5 Literature Review

Weidenhaupt and Lagor (2008) uses the information in the hospital information management system. This information; patient information, pre-determined functional rules of the disease, can be predicted as a result of these procedures are diagnostic information. Using this information, an ideal workflow is created to solve the patient's problem (Figure 1.1). The resulting workflow also includes the estimated time information, except for this information. This system gives doctors a recommendation about the way they can follow.

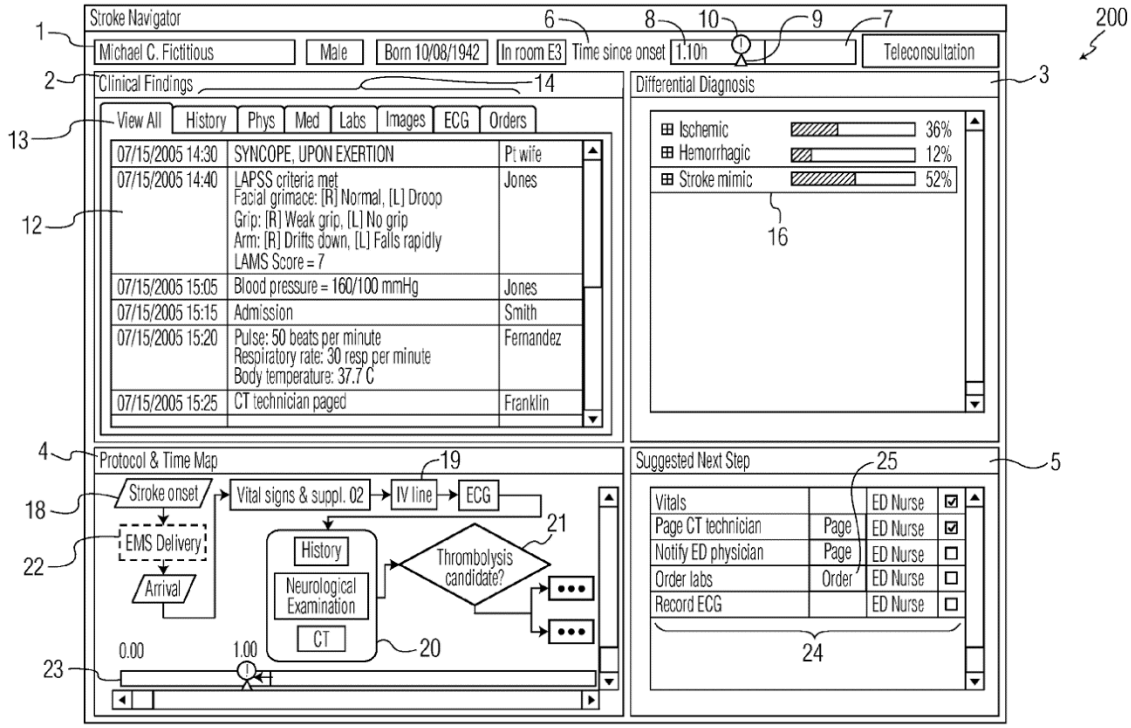


Figure 1.1 Workflow screen created by Weidenhaupt and Lagor (2008)

Loutzenhiser and Masarie (2007) uses a large electronic patient database. In this database appropriate laboratory techniques and drug interactions are determined. According to this plan, some drugs are prescribed automatically and a laboratory test or a medical procedure is ordered.

Rosales et al. (2006) provides a road map to be followed in general. Instead, he predicts the next steps that the physician will take. With these estimations, it is aimed to eliminate uncertainty in both diagnosis and treatment options. It also uses additional data such as risk, cost and convenience when making these estimates. Figure 1.2 shows a sample drawing drawn with Rosales et al. (2006).

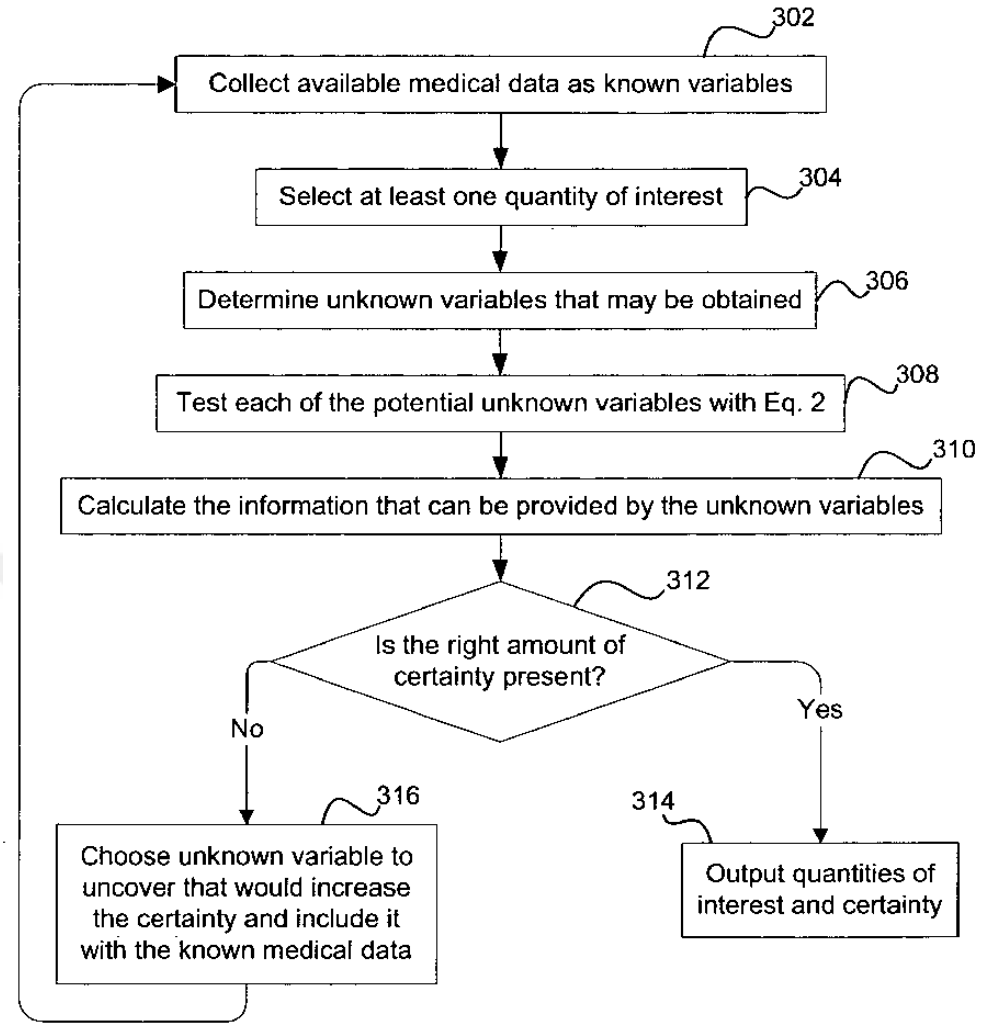


Figure 1.2 Workflow created by Rosales et al. (2006)

It was observed that the system did not make an estimate and it was completely different from Rosales et al. (2006) because it was followed by a general algorithm determined by the physician. The flow generated by the system in Weidenhaupt and Lagor (2008) will be determined by the physician in my thesis. By examining this flow, the system will generate FSM and automatically determine the paths to be monitored. This way can make requests automatically. My project aims to make the physician start working only by making a request, informing the physician in the intermediate steps and informing the physician about the final situation.

The results of a survey conducted with the participation of 412 physicians at the University of Pennsylvania Medical Faculty were presented. According to the results of this research; They accepted that 35% of physicians requested unnecessary / inappropriate laboratory tests every day, 46% twice or three times a week and 11% once a week. Survey result can be seen below (Nachamkin, 2015).

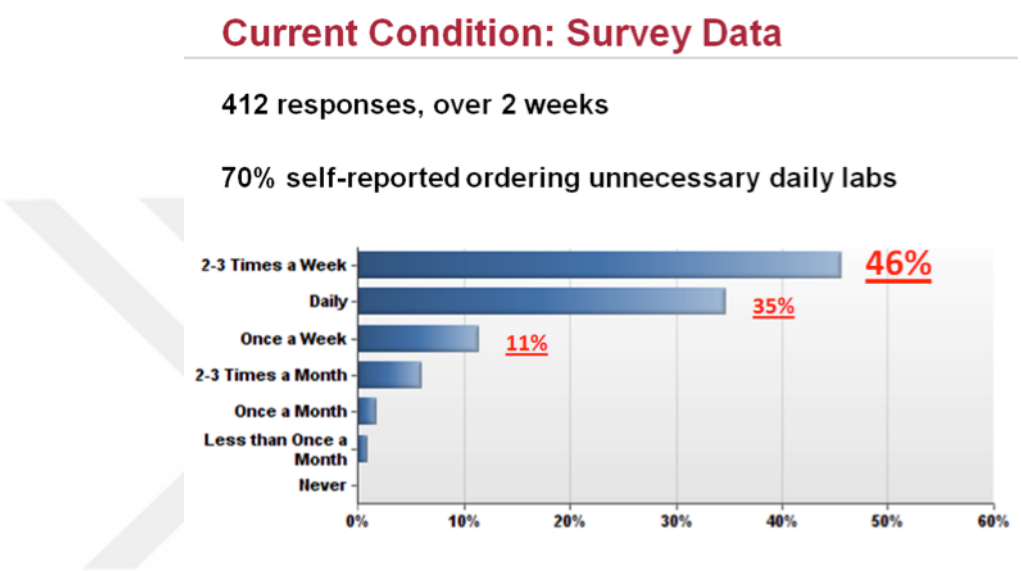


Figure 1.3 Survey result (Nachamkin, 2015)

The laboratory is often asked to find a way to control laboratory test use, but the way to check the test order begins with the provider. If providers cannot learn to do testing only when clinically necessary, there is not much that can be done in the laboratory only after blood samples are taken. It is clear that reducing unnecessary laboratory tests will increase patient satisfaction (less needle rods, fewer sample tubes), reduce negative impacts from false positive tests, reduce hospital waste and reduce health costs (Nachamkin, 2015).

It is concluded that 719 (41%) of 1750 test requests evaluated according to pre-determined criteria are unnecessary. Since the study was performed in intensive care

units, the problem was not limited to any unit in the hospital. Counts and percentage ratios can be seen by days of the week on Figure 1.2 (Oliveira, 2014).

Distribution of results of laboratory tests ordered for patients admitted in intensive care unit, sorted by days of the week

Day of the week	Order (*n*)	Normal tests1 (%)	Alterated tests1 (%)
Sunday	128	824 (13.3%)	859 (14.2%)
Monday2	1562	1.118 (18.1%)	1.043 (17.3%)
Tuesday	118	734 (11.9%)	812 (13.5%)
Wednesday	119	851 (13.8%)	781 (12.9%)
Thursday	136	883 (14.3%)	848 (14.0%)
Friday	130	825 (13.4%)	886 (14.7%)
Saturday	124	943 (15.3%)	808 (13.4%)

Figure 1.4 Alterared test order (Oliveria, 2014)

The result of the research, significant number of unnecessary testing was found, with occurrence pattern on Monday and not dependent on age or length of stay. The data indicate the need to implement guidelines or protocols for ordering laboratory tests, which have proven effective in helping the prescribing professional in their clinical practice, and contribute to optimize for health care spending (Oliveira, 2014).

As can be seen from the research, it is seen that the problem is not limited to any unit in the hospital since the research study was carried out in intensive care units. The problem is a general problem. Research shows that for the solution of the problem, proven rules or protocols should be applied during the test request.

In another research study, on the 235,440 test, 11% of them were revealed as unnecessary / unwarranted requests. Counts and percentage ratios can be seen by sections of the hospital on Figure 1.3 (Khalifa, 2014).

Table 3. A sample table to compare ordering index of the top ordering physicians to their specialties, showing percentage of increased tests and calculated over-utilization above the average for their specialties.

Section	Index	Physician	Patients	Tests	Physician Index	Increase	Unnecessary Tests
Medical Oncology	18.9	Physician 1	110	3,251	29.6	56.6%	1,175
Adult Critical Care	16.2	Physician 1	72	2,463	34.2	111.1%	1,296
		Physician 2	84	2,859	34	109.9%	1,497
Neonatology/Perinatology	12.6	Physician 1	62	1,422	22.9	81.7%	640
Internal Medicine	12.2	Physician 1	246	5,421	22	80.3%	2,415
		Physician 2	195	4,133	21.2	73.8%	1,755
Pediatric Hematology	12	Physician 1	37	3,428	92.6	671.7%	2,984
		Physician 2	29	2,325	80.2	568.3%	1,977
		Physician 3	39	3,039	77.9	549.2%	2,571
		Physician 4	33	2,490	75.5	529.2%	2,094
		Physician 5	7	375	53.6	346.7%	291
		Physician 6	28	648	23.1	92.5%	311
		Physician 7	27	529	19.6	63.3%	205
Cardiac Surgery	11.7	Physician 1	102	2,625	25.7	119.7%	1,430
Nephrology	9.4	Physician 1	102	1,713	16.8	78.7%	755
Gynecology/Gynecologic	8.9	Physician 1	75	1,712	22.8	156.2%	1,044
General Pediatrics	8.5	Physician 1	161	2,667	16.6	95.3%	1,301
Adult Cardiology	8.3	Physician 1	219	2,948	13.5	62.7%	1,136
Reproductive	8.2	Physician 1	30	373	12.4	51.2%	126
Neurosurgery	7.2	Physician 1	67	864	12.9	79.2%	382
		Physician 2	133	1,447	10.9	51.4%	491
All Hospital	**10.8**	**All**	**21,801**	**235,440**	**10.8**	**11%**	**25,876**

Figure 1.5 Unnecessary laboratory tests (Khalifa, 2014)

The study recommended two types of approaches; a user approach and a system approach, where user approach includes different types of orientation, education and training of physicians and other users on the importance and ways of decreasing unnecessary lab test ordering, mainly through avoiding unnecessarily repeated tests, while system approach includes the implementation of different computerized clinical decision support interventions that would help during the order entry process to alert and remind users with the potential of ordering an unnecessarily repeated lab test. As a result of the study, the author recommends the use of disease-specific test order guides and the use of order paths that may be effective for target areas or target diseases (Khalifa, 2014).

The author's view clearly states that there is a need for a systemic test system based on diseases.

In a research study entitled “Factors contributing to inappropriate ordering of tests in an academic medical department”, 426 patients were followed-up for 6 months and

it was determined that 67.9% of them could not contribute to patient management in the test requests(24482 test) for these patients (Miyakis, 2006).

During work, senior trainees ordered more laboratory examinations, but the percentage of avoidable tests requested by junior trainees was higher. A moderate and disparate level of trainees' awareness about the cost of common laboratory examinations was disclosed. The avoidable tests/patient/day were significantly decreased after the intervention (mean 1.58, p = 0.002), but containment of unnecessary ordering of tests gradually waned during the semester after the intervention. Result of intervention can be seen on Figure 1.4 (Miyakis, 2006)

	Before intervention (March–August 2003)	**After intervention (November 2003–April 2004)**
Patients, n	426	214
Mean age (range)	67.8 (15–98)	66.1 (16–104)
Hospitalisation days (mean, range)	3845 (9.0, 2–50)	1797 (8.4, 2–47)
Males (%)	212 (49.7)	110 (51.4)
Multiple problems* (%)	50 (11.7)	22 (10.3)
Died or undiagnosed (%)	41 (9.6)	27 (12.6)
Emergency department† (%)	341 (80)	176 (82.2)

Figure 1.6 Before and after intervention (Miyakis, 2006)

As a result of the study, the evaluation of a large number of unnecessary laboratory investigations, which were ordered during routine practice, by interns of an academic medical department were used to determine the factors that were predisposed to laboratory overuse. A feedback approach based on the results of this assessment has caused an important but temporary limitation of inappropriate test sequence behavior. It was observed that reordering of the tests did not resolve the real problem

permanently. This observation proves that the aim of the thesis is to create a permanent and real solution to the problem. The method to be developed by the thesis, interns will also be able to benefit from the protocols prepared by the specialists in their field, they will be able to improve themselves by learning from the knowledge of experts.

In another study, it was observed that the total number of test requests made by the physicians decreased by 25% as a result of the training activities aimed at changing the daily practices of the physicians (May, 2006).

Comparison of Inpatient Statistics for Fiscal Year 2002-2003 vs 2003-2004*

	2002-2003	2003-2004	Absolute (%) Difference†	Ordering Rate Ratio (95% CI)‡	P
Inpatient admissions	17,850	17,553	−297 (−1.7)	—	—
Inpatient admission days	115,715	114,936	−779 (−0.7)	—	—
Inpatient days/inpatient	6.48	6.55	0.07 (0.1)	—	—
Total inpatient laboratory tests§	604,847	532,208	−72,639 (−12.0)	—	—
Average laboratory tests/inpatient	33.9	30.3	−3.6 (−10.6)	—	—
Average laboratory tests/inpatient day	5.2	4.6	−0.6 (−11.5)	0.89 (0.88-0.89)	<.0001
Total inpatient phlebotomies performed	80,294	63,087	−17,207 (− 21.4)	—	—
Average phlebotomies performed/inpatient day	0.69	0.55	−0.14 (−20.3)	0.79 (0.78-0.80)	<.0001
Total laboratory tests performed (inpatient and outpatient, including for ED)	1,112,689	1,142,958	30,269 (2.7)	—	—
Total outpatient laboratory tests performed	429,990	484,836	54,846 (12.8)	—	—
Total laboratory tests performed for ED	77,852	93,142	15,290 (19.6)	—	—

CI, confidence interval; ED, emergency department.
* Data are given as number unless otherwise indicated.
† 2003-2004 volume relative to 2002-2003 volume.
‡ Ordering rates (ie, average number of inpatient tests/inpatient day, average number of phlebotomies performed/inpatient day) were calculated for each fiscal year and compared using a ratio measure as described in the "Materials and Methods" section.
§ Derived from the Financial and Accounting Management Information System reports for fiscal years 2002-2003 and 2003-2004.

Figure 1.7 Decreasing result (May, 2006)

There was little change in the total number of inpatient admissions, inpatient admission days, or number of inpatient days per inpatient. In contrast, the total inpatient laboratory test volume decreased by 72,639 tests (12.0% decrease from the previous fiscal year). Statistically significant decreases were observed in the average number of laboratory tests per inpatient day and average number of phlebotomies performed per inpatient day (May, 2006).

After identified 5 tests commonly ordered on a recurring basis, a multidisciplinary committee developed a proposal to minimize unnecessary laboratory tests by

reconfiguring the electronic order function to limit laboratory test requests to occur singly or to recur within one 24-hour window. The proposal was implemented in June 2003. Comparison of fiscal year volume data from before (2002-2003) and after (2003-2004) implementation revealed 72,639 (12.0%) fewer inpatient tests, of which 41,765 (57.5%) were related directly to decreases in the 5 tests frequently ordered on a recurring basis. Because the electronic order function changes did not completely eliminate unnecessary testing and concluded that the decrease in inpatient testing represented a minimum amount of unnecessary inpatient laboratory tests (May, 2006).

Impact of Limiting Phlebotomy and Test Orders to 24 Hours on Previously Identified, Frequently Recurring Inpatient Laboratory Tests

	No. of Tests				
Test	**FY 2002-2003**	**FY 2003-2004**	**Absolute (%) Difference***	**Incidence Rate Ratio (95% C)†**	***P***
CBC count	162,039	149,953	–12,086 (–7.5%)	0.93 (0.92-0.94)	<.0001
Basic metabolic panel	98,350	88,690	–9,660 (–9.8%)	0.91 (0.90-0.92)	<.0001
Calcium	3,736	2,650	–1,086 (–29.1%)	0.71 (0.68-0.75)	<.0001
Magnesium	50,243	40,198	–10,045 (–20.0%)	0.81 (0.79-0.82)	<.0001
Phosphorus	49,324	40,436	–8,888 (–18.0%)	0.83 (0.81-0.84)	<.0001
Total	363,692	321,927	–41,765 (–11.5%)	0.89 (0.89-0.90)	<.0001

CI, confidence interval; FY, fiscal year.
* FY 2003-2004 test volume relative to that of FY 2002-2003.
† Incidence rates (eg, No. of CBC counts performed in FY 2003-2004/No. of CBC counts performed in FY 2002-2003) were calculated and compared using a ratio measure as described in the "Materials and Methods" section.

Figure 1.8 Difference of decreasing of five test (May, 2006)

As a result of this change made in the test requests, it was observed that there was a decrease in the unnecessary laboratory tests. Completely unnecessary laboratory testing demands have not been eliminated.

In another article investigating the reasons for laboratory testing and the relationship between tests, as a result of investigations carried out on 1621 patients and 15603 tests requested for these patients; It is understood that non-normal test results are common and unnecessary laboratory test demands produce false positive results, and these results are likely to cause the diagnosis chaos (Houben, 2010).

Distribution of abnormal test results at test level.[1]

	N tests	% abnormal test results according to reference values	% markedly abnormal tests according to action limits
Patients (n= 1484)	7400	20.7%	4.9%
Sex			
Men	2857	21.4%	4.3%
Women	4543	20.3%	5.3%
Age			
< 40 years	1747	13.7%	2.5%
40–60 years	2773	19.8%	4.0%
> 60 years	2880	25.8%	7.4%
Reason for ordering tests			
Confirm/determine treatment	725	27.2%	9.1%
Exclude/physician's uncertainty	3654	18.6%	4.2%
Reassure/patient's request	1139	13.4%	2.0%
Monitoring (screening/check-up)	1761	27.0%	6.6%
Other reasons	33	15.2%	3.0%
Estimated pretest probability[2]			
Definitely no disease	541	13.9%	2.2%
Probably no disease	1990	14.2%	2.6%
Possibly disease	1651	19.9%	4.5%
Probably disease	1127	24.3%	7.0%
Definitely disease	196	34.7%	14.3%
Presumed diagnosis[2]			
A97, P, Z	1083	13.3%	2.7%
Other ICPC codes	4350	20.1%	4.9%

Notes: [1]Limited to the 14 tests in Table I. [2]The physicians did not estimate a pretest probability or mention a presumed diagnosis if they ordered the tests for screening or check-up.

Figure 1.9 Abnormal test result (Houben, 2010)

At test level, the frequency of markedly abnormal test results was clearly smaller than the frequency of abnormal test results. The frequency of abnormal test results increased with increasing age (13.7%–25.8%) and increasing pretest probability (13.9%–34.7%), and was high for tests ordered to confirm disease (27.2%) and for screening (27.0%) (Houben, 2010).

This article showed us that there are no unnecessary requests in laboratory tests, but also unnecessary requests for radiology and imaging testing. This part is also considered in the content of the thesis, and it provides the execution of these demands by providing interaction with subsystems such as system radiology when there are imaging test requests in defined pre-diagnosis protocols. In this way, the unnecessary demands in this area will be prevented and the standard will be provided.

According to a study in Ireland, the ability to prevent inappropriate tests performed again after patient transfer between hospitals was investigated. Patients are not satisfied with these tests. In order to prevent these unnecessary tests, detailed results of laboratory tests are included in the patient transfer files. Even as a result of this intervention, the numbers of FBC, Bio profiles and Coag tests were 758, 749 and 268, respectively, while these figures decreased to 85, 84 and 0 respectively. As a result, the author; until May 2014, interventions led to a steady drop in inappropriate test rates. Estimated cost reductions exceed two million euros annually (Brady, 2018). The problem of unnecessary testing is a chronic problem and it creates serious costs.

Even in high-volume, low-cost tests, this unnecessary use leads to serious costs. In another study, even the decision support method for CBC, PT, aPTT tests was tried to reduce the usage rate (Jackups, 2017). It is not a complete solution in terms of its application to partial tests.

Another study explored the efficiency of opportunities and existing systems in this area. He stated that the most promising method is computerized laboratory management systems and diagnostic-treatment reimbursement system (Janssens, 2010). This thesis propose to use the diagnostic-treatment system that the author mentioned, not only in the reimbursement system, but throughout the entire treatment process.

In the US in 2015, $ 7 billion was spent on laboratory tests by Medicare alone, with 16 to 40% of these tests being clinically unnecessary. Educational interventions, clinical decision support interventions have been tried to prevent this unnecessary. In addition, the author sheds light on this thesis by mentioning that it would be attractive to have automatic order entry systematic in the future (Jane, 2018).

Reasearch was conducted in the emergency department. Some laboratory tests were taken from the electronic request form and the behaviors of the physicians were observed. As a result, it was observed that physicians' test ordering behaviors changed (Lapić, 2017), which shows us that putting all tests in front of physicians does not have good results for real analysis, causing confusion and causing unnecessary tests by physicians.

Another good research study; As a result of inappropriate testing, 70% of the test results, a computerized alarm system was tried to be prevented in the repeat tests. At the University of Parma Hospital, 15 laboratory tests were conducted for a 6-month period, with a pop-up alarm in case of non-compliance. In general, 765/3539 (22%) test records were followed by alarms. After the alarm view, 591 test requests were canceled (Lippi, 2015). This shows us that even with the alarm, unnecessary test requests can be avoided, but not completely solved. After a while, physicians will not pay attention to these alarms.

CHAPTER TWO

IMPLEMENTATION

2.1 Architectural View

The application architecture is the front-end functional properties of the system. When the user enters the application, the main screen is displayed and the user can select an option. Once an option is selected, the user can manage other operations in the application. In our applications, users will be able to add and manage the pre-diagnosis protocols. The other problem is the platform on which the program will be developed. This platform was selected as Visual Studio. JavaScript programming language. All this is shown in Figure 2.1 below.

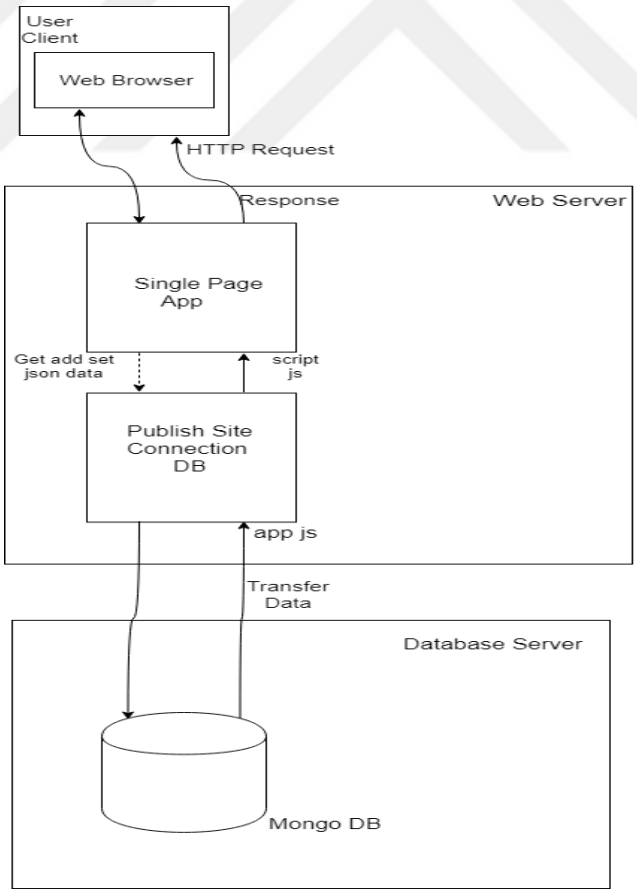


Figure 2.1 Architectural View Diagram of DDX

2.2 ER Diagram

Entity relationship diagram of the application that include necessary tables and tuples of the application with primary keys. In our application also need a database for user registration and storing users' friend lists. And we needed an Entity-Relationship diagram for designing a database. Necessary tables are defined for application's database. These are user table and contacts table. The description in this paragraph is shown below in Figure 2.2.

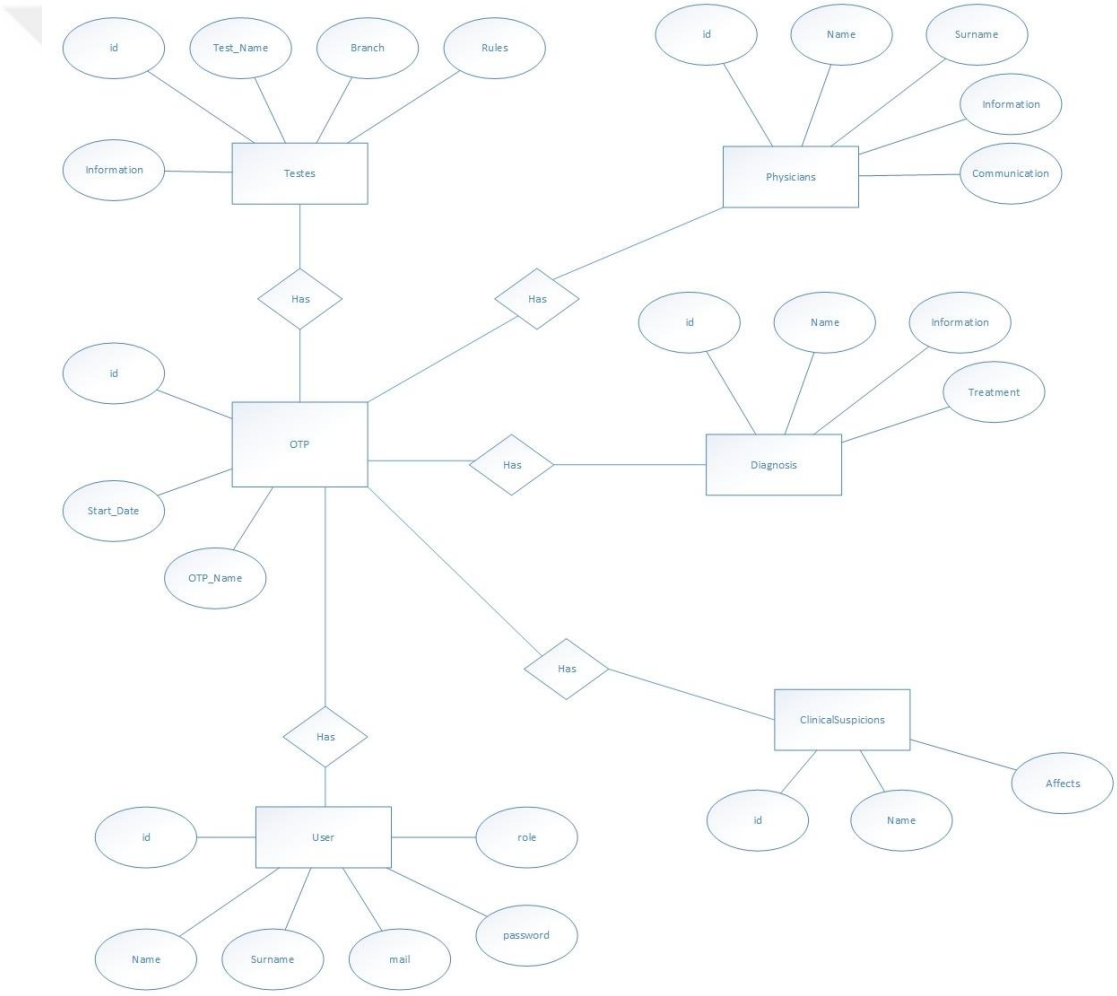


Figure 2.2 ER-Diagram of DDX

2.3 UML Class Diagram

Class diagram of the application with necessary attributes and data types. It is important to determine which classes application need before starting to develop the application. Otherwise we have to make major changes on the application code. And it can take lots of time. This is not a satisfied condition. All needs must be defined before starting to develop an application. In our application, three classes are defined for according to project's needs. These are "User Operation", "Screen Operation", "Create OTP", "List OTP" and "Save OTP". The description in this paragraph is shown below in Figure 2.3.

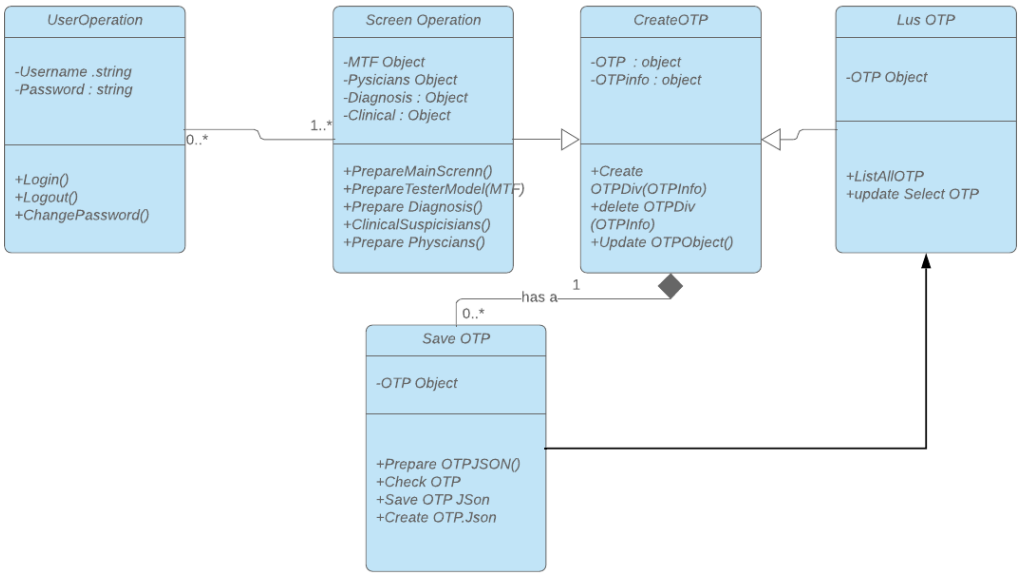


Figure 2.3 Class Diagram of DDX

2.4 Use Case

Use case diagram is used for defining user's options. User may select many options and manage them that necessary flows. In our applications users have lots of options. User must register before login. Another thing is editing contacts option. User must follow some specific steps. First step is choosing control panel and then manage

contact option must be choose, so users can edit their contacts. Users also can change their profile setting and must follow some steps. The description in this paragraph is shown below in Figure 4.4.

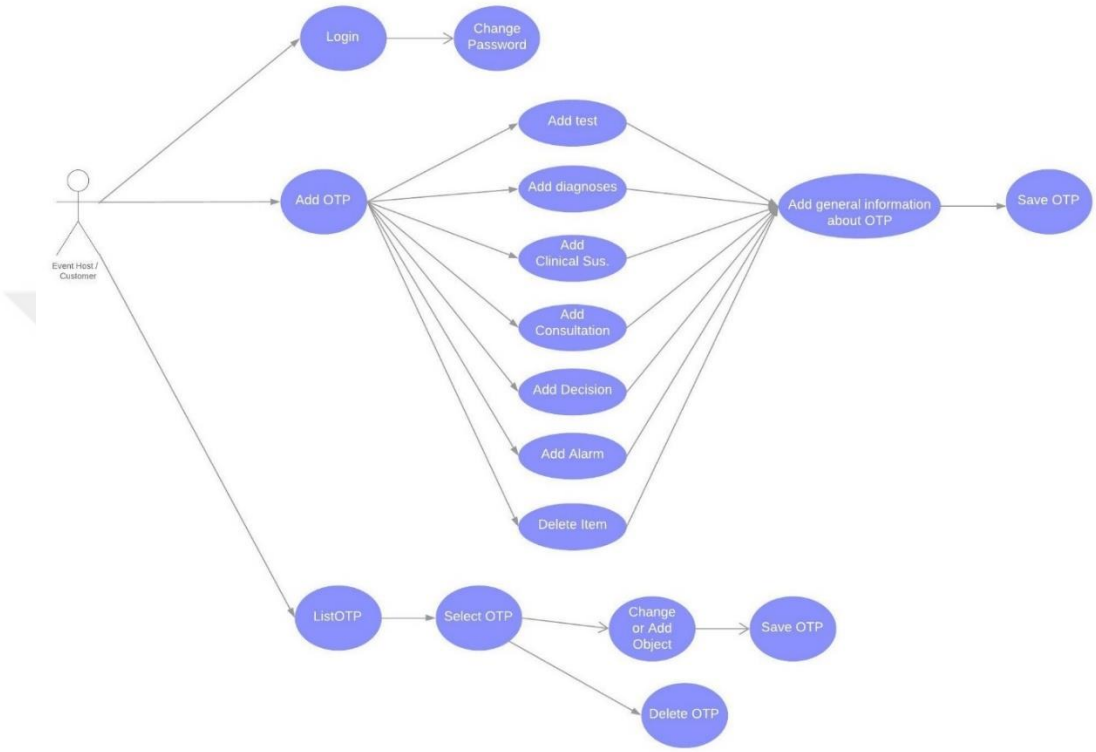


Figure 2.4 Use case diagram of DDX

2.5 Activity Diagram

In Figure 2.5-7 activity diagram of project is shown. All possibilities that user can select is clearly specified. User may select many options and manage them that necessary flows. For example, user must be register before login. User have lots of options like add OTP, delete OTP, and update OTP. Activity diagrams show clearly these options.

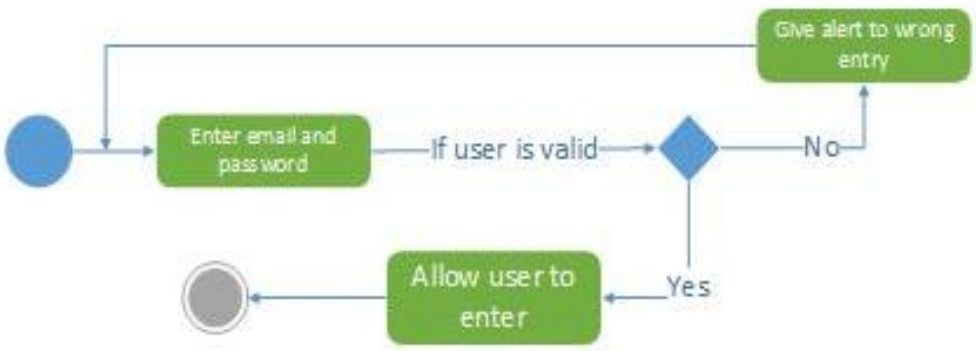


Figure 2.5 Activity diagram for login operation of application

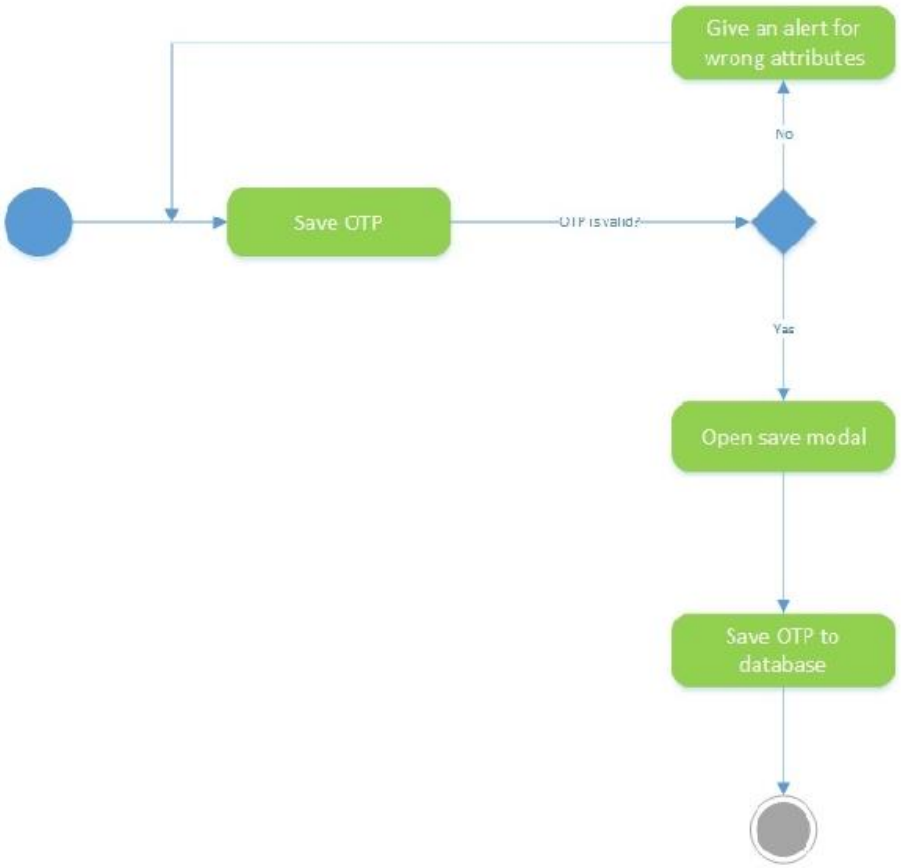


Figure 2.6 Activity diagram for save OTP operation of application

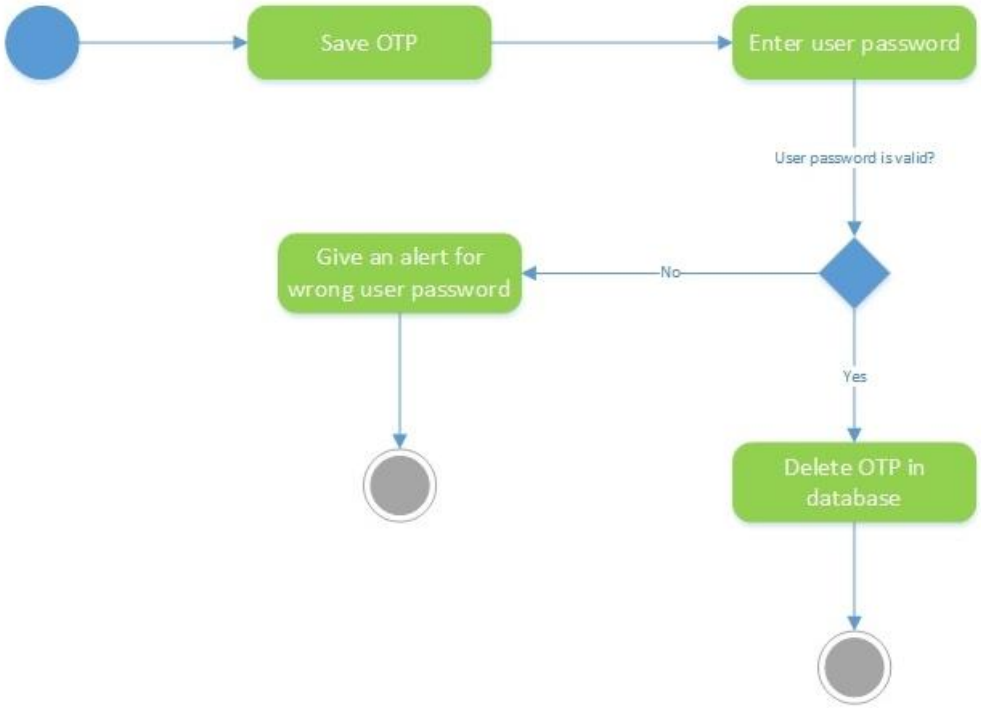


Figure 2.7 Activity diagram for delete otp operation of application

2.6 Sequence Diagram

Sequence diagrams show object interactions arranged in time sequence. According to users' choose functions runs a specific order. For our applications this functions and its sequences' shown in Figure 2.8 and 2.9.

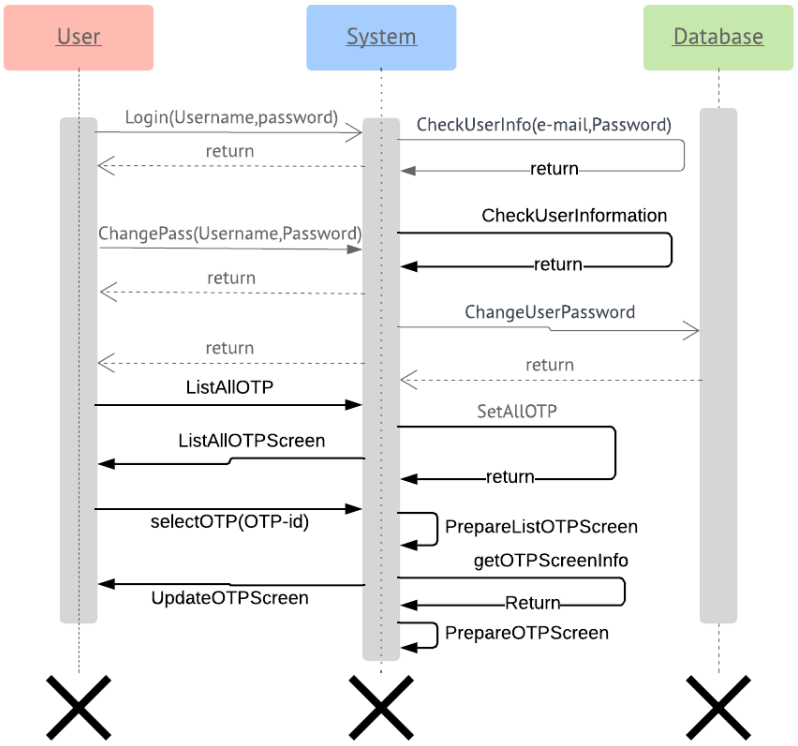


Figure 2.8 Sequence diagram of the application 1

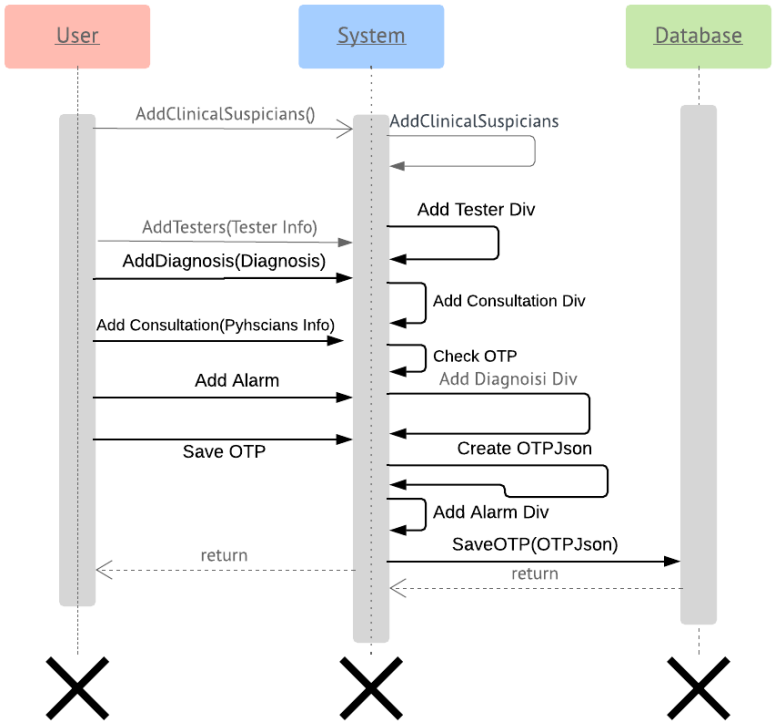


Figure 2.9 Sequence diagram of the application 2

2.7 Deployment Diagram

There is a database to store user information. The contact list will be kept in the database. It is shown in Figure 2.26.

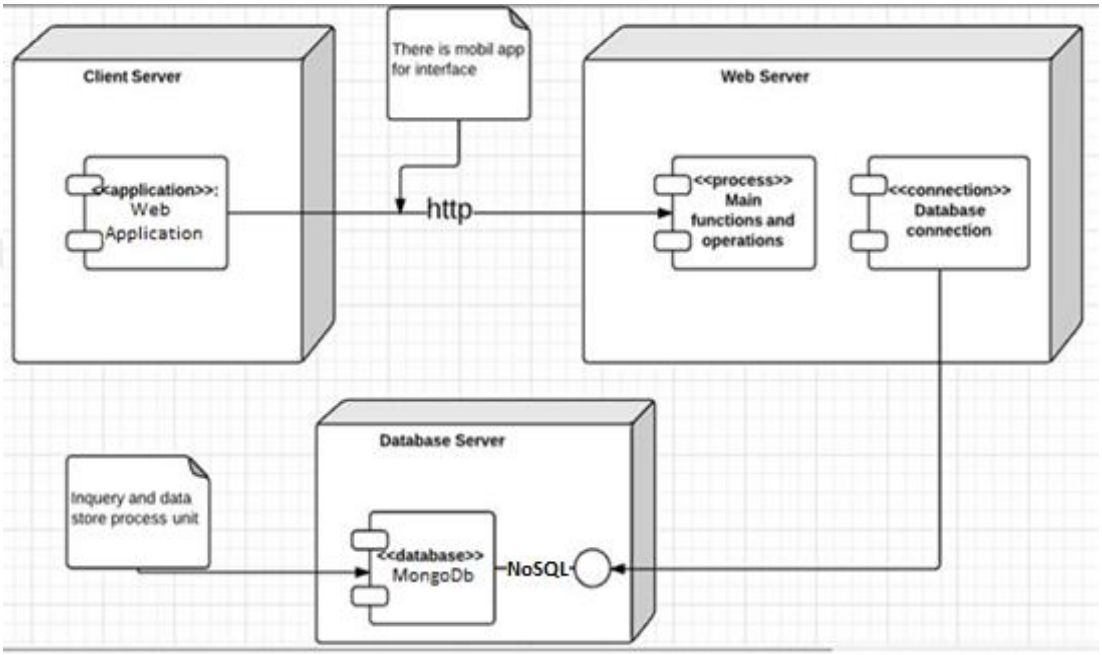


Figure 2.10 The system will be created by using three parts

2.8 User Interface Model With Graphical Language

The creation of the Pre-Diagnosis Protocols (PDPs) has been done through flow diagrams because of the understanding and ease of follow-up. While designing PDPs, Scalable Vector Graphics (SVG) has been used as a visual model to ensure easy interaction with the system and to avoid visual disturbances in the use of different types of devices. First of all, PDPs are divided into sub-components so that the SVG designs of each sub-component have been specially designed. In this way, users can easily create their own PDPs by selecting these sub-components separately from the screen and dropping them into the drawing area. System-specific customizations have been made by using the mxGraph library in the drawing screens and subcomponents

designs. The visual flow diagrams of the created protocols were converted to eXtensible Markup Language (XML) data structure and sent to server side by using mxGraph library. PDP has also been converted from XML to JavaScript Object Notation (JSON) data structure and stored in both data in server side. Thus, described PDPs have been made possible to be easily placed on the screen together with their former positions when it is requested to be accessed via the system and to make changes again through these diagrams. As the system database, security, speed, licensing fee, conformity to the outputs of the system, the convenience of storing the PDPs and the real situations modelling were taken into consideration and the non-relational database, MongoDB has been used

User start with home screen and it include the wall. It is shown in first part of Figure 2.11 and the last part of Figure 2.24.

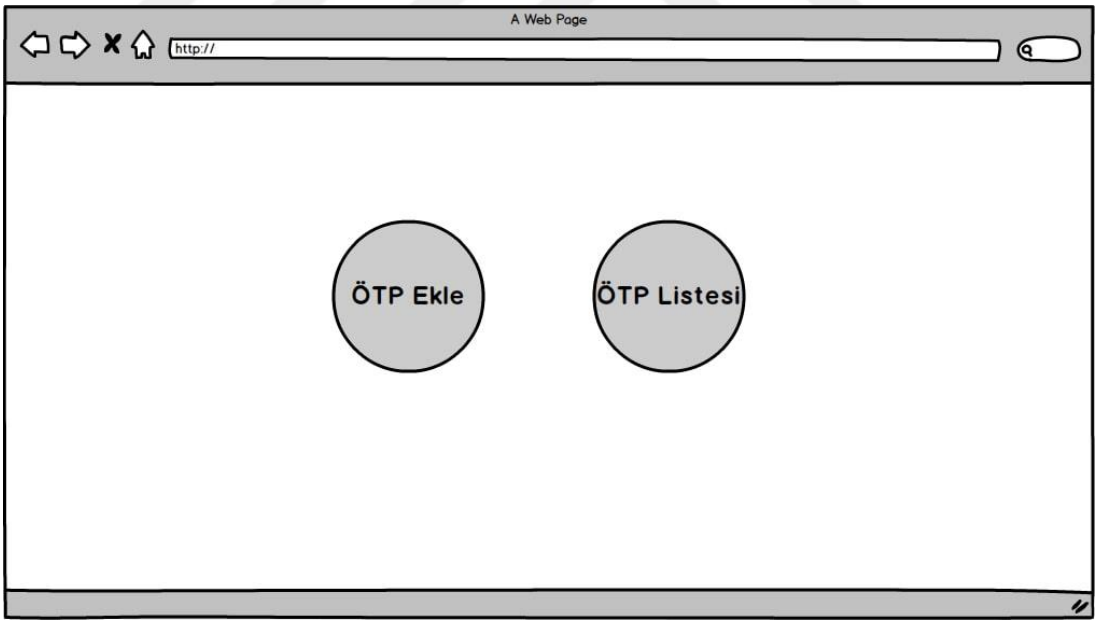


Figure 2.11 DDX main screen, user select operation

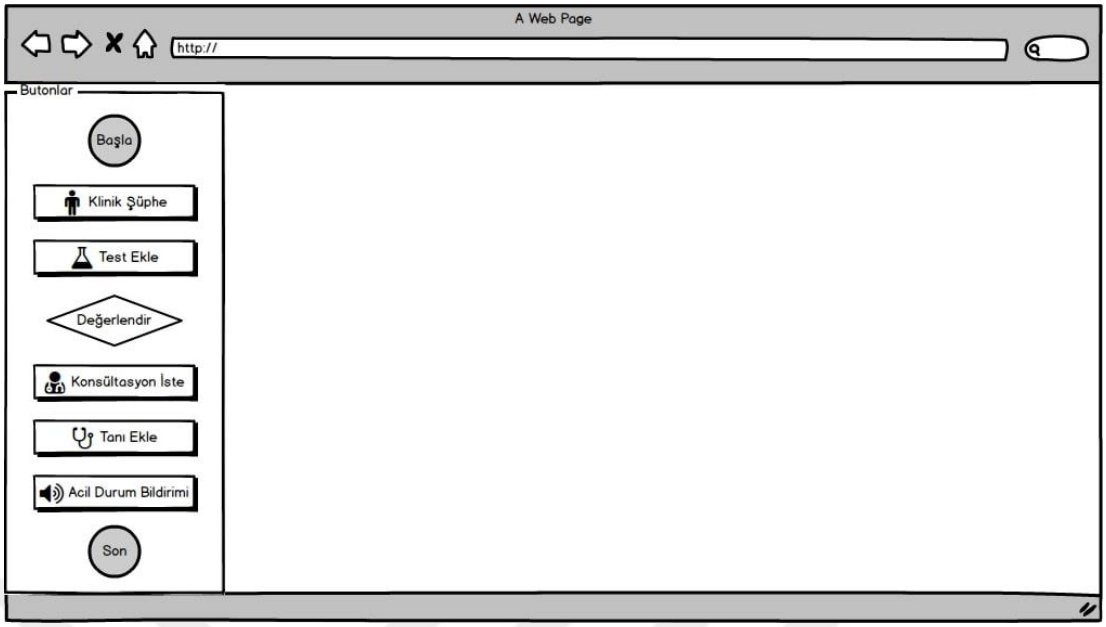


Figure 2.12 DDX Add OTP main screen, all components in left area

Drag-and-drop management from the components menu on the left side creates a pre-diagnosis protocol by dropping components into the main drawing area. Each component has its own unique fields. required information is requested from the user at the time of drag and drop. The user cannot cross the screen without completing these steps. It is mandatory to start with the element Start. Validation checks are activated every time a component is added to prevent incorrect drawing. The diagram can be terminated with more than one Finish element depending on the situation. No components can remain in the open which does not end with the finish element. Auto save time is 2 minutes. It is designed to allow users to use frequently used shortcuts in the drawing area.

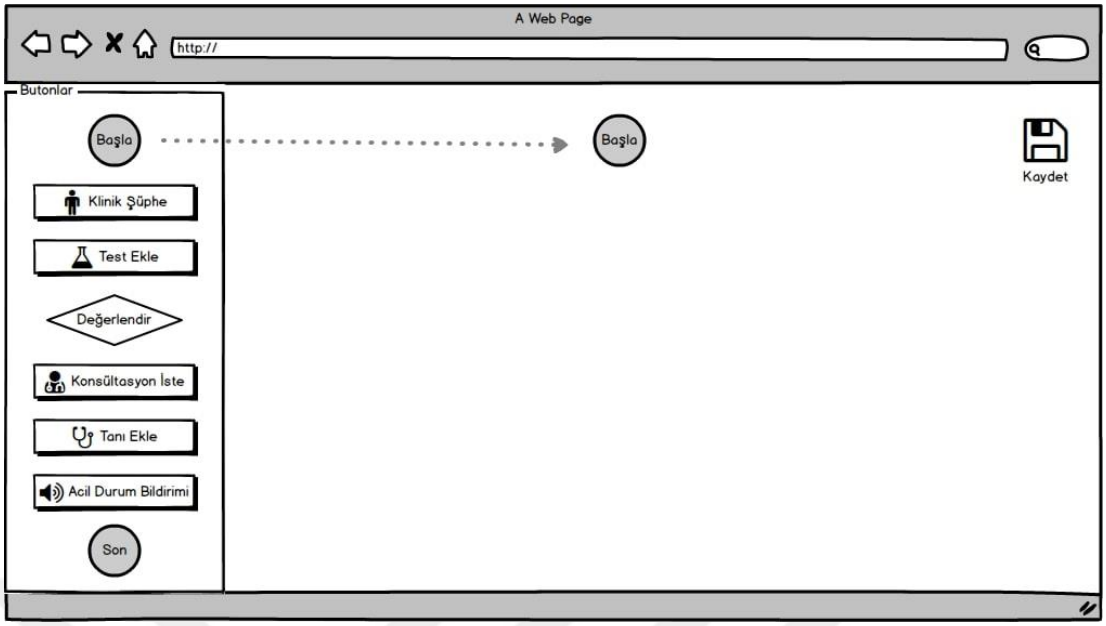


Figure 2.13 DDX adding start operation

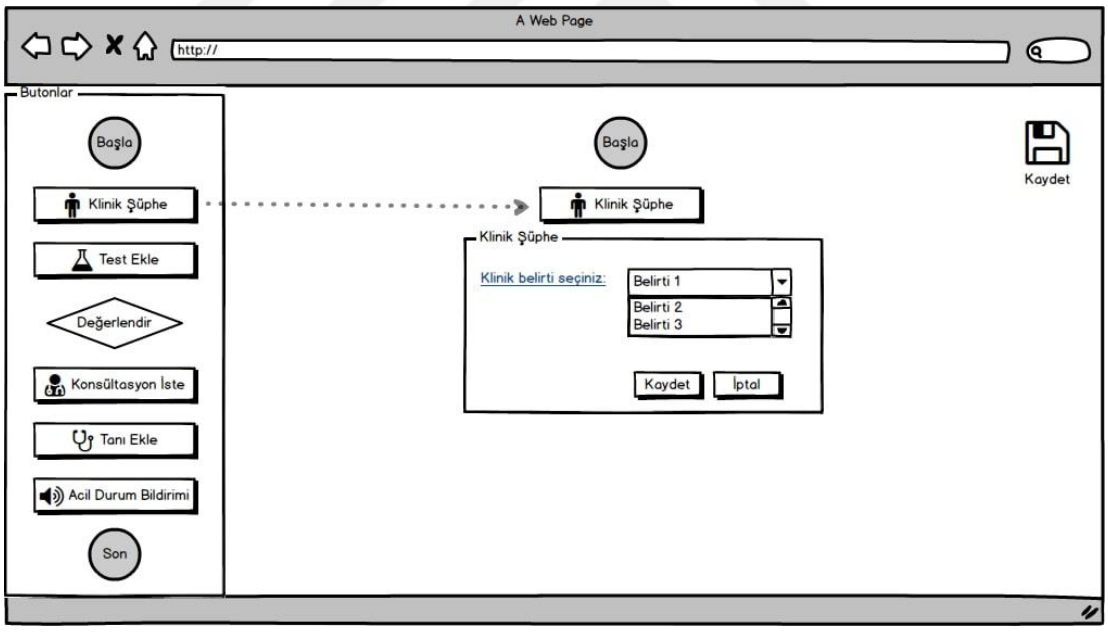


Figure 2.14 DDX add clinical suspicion operation, user select suspicions and save

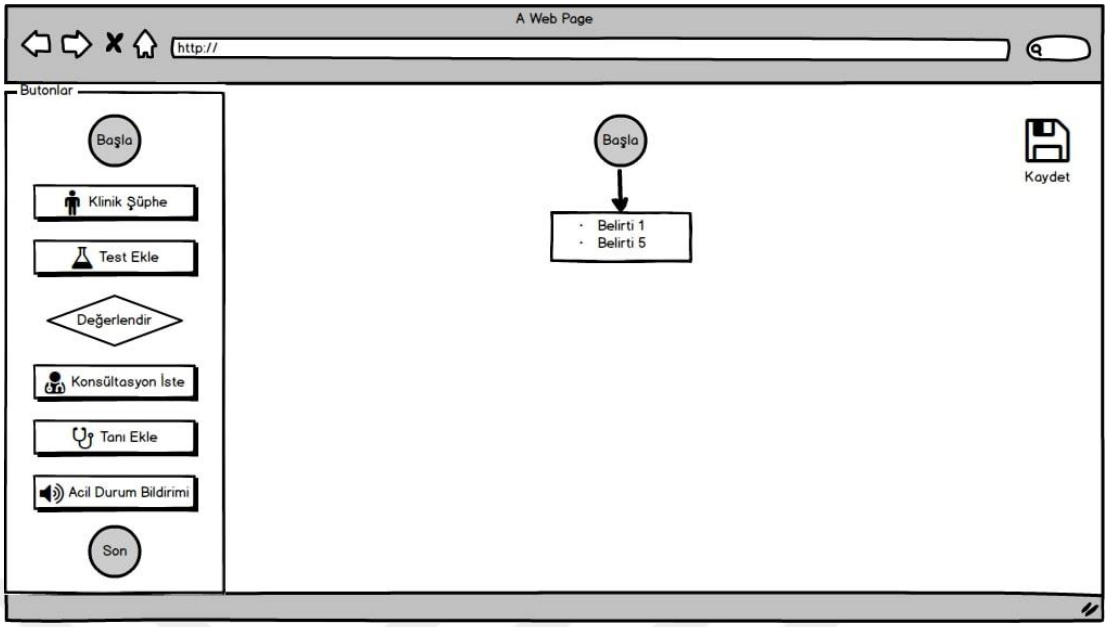


Figure 2.15 DDX after adding clinical suspicions

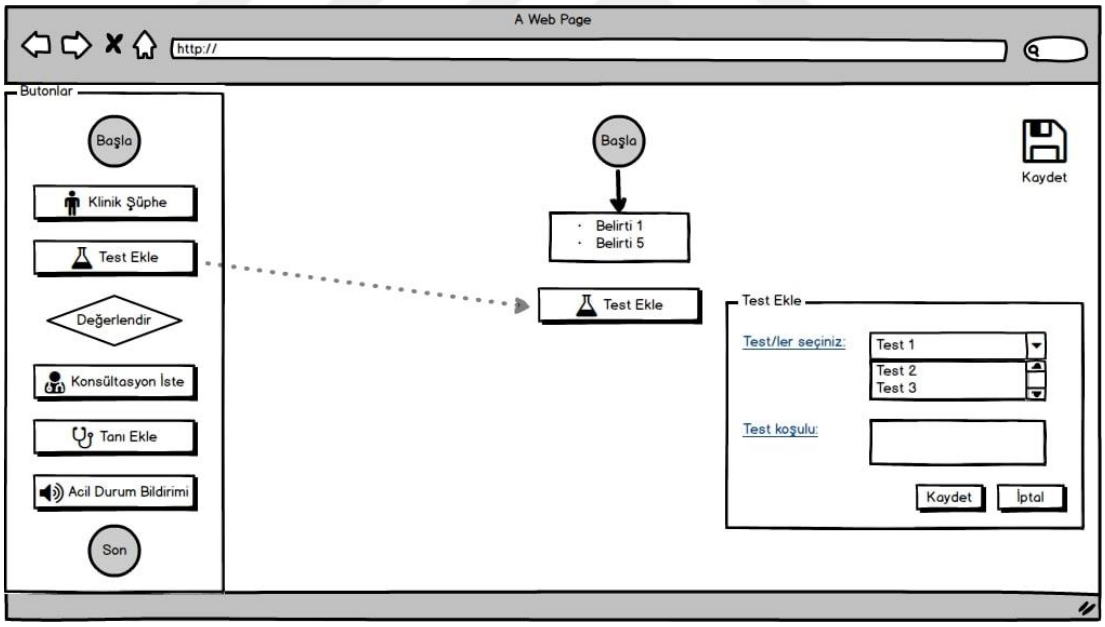


Figure 2.16 DDX adding clinical test, user select test requests

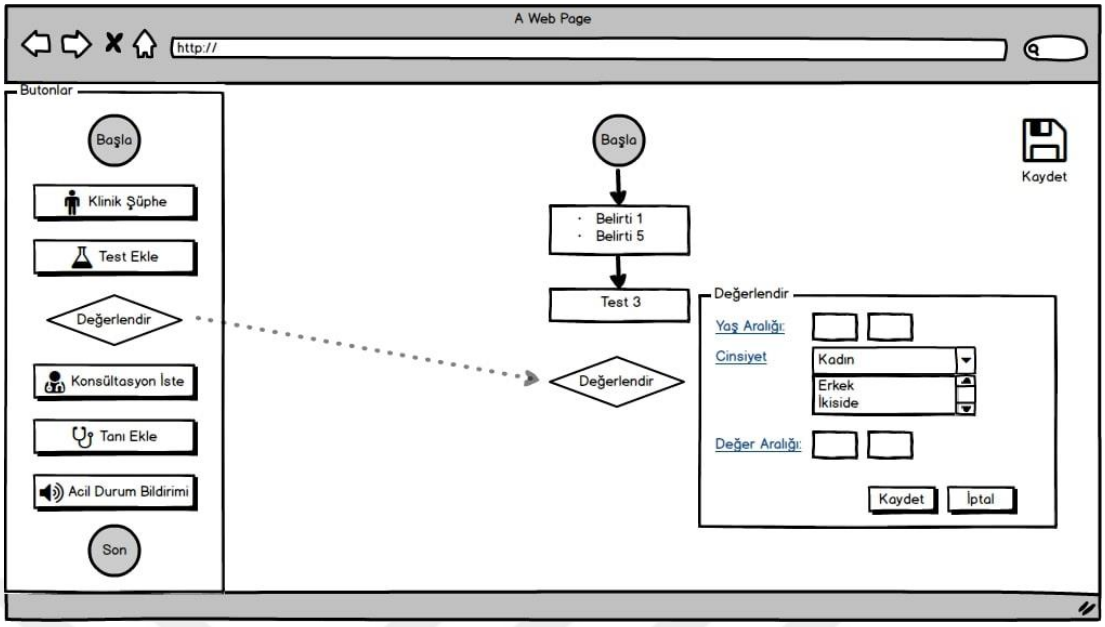


Figure 2.17 User add evaluating test results.

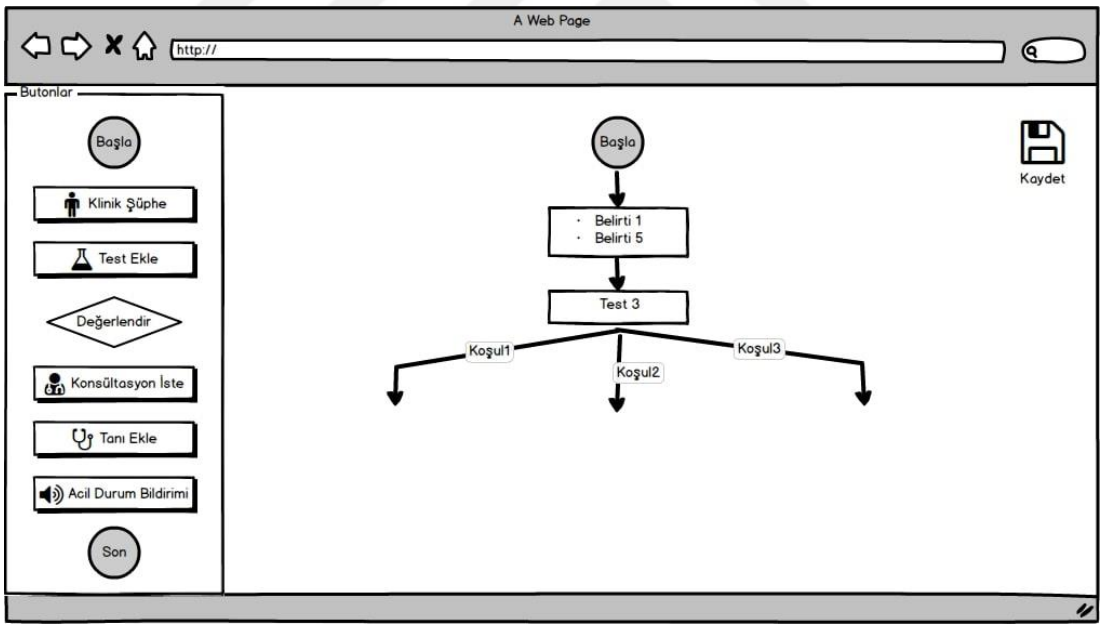


Figure 2.18 After defining test rules

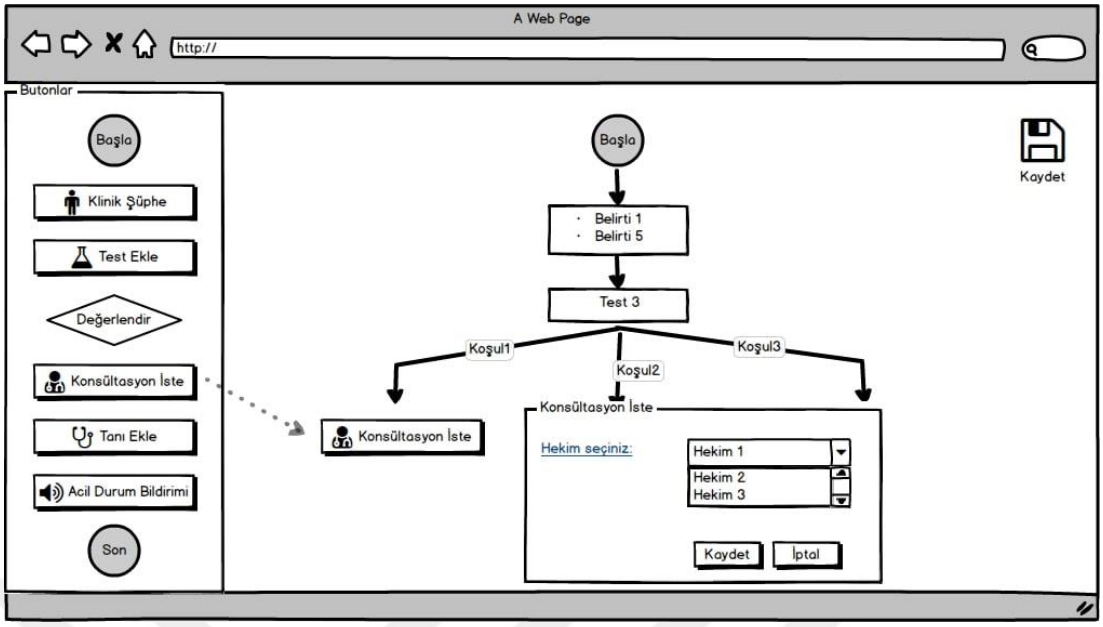


Figure 2.19 User adding consultation request and select physician

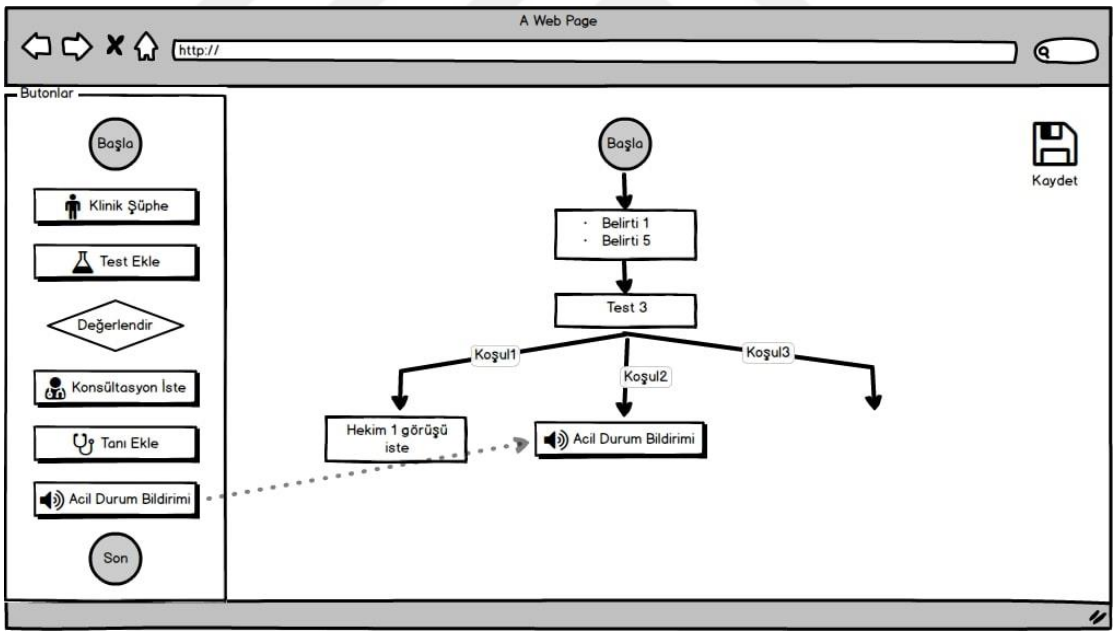


Figure 2.20 User adding emergency notification

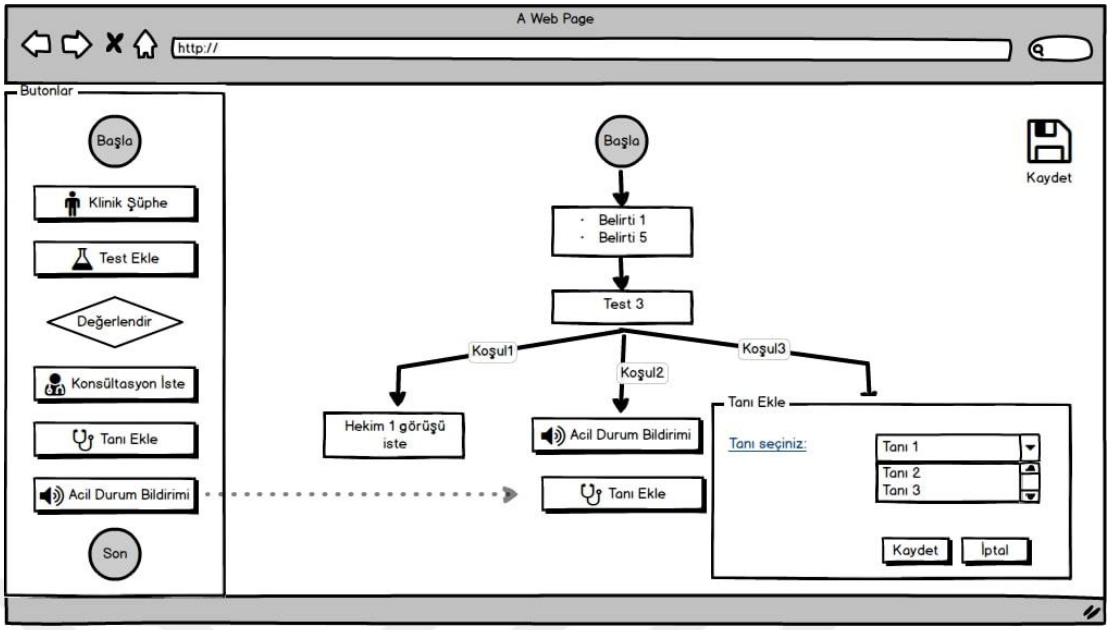


Figure 2.21 User add clinical diagnosis

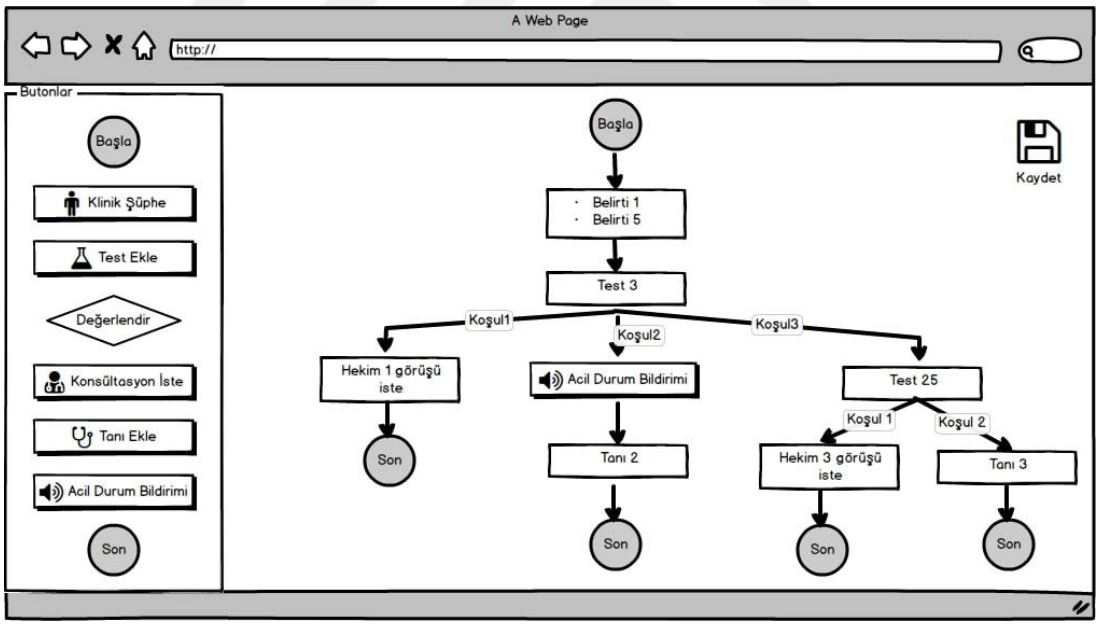


Figure 2.22 An example of an OTP where all operations are defined

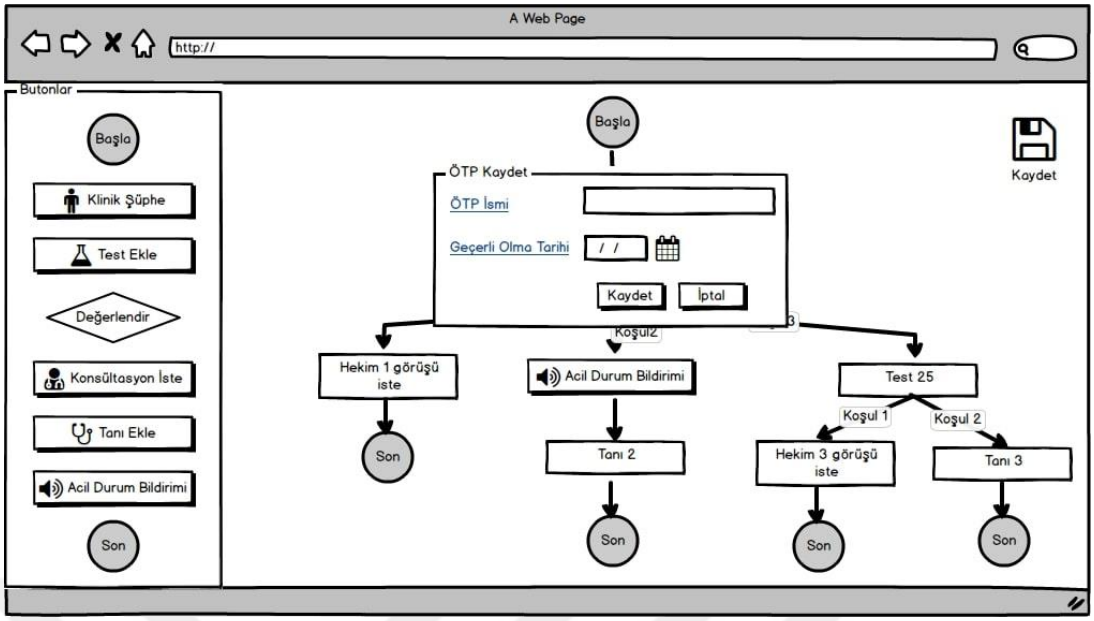


Figure 2.23 Saving OTP, user define OTP name

Figure 2.24 All OTP defined in the system are listed on this screen

2.9 Back-end Operation

The project is implemented two side. Server side is second side of the project that is going to be implemented on Node.js platform. Node.js is the famous platform for web technologies that gives high performance, simply code environments and many libraries to communication. Another advantages of the Node.js is high performance and response time is very short.

```
//#region Requirements
const tls = require('tls');
const express = require('express');
var compression = require('compression')
var minify = require('express-minify');
var session = require('express-session');
var fs = require('fs');
var http = require('http');
var webSocketServer = require('ws').Server;
const bodyParser = require('body-parser');
const MongoClient = require('mongodb').MongoClient;
const ObjectId = require('mongodb').ObjectID;

https = require('https');
var helmet = require('helmet');
global.config = require('./config.js');
global.lsputils = require('./lsputils.js');

const {DOMParser} = require('xmldom');
const xmlToJSON = require('./xmlToJSON.js');
xmlToJSON.stringToXML = (string) => new DOMParser().parseFromString(string, 'text/xml');

var dbServer = null;
var adminServer = null;
var doctorServer = null;

var adminWsWebServer;
var doctorWsWebServer;

var adminWsServer;
var doctorWsServer;

global.adminWsClients = [];
global.doctorWsClients = [];

global.authWsAndAdminClient = Object.create(null)
global.authWsAndDoctorClient = Object.create(null)

global.MasterSocket;
var MasterSocketAuth = false;
var MasterReconHandler = null;

process.on('uncaughtException', function (err) {
    console.error(err.stack);
    console.log("Node NOT Exiting...");
    fnSaveLog(err.message + "  " + err.stack);
});

process.on('unhandledRejection', (reason, p) => {
    console.log('Unhandled Rejection at: ' + p + ' reason: ' + reason);
```

Figure 2.25 Server.js example codes

The Figure 2.25 represents communication between clients on server using Node.js socket technologies. Client send request to server for sending message to other client.

Server checks specific client based on unique id and then match two client as a result client send message to other client.

All attributes defined in the project are defined by using Hungarian notation. In this way, all related persons are able to understand the attributes (Figure 2.25). In the developed project, all the main operation which access to data set are in the same format. This format provides integrity between the functions. If the operation is successful, the function returns the success message or calls another function. However, if the operation is not successful, the function returns an error message or performs the defined operation (Figure 2.26).

```
43  let obUsers = {};
44  let obPhysicians = [];
45
46  let obOTPSavedList = {};
47
48  let obCurrentOTP = {};
49
50  let arTests = [];
51  let obTests = {};
52
53  let arIcd10 = [];
54  let obIcd10 = {};
55
56  let arLabs = [];
57  let obLabs = {};
58  //#endregion
59
60  //#region Loader Functions
61  function fnLoaderStart() {
62      //input: nothing
63      //output: nothing
64      //modify: nothing
65      //Opens loader screen
66
67      try {
68          $("#page-loader").removeClass('hidden');
69      } catch (e) {
70          console.log(e);
71      }
72  }
73
74  function fnLoaderStop() {
75      //input: nothing
76      //output: nothing
77      //modify: nothing
78      //Closes loader screen
79
80      try {
81          $("#page-loader").addClass('hidden');
82      } catch (e) {
83          console.log(e);
84      }
85  }
```

Figure 2.26 Hungarian notation example

The relational database that web and mobile applications will share has prepared with MongoDB program. Each Pre-Diagnosis Protocols is different from each other, they do not have a classical data structure. Each of them can be of different length and format. So MongoDB was chosen as the database.

Database connections are prepared according to the individual diameters specified in heading 2.2. Since the content of the database is not evaluated during the test, it contains random information. However, the development process may continue according to the changes in the program when the program development process is completed and when the actual user is turned on.

2.10 PDP to FSM Conversion

Protocols must be converted into an executable form so that computers can understand and execute them. The protocols drawn on the drawing screen are at a level that people can understand. The implementation of the protocols on the system has been made possible by FSM. In the system, it is possible to run more than one PDP per patient in parallel. Each protocol is converted into FSM that specific to each patient. Before converting to finite state machines, the PDP's XML data is converted to JSON data structure. For this conversion, "xmlToJSON, version: 1.3.4" package has been used on Node.js. The valid XML data from the client is first converted to the JSON data structure by using the conversion library after the server-side has been verified whether the XML is valid or not. Shape, location, color information and other unnecessary fields are removed from the resulting JSON data. Finally, the necessary transition actions are added or edited in the appropriate format to convert FSMs. It is shown in figure 27.

Figure 2.27 PDP to FSM conversion stages

2.11 FSM Execution

The resulting FSMs by the PDP to FSM conversion are recorded to the database with the corresponding PDP. From the physician terminal screens, the appropriate protocol for diagnosis for the patient is selected and the start command is pressed. With the FSM of the PDP that requested, patient information is initially recorded in the database and the FSM starts to run through the run-time system. All the steps in the protocol are automatically executed by the system. For example, for laboratory tests order, system executes the relevant socket command to the laboratory subsystem and if there is progress on the result, it waits for the result or if it needs to send a message, it sends the corresponding socket message to the messaging node and puts the state to the next state. In each case the progress of state or result is updated in the database is provided. The system is designed to communicate with multiple subsystems. These systems can be laboratory information management systems, external company systems, radiology systems and other image archiving and communication systems. In this way, the protocols to be drawn can include all medical examination at the whole diagnostic stage of the hospital.

2.12 Front-End Operation

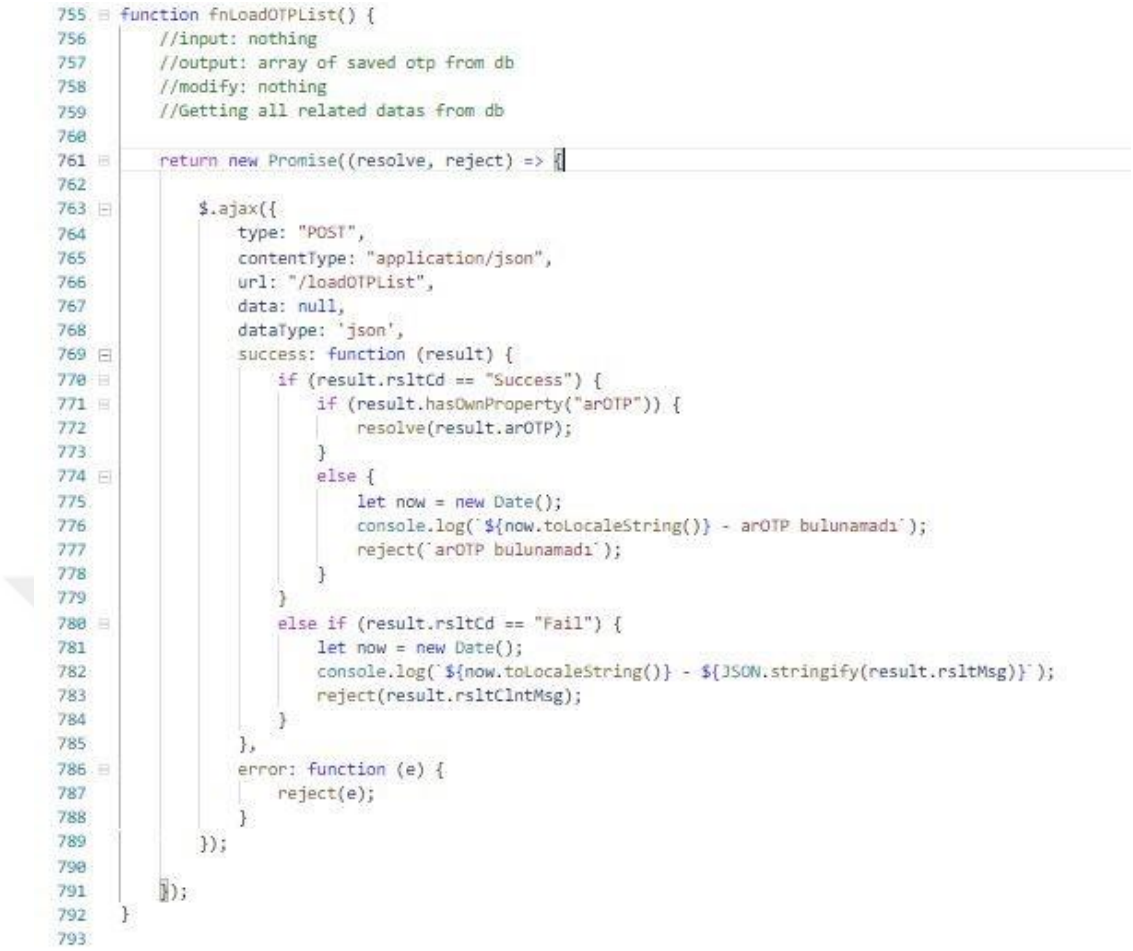
The interface of the web application has been designed to suit the members. Users will not be able to enter the system without login in the site's login page. Users will be

directed to the site's homepage when they log into the system. In the main menu, users will be able to use all features without any restrictions.

Users operations are displayed on the screen which is opened after login. Here, user can view registered Pre-Diagnosis Protocol records, edit registered Pre-Diagnosis Protocol or create new Pre-Diagnosis Protocol record. An empty drawing page will open and users add and save the desired options, when the user wants to add a new Pre-Diagnosis Protocol. The Pre-Diagnosis Protocol list contains information about the person who has prepared the Pre-Diagnosis Protocol, the date of preparation, Pre-Diagnosis Protocol short name and long name and editing option. If the user clicks the edit button, the system will open the drawing screen of the corresponding the Pre-Diagnosis Protocol and allow the user to make changes.

HTML5 and CSS languages were used predominantly for interface design. Different CSS files and features were created for each section on the site. We used the jQuery language to quickly and easily manage the pages. The site design was made to fit the all screens in all sizes. The responsive structure was fully provided. The dashboard structure was developed with different designs to match the device used. ease of use was provided for everyone at this address.

In front-end side, display of data or new data entry was done with ajax post. It was only necessary to update the relevant changes without having to reload the page completely when the data entry was made. For this, we only send and receive the entry area as a separate html section in ajax post. In this way, the speed and performance of the site have been increased. The example code is shown in Figure 2.27.



```
function fnLoadOTPList() {
    //input: nothing
    //output: array of saved otp from db
    //modify: nothing
    //Getting all related datas from db

    return new Promise((resolve, reject) => {

        $.ajax({
            type: "POST",
            contentType: "application/json",
            url: "/loadOTPList",
            data: null,
            dataType: 'json',
            success: function (result) {
                if (result.rsltCd == "Success") {
                    if (result.hasOwnProperty("arOTP")) {
                        resolve(result.arOTP);
                    }
                    else {
                        let now = new Date();
                        console.log(`${now.toLocaleString()} - arOTP bulunamadı`);
                        reject(`arOTP bulunamadı`);
                    }
                }
                else if (result.rsltCd == "Fail") {
                    let now = new Date();
                    console.log(`${now.toLocaleString()} - ${JSON.stringify(result.rsltMsg)}`);
                    reject(result.rsltClntMsg);
                }
            },
            error: function (e) {
                reject(e);
            }
        });

    });
}

```

Figure 2.28 AJAX post code

2.13 Libraries

Socket.IO provides an event-oriented API that works across all networks, devices and browsers. It is incredibly robust (works even behind corporate proxies!) and highly performance, which is very suitable for multiplayer games or real time communication.

MxGraph is an interactive JavaScript html5 library that is being developed since 2005. MxGraph allows you to create fast interactive graphics and graphical applications. It is a client-side library that uses SVG and HTML to create models.

Therefore, it does not create a density on the server. It is one of the best libraries used in this area because it does not want to download any plugins from the user.

The client-side model is kept as an XML structure. This XML structure contains the type, name, property, and additional information of each graphical object. The library needs a web server to deliver the generated XML to the server. In this project, this transmission was performed with web socket.

CHAPTER THREE

CONCLUSION

At the end of the thesis, an application was introduced to reduce the workload of specialist physicians and enable them to record the patient control systematics algorithmically. In practice, physicians will be able to draw and record the laboratory algorithmic behaviours applied on the screen during the disease diagnosis, to examine the algorithms previously drawn, to make the algorithm different or to remove the algorithm from the system. These drawn streams will be converted to a finite state machine by the application. The application will make an external integration connection to another application and physicians will be able to request these pre-diagnosis protocols for patients. The flow in this request will be automatically monitored via finite state machine. But of course new features can be added later on in the, for example fully otomatic OTP designer with artificial intelligance. The progeny development process can be continued with new features in the future.

At the beginning of the project development phase, the project was developed at two different stages. These stages are the preparation of the client side that will provide the algorithm drawing and the preparation of the server side which will convert the drawn algorithms into finite state machine. These phases were combined at the end of the project. The application is prepared to be responsive for ease of use and is compatible with all web browsers.

There is no need for any additional security as other users are not connected to each other in the application. The measures in the application have been used for the correct preparation of algorithms. Restrictions have been imposed on the majority of users. New constraints can be added if users need new restrictions or provide feedback. Any mistakes, exploits, problems, and changes that could increase the user experience can be remedied by returning users after the completion of all phases of the project.

The system that fully simplifies, standardizes and coordinates the selection, planning and staging of pre-diagnosis examination requests to facilitate this challenging process of physicians has been proposed. With the use of this system; the individual variations between the standardized pre-diagnosis protocols and the demands will be eliminated, a pre-diagnosis protocol database will be established, which will last for generations to be permanent, unnecessary or redundant test demands will be eliminated, the costs of the excess test order will be eliminated, the demands of missing test will be completely eliminated, in the right time, the patient and physician information's will be done in a timely manner to increase the safety of the result and finally the risk of misdiagnosis will be reduced by increasing the accuracy of the results.

REFERENCES

Algorithms, (n.d.). Retrieved September 12, 2018, from https://www.mayomedicallaboratories.com/articles/resources/algorithms.html

Arup Algorithms, (n.d.). Retrieved September 17, 2018, from https://arupconsult.com/algorithms

Atasever, M. (2014). *Türkiye sağlık hizmetlerinin finansmanı ve sağlık harcamalarının analizi 2002-2013 dönemi*. Ankara: Strateji Geliştirme Bakanlığı.

Brady, H., Piggott, L., Dunne, S. S., O'Connell, N. H. & Dunne, C. P. (2018). Clustered interventions to reduce inappropriate duplicate laboratory tests in an Irish tertiary hospital. *Clinical Biochemistry, 52*, 26-32.

Houben, P. H. H., et al. (2010). Reasons for ordering laboratory tests and relationship with frequency of abnormal results. *Scandinavian Journal of Primary Health Care, 28*, 18-23.

Jackups R. J., Szymanski J. J. & Persaud S. P. (2017). Clinical decision support for hematology laboratory test utilization. *International Journal of Laboratory Hematology, 39*(1), 128-135.

Janssens, P. M. W. (2010). Managing the demand for laboratory testing: Options and opportunities. *Clinica Chimica Acta, 411*(22), 1596-1602.

Khalifa, M. & Khalid, P. (2014). Reducing unnecessary laboratory testing using health informatics applications: A case study on a tertiary care hospital. *Procedia Computer Science, 37*, 253-260.

Lane, A. B., Robinson, S. L., Maddy, N. C. E., Shimeall, W. T. & Sadowski, B. W. (2018). Optimizing order entry automaticity reduces inpatient laboratory utilization. *The American Journal of Medicine, 131*(8), 908-912.

Lapić, I., Juroš, G. F., Rako, I. & Rogić, D. (2017). Changing the electronic request form proves to be an effective tool for optimizing laboratory test utilization in the emergency department. *International Journal of Medical Informatics, 102*, 29-34.

Lippi, G., et al. (2015). Effectiveness of a computerized alert system based on re-testing intervals for limiting the inappropriateness of laboratory test requests. *Clinical Biochemistry, 48*(16), 1174-1176.

Loutzenhiser, H. & Masarie, F. (2007). *System and method for macro-enhanced clinical workflow.* 14 June 2007, Patent Application Publication.

May, T. A., et al. (2006). Reducing unnecessary inpatient laboratory testing in a teaching hospital. *American Journal of Clinical Pathology*, *126*(2), 200-206.

Miyakis, S., Karamanof, G., Liontos, M. & Mountokalakis, T. D. (2006). Factors contributing to inappropriate ordering of tests in an academic medical department and the effect of an educational feedback strategy. *Postgraduate Medical Journal, 82*, 823-829.

Nachamkin, I. (2015). *How many lab tests do patients really need?*. Retrieved Agust 12, 2018, from http://pathology.med.upenn.edu/department/blogs/pepper-talk/how-many-lab-tests-do-patients-really-need

Oliveira, A. M., Oliveira, M. V. & Souz, C. L. (2014). Prevalence of unnecessary laboratory tests and related avoidable costs in intensive care unit. *Journal Brasileiro de Patologia e Medicina Laboratorial*, *50*(6).

Rosales, R., Mueller, M., Krishman, S. & Rao, B. (2006). *Guiding differential diagnosis through information maximization.* 7 September 2006, Patent Application Publication.

Weidenhaupt, K. & Lagor, C. (2008). *Clinical workflow management and decision system and method*. 25 September 2008, Patent Application Publication.