

School of Electronic
Engineering and Computer
Science

Final Report

Programme of study:
Computer science with
Management BSc

Project Title:
**Web application
supporting online
medical consultancy**

Supervisor:
Steve Uhlig

Student Name:
Khuseyn Kuliev

Final Year
Undergraduate Project 2020/21



Date: 04.05.2021

Acknowledgement

I would like to thank my supervisor Professor Steve Uhlig for his guidance and encouragement through out the project. Also would like to thank doctors and patients who have participated in interview sessions and user acceptance testing. Thanks to respondents who participated in questionnaires. I am also grateful to my family members who were supporting me throughout the project and providing feedback that has significantly improved project deliverables.

Abstract

There is a substantial problem of inappropriate treatment methodology provided by doctors in response to patient's medical problems.

A reasonable solution to this problem was considered in this project. Software enabling patients to reach multiple doctors distantly was implemented. Prior to the development existing systems analysis were conducted alongside with potential user research that includes interviews with potential patients and doctors. Web application designed allows doctors to provide online medical consultancy as well as publish their own articles in order to share their experience with others, especially with less experienced doctors.

Adequate testing and evaluation is conducted in order to ensure that the web application satisfies its requirements properly and addresses the stated problem.

During the project specific implementation related difficulties were faced, especially due to lack of software development experience but these were successfully overcome. However, personal health related issues have delayed delivery of project deliverables.

Contents

1. Introduction	8
1.1. Background	8
1.2. Problem Statement	8
1.3. Aim	9
1.4. Objectives	9
1.5. Report Structure	9
2. Related work	11
2.1. Interviews	11
2.1.1. Why do we need interviews?	11
2.1.2. Who are the interviewees?	11
2.1.3. Interview analysis	11
2.2. Existing systems analysis	13
2.2.1. Existing systems list	14
2.2.2. Comparison	14
2.2.3. Conclusion	17
3. Design and Requirements	18
3.1. Users and User interactions	18
3.2. Functional Requirements	20
3.3. Non-Functional Requirements	22
3.4. Design Pattern	23
3.5. System design principles	23
3.6. Development approach	24
3.7. Development methodology	24
4. Implementation	26
4.1. Languages used	26
4.1.1. Web-framework and Language for Backend	26
4.1.2. Frontend Logic Technologies	26
4.1.3. Frontend Structure and Design	26

4.2.Database setting	27
4.3.Core functionality	27
5.Testing	39
5.1.Core features unit testing.....	39
5.2.Browser Compatibility Test.....	39
5.3.User Acceptance Test	39
6.Evaluation	41
6.1.User evaluation	41
6.2.Weaknesses and strengths.....	41
6.2.1.Strengths.....	41
6.2.2.Weaknesses.....	43
6.3.Legal aspects	44
7.Conclusion	45
7.1.Project achievements.....	45
7.2.Difficulties and challenges.....	45
7.3.Further development.....	46
7.4.Overall conclusion.....	47
References	48
Appendix A - Risk assessment	49
Appendix B: Interview structure and questions	51
Appendix C - Existing systems analysis structure	53
Appendix D: Existing systems home page screenshots	54
Appendix E: UML Class diagram.....	56
Appendix F - Summary of unit testing	57
Appendix G - User evaluation questionnaire	61
Appendix H - Questionnaire results graphs.....	62

List of Tables

Table 1: Functional Requirements List	22
Table 2: Non-functional Requirements List.....	23

Table 3: Summary of unit test	60
-------------------------------------	----

List of Figure

Figure 1: Model-View Controller scheme(Doherty, 2020).....	23
Figure 2: Screenshot of signup form	28
Figure 3: Screenshot of login page.....	28
Figure 4: Search doctor algorithm	29
Figure 5: Screenshot of search doctor result.....	29
Figure 6: Screenshot of modal appearing to send request to doctor.....	30
Figure 7: Screenshot of modal appearing to viewing request detail. Doctor's view on the left, Patient's View one the right.	31
Figure 8: Screenshot of patient profile page.....	32
Figure 9: Screenshot of doctor profile page	32
Figure 10: Screenshot of consultancy page(Doctor's view)	33
Figure 11: Screenshot of Review modal	33
Figure 12: Patient has closed consultancy(Doctor's view on the left, patient's view on the right)	34
Figure 13: Screenshot of closed consultancy page(Doctor's view)	34
Figure 14: Screenshot of consultancy/request block(Doctor's view)	35
Figure 15: Screenshot of doctors full card page	35
Figure 16: Screenshot of anamnesis page.....	36
Figure 17: Screenshot of registry page	37
Figure 18: Screenshot of modal appearing to add illness/treatment in registry.....	37
Figure 19: Screenshot of modal appearing to edit illness/treatment in registry.....	38
Figure 20: Example of help message on comments section.....	43
Figure 21: Medved web service's home page screenshot.....	54
Figure 22: OnlineDoctor web service's home page screenshot	54
Figure 23: Dr.Clinic web service's home page screenshot.....	54
Figure 24: Medihost web service's home page screenshot.....	55

Figure 25: 03Online service's home page screenshot.....	55
Figure 26: MedApp UML Class diagram.....	56
Figure 27: Questionnaire results for 20 potential users include both doctors and patients	62
Figure 28: Questionnaire results for 10 potential user patients	62
Figure 29: Questionnaire results for 10 potential doctor user.....	63

1. Introduction

1.1. Background

Nowadays Information technology and systems allow solving many problems in absolutely different areas. Examples start with basic day to day routine planning and end with major space discoveries. The question of public health is not an exception in this case. There are already thousands of examples when Information technologies are being used in surgeries, determining diagnosis, supporting the life of a critically damaged organism. One more example of using information technology in health care is telemedicine.

Telemedicine is „... delivery of health care services, where distance is a critical factor, by all health care professionals using information and communication technologies for the exchange of valid information for the diagnosis, treatment and prevention of disease and injuries, research and evaluation, and for the continuing education of health care providers, all in the interests of advancing the health of individuals and their communities“ (**WHO**, 2011:p.9). It has been possible to provide medical services distantly due to the development of the Internet as well as new ways for communication some of these are online video calls, emailing, live chats.

According to WHO there has been a substantial increase in the number of countries promoting telemedicine through applying policies and strategies in comparison to the last 5 years from the moment survey was conducted (**WHO**, 2016:pp.57-60). That may indicate rising recognition of the usefulness of telemedicine on a national level. But except for national or state representatives, private organisations also attempt to develop a full working system to support telemedicine by this making the progress in the deployment of telemedicine services quicker.

1.2. Problem Statement

There is a problem with inappropriate or ineffective treatment methodology provided by doctors to patients. From a doctor's perspective root cause of the problem may vary from a specialist's disinterest in his/her job to lack of access to proper information about the latest inventions and methodologies in medicine or about other specialists experience. While from the patients perspective problem aggravates by the lack of patient's opportunity to reach and communicate with different doctors independently from their location. Without proper access to various independent opinions, a patient cannot assess the proposed solution's appropriateness.

1.3.Aim

Develop a website with a search function allowing patients to find and contact a doctor through online chat available on the platform. Besides that patients and doctors can also have access to a section on web

site with an article written by a doctors where experts are sharing their experience related to a specific illness, treatment methods, new technological solutions in medicine. A developed website should make it easier to receive medical consultancy distantly as well as make more available and effective information flow from the medical world to the public and other specialists.

1.4.Objectives

To achieve the set above aim of the project following objectives should be reached.

- Develop web platform enabling online medical consultancies. Patient and doctors should be able to communicate via an integrated chat system and should have the opportunity to exchange document.
- Web-platform should allow patient to keep a record of illnesses and treatments in the registry as well as allow to complete anamnesis. Both registry and anamnesis should be visible to the doctor during a consultancy.
- Web-platform should allow doctors to publish articles that will allow sharing their knowledge and experience.

1.5.Report Structure

Achievement of my set aims requires conducting background research, understanding system users, designing, implementing, testing a prototype and finally evaluating work done. These start in chapter 1 from understanding the problem we want to solve with background research. Next, we set objectives that should be reached to achieve set aim. The second chapter of report is understanding potential users and their needs. For this reason, we have designed and conducted interviews with doctors and patients. A part of the second phase is also analysing the existing systems providing similar services. Existing systems that already operate can be seen as an indirect indicator of users needs and can be used to identify new features that can be useful in proposed system. The third chapter of project is about the actual design of the system with the following requirements list and development aspects described. But first, we have to identify roles and their core interactions with the system. In the design phase, we should also think about the structure of the whole application and identify what is going to be implemented in the prototype. Once we have the design decisions made, ready models, diagrams, requirements we will move on to actual implementation. In chapter 4 we are discussing the implementation of core functionalities. In Chapter 5 we test our application and ensure that no bugs or errors arise. Chapter 6 refers to evaluation of developed prototype's strengths and weaknesses. Finally, the last chapter is the conclusion to the whole project with summarised achievements and challenges

we faced during the whole project, with a description of further work to be done and conclusion to the report.

2.Related work

2.1.Interviews

2.1.1.Why do we need interviews?

In this project, there are two main reasons for carrying out interviews. The first one is that we need to gather requirements for the system. And potential user's opinions on what functionality is needed, might be very useful to hear. We can do it with the questioner, but for higher quality answers it is better to perform interviews. The second reason is to find answers to doubtful issues related to the project. For example, during the design of the system, I was thinking about doctors motives to spend his/her time consulting people online.

Note that I am not planning to carry on two different sets of interviews for each of the categories: RQs elicitation and doubtful issues since doctors are being very busy. Interviews will be designed to serve both purposes simultaneously.

2.1.2.Who are the interviewees?

We are going to interview up to 4 doctors with work experience of over 7 years. These are doctors specialised in different areas from different countries and with different backgrounds. There are many doctors around me, but these are the ones with whom I was able to arrange meetings. The second group of people are regular patients. I plan to interview 4 patients that are potential users of the system.

The interview plan and structure is presented in the Appendix B. Interviewees response can be viewed in supporting materials.

2.1.3.Interview analysis

Doctors:

For our 4 interviews above we had 4 different people with different specialisation, experience and even country of work. It can be observed that all of them stated that the system will be demanded and useful to both patients and doctors. Moreover, they recognised the importance of addressing both sides of the stated problem. Patients can easily find doctors and contact them with no need to travel from place to place. While doctors can provide their services to wider auditory and enhance their knowledge there will be a section where specialists can share their experience and knowledge.

All 4 interviewees mentioned that usually, even the busiest doctors can find time to participate in such a system, but in return, such a platform should bring them benefits in form of clients or finances. Things like achievements were pointed

out by interviews as an important addition to motivating doctors to participate even more actively, but still not enough to keep them.

One of the most interesting findings for this project is the facts that online meaningful consultancies are possible for various doctors specialisations although degree of consultancy varies.

According to the ENT(Ear Nose and Throat) doctor, he can only consult about different solutions, operations and methods in medicine, but cannot write a receipt for the patients since doctors must observe the problem in real life. According to the surgeon he can initially state a diagnose based on pictures send and therefore tell about possible operations, but of course, before a real surgery doctor will have to look again at the patient's condition. On the other hand, a paediatrician can determine a diagnosis as well as write a receipts for solving existing illness in most cases if patients send needed documents and photo or video. Similarly, an endocrinologist can form the required diet for patients once required analysis results are sent to a specialist.

The above justifies that regardless of the specialisation of illness and doctors, the proposed platform still can be useful for consultancies. But for some doctor specialisations, it can even be used to provide the whole treatment process.

Apart from the points discussed above interviews have also mentioned some features and functionality they expect a proper system to have. These include a medical registry of illnesses and treatments, anamnesis and bidirectional review mechanisms between patient and doctor.

In the development phase, we will come back to interviews to justify the importance of some features.

Patients:

Similarly to doctors patients expect online medical consultancy services to be demanded. Reasons for using such application are: the opportunity to contact different doctors from home, receive some services online, have access to an independent opinion. Patients have also agreed that the proposed system should address both discussed issues with the stated problem since this will allow the solving problem of ineffective treatment or medical services provided by doctors to patients from both sides simultaneously.

During interviews, I found out that each patient I talked with had at least once experienced poor medical service. In most of the cases, the reason was wrong procedures carried on in order to treat a disease. In best case, ineffective treatment was just a waste of finances, while for one of my interviewee's mistreatment almost cost them their life . All Interviewees mentioned that often doctors are working hard to help them but later patients found out from more qualified specialists that there are less effortless and less risky and more effective methodologies. According to patient's opinion reason for such a

difference of knowledge between specialists is no organised effective and simple communication between doctors that could help establish a ground for effective experience and knowledge sharing.

One of the most important aspects raised at the interview was a question of personal data and trust to share patients problem with doctors. All 4 interviewees told that they do not mind sharing their problem online but only if there will be some technical or legal way to control confidentiality.

Patients do not mind paying for consultancies even if these are online and recognised importance for doctors to earn on this platform. All 4 interviewees had experience at least one treatment or another medical service that could be provided online through proposed system that indicates potential demand for the platform.

Interview Conclusion

Conducted interviews with doctors and patients lead to a better understanding of their view on proposed system as well as helped to find out answers to doubtful questions. Moreover, we have heard about their preferred features that should be in this system. Some of them are going to be added to the requirements list while others may be left as an optional requirements. Undoubtedly enough attention should be paid to the development of the feature „Registry of illnesses and treatments“ since every interviewee doctor has pointed it out as an important feature the system should have. I have also found answers to the questions I had in general, not specifically related to features of the proposed system. These are helping to understand the main users of the system better and therefore to develop a better application.

2.2.Existing systems analysis

Part of the RQs elicitation process is analysing already existing systems that provide similar services like the one I propose. I have found out that several websites that host remote medical consultancy. The selection was made based on one criterion: serving purposes we have set. Therefore we have only those websites that provide online medical consultancy and might have an articles section. Meanwhile, let me note that we will focus on Russian telemedicine websites. Due to COVID-19 I couldn't travel back to the UK and had to stay at home in Russia. Since I cannot effectively contact UK services from Russia upon demand, basing analysis of Russian websites will save a lot of time and enable more effective analysis.

In Appendix C structure of the analysis conducted is presented.

2.2.1. Existing systems list

- <https://telemed.chat> (Medved) ;
- <https://onlinedoctor.ru/doctors/> (OnlineDoctor);
- <https://www.drclinics.ru> (DrClinics);
- <https://medihost.ru> (Medihost);
- <https://03online.com> (03Online);

2.2.2. Comparison

How focused is a website on providing online consultancy?

Homepage

Let's start with Medved. The search bar is the first element displayed on the home page of the Medved website where a user is expected to type in the specialisation of a doctor. A similar design is made by OnlineDoctor, but the difference is that OnlineDoctor provides a search bar with several filters for the search: full name of a doctor, child/adult specialist, specialisation, etc.

DrClinics displays on the home page a standard site with availability to read about an organisation, select a service, see contacts, etc. To enter the telemedicine part of the site we have to go into the „services“ tab and select „Online consultancy“. When clicking „Online consultancy“ the user will be redirected to a page with two boxes: one is to contact the main doctor immediately another is to select a special doctor through a search bar. Nearly similar design made by Medihost, where the home page does not meet visitor with the opportunity to search for a doctor immediately, instead it has a search bar to find clinics and make an appointment. To find telemedicine user has to click on „Consultancy“ and then on „Telemedicine consultancy“. Next, a new page opens with a search bar enabling the search of a doctor for online consultancy. 03Online meets visitor with information about the website and a list of specialisation to select immediately.

In Appendix D screenshot of each existing system is provided.

Services provided

OnlineDoctor, 03Online provide only online consultancy service, while Medved, Medihost also provide an opportunity to make an appointment in a clinic. DrClinics provides online consultancy, as well as standard medical services to be done in their clinics since it's a clinic primary focused on standard interaction with patients rather than online.

It can be concluded that OnlineDoctor is most focused on telemedicine since it does not provide other services and its home page design engages searching doctors. Next comes Medved and 03Online. Least focused is DrClinics since it has a lot of information and provides many other services.

How strong does it engage direct communication between patients and doctors?

OnlineDoctor and Medved strongly engage in communication since they allow users to contact doctor easily, not requiring many steps to be done. Also on these websites, intuitive design helps to find a contact specialist. While DrClinics, Medihost have many different tabs, sections on one page that may confuse the new user who looks for an opportunity to contact a doctor. About 03Online I cannot say it engages communication between patient and doctor since it does not provide an opportunity to contact a specific doctor directly. Instead users attention is focused on sending a question to a random doctor.

Type of consultancy implementation: online chat, video call, other

OnlineDoctor provides consultancy mainly through video calls, while 03Online and Medihost provide consultancy only in form of emailing/chats. Dr.Clinics provides consultancy through emailing and phone calls. Meanwhile, maybe the best option is provided by Medved that has enabled online chat as well as video call to support interaction between patient and doctor.

When deciding which option is best to provide consultancy we should also pay attention to how these options are implemented. For example, chat function is available on Medved, Medihost, 03Online, but Medved has an intuitive, user-friendly interface of chat that enables simple communication, while others have poor design, with small font size, unattractive colour set and complicated navigation.

Patient Profiles

One of the most important features mentioned by doctors in the interview was having kind of a record for medical registry for patients that will consist of illnesses and treatments or anamnesis that may help a doctor to understand the patient better. But none of the analysed systems has such a feature. Their profile data is limited to trivial contact information and personal data like a place of work, email, etc.

Review system

All 5 examples have a standard review system. When patients write their review and evaluate doctor on 5 star basis. There is no complicated and detailed system of doctor evaluation that can help to analyse doctors performance thoroughly.

Interesting/unique features

Among other services discussed Medihost provide an opportunity for a user to contribute to the promotion of a doctor on the website by paying for a specific advertisement service. These ad services cost very little by this making such a way of expressing gratefulness attractive for patients, but the disadvantage is that anyone can use this feature to promote themselves. In this case, it would make more sense to enable it for patients who already received consultancy from a doctor.

Connection of doctors: Although the 5 services we analyse are serving similar purposes they have a different way of forming. Medved, Medihost and 03Online

allow anyone to register as a doctor and after moderation, the specialist can start to work. Meanwhile OnlineDoctor, DrClinics carry on the recruitment process for doctors and doctors on the website act more like employees of service rather than independent specialists. So, last two websites are acting as real clinics with their doctors.

Medihost has an indicator that shows whether a doctor is currently online or not. It is a very useful feature that will for sure help patients who need to contact someone urgently.

'Select doctor' Use case steps

The next stage of analysis is the comparison appearance of the use case: select a doctor on different websites. I have selected „Select doctor“ use cases to compare different websites on how easily one of the core functionalities is provided.

Medved:

Prerequisite: User is logged-in
Steps: Enter a homepage > Select specialisation > Select a doctor from a list>Select available time for an appointment>Pay for consultancy.

OnlineDoctor:

Prerequisite: User is logged-in
Steps: Enter a homepage > Select specialisation > Select a doctor from a list>Select available time for an appointment> Pay for consultancy.

DrClinics:

Prerequisite: User is logged-in
Steps: Enter a homepage >Open „Services“ tab> Select „Online Consultancy“>Click on „Make an appointment“> Select specialisation > Select a doctor from a list>Select available time for an appointment> Pay for consultancy.

Medihost:

Prerequisite: None
Steps: Enter a homepage >Open „Consultancy“ tab> Select „Online Consultancy“>Click on „Make an appointment“> Select specialisation > Select a doctor from a list>Enter patients details(Full name, email, etc)>Select available time for an appointment>Receive confirmation from doctor>Pay for consultancy.

03Online:

Prerequisite: None
Steps: Enter a homepage >Select specialisation > Select a doctor from a list>Click on „Ask a question“ button>Enter patients details(Full name, email, documents, etc)>Pay for consultancy.

Medihost and DrClinics have the longest path for use case: select a doctor that plays a critical role in serving one of my app's purposes, providing online

consultancy. These 2 websites have the longest path because they are providing telemedicine as an extra feature, service therefore the user has to go through extra steps to find out a doctor. While other websites strongly focus on online consultancy and enable selecting a doctor immediately once the user has entered the home page. Once proposed system is implemented, we will have to compare length of tested use case of MedApp and existing systems one to check effectiveness.

2.2.3. Conclusion

Overall there is a fundamental difference in the position of the discussed system. OnlineDoctor, 03Online position themselves as an alternative for visiting real clinics with their doctors being hired. However, Medved, DrClinics, Medihost allow independent doctors to register on the website freely and start providing their services. The second positioning is much closer to this idea and concept. Since I do not want to position as 'online' clinic hiring just a few doctors for each area. Instead, this project aims to provide a large variety for patients and make the platform a place to bring patients and doctors together.

3.Design and Requirements

3.1. Users and User interactions

One of the steps in this project is designing the system. By design here I mean determining functional RQs and non-functional RQs for a system that is expected to be implemented, planning users interaction with the system and making UML diagrams to visualise the software part. The first thing I should start from in designing a system is identifying its users.

Users

Expected users are:

1. **Guest** - user that is not registered in the system and has access only to searching doctors without contacting them, without leaving reviews, with access to read articles published.
2. **Patient** - a regular person that has a problem and seeking consultancy or treatment. Age category expected: 18+. Children below 18 are not expected to contact doctors and request consultancy. The wage category of a potential customer may vary.
3. **Doctor** - a regular professional with medical education confirmed by a specific certificate, diploma. The age of doctor is expected to be over 26 since younger people are still studying in universities in the UK as well as in Russia. For this platform, I expect to prevent graduate doctors without any work experience to register on the website but to realise this limitation this project may require moderators doing experience checking manually.
4. **Moderator** - employee doing procedural tasks to control that rules of the website are not violated. If any issue is raised by other users moderator have to deal with it in most of the cases except for some technical faults. Moderator can be seen as customer supporter.
5. **Admin** - a person with technical background, having the strong technical knowledge to keep the system working.

User interaction

Let's start by describing user interaction. In this system users are expected to interact in many different ways. Some of these are:

Guest

1. **Signup** for a website. Expected to have 2 types of accounts(Doctors/ Patients).

2. **Search for a doctor.** Guests will be able to search for a doctor on the website but cannot contact that doctor until a request sent to doctors and the doctor accepts it.

Patient/Doctor

1. **Logging** into a website account.
2. **The chatting** function will be available for both types of the users only if there is an active consultancy between them.
3. **Sending documents** one of the essential features needed for the system to serve its purpose.
4. **Closing consultancy** is an action that will enable both sides to confirm that a service was provided and the system may count consultancy received to a patient and provided by a doctor. It happens through clicking the specific button by patients and doctor. However if the patient, for example, does not agree that doctor provided consultancy then a patient may click a specific button and write an explanation for the moderator. Then the message and request will be sent to a moderator who can start procedural activities to resolve issues, disputes. Meanwhile, a doctor may not close the dialogue if he/she thinks that something is left to send or say to the patient or if any problems arise and further investigation required from moderators.
5. **Leaving a review**
 - For patients leaving a review about the doctor is essential since its one of the criteria playing an important role for other patients to select a doctor.
 - However, leaving a review about the patient may also be important and useful to inform other doctors about possible difficulties arising with patients or patients behaviour. A similar mechanism is implemented in Uber. Both driver and passenger can leave feedback about each other,

Patient

1. **Search for a doctor.** Patients will be able to search for a doctor on a website using a search engine.
2. **Send a request.** Once the patient found a doctor he/she might send a request to establish a dialogue.
3. **Leave a review about articles.** Patients may have some experience related to the topic of the article and leaving a review can be helpful for

others to share the experience.

Doctor

1. **Provide consultancy** to the patient through a web platform.
2. **Confirm an article** written by another doctor. To ensure that from the medical point of view the article is correct not fictional doctors can read articles and confirm or reject them. Several confirmations will make people trust it more. There will be a minimum number of confirmations for an article to be published. Doctors approved articles will be recorded in the system to prevent approval of every single article without reading it.
3. **Publish an article**. Since one of the problems we try to solve through MedApp is making a more effective share of experience we must have a section where doctors can write an article to share their experience on a specific problem and issue quicker and effectively with the public

Moderator

1. **Answer users queries**. Moderators primary task is to support users using the system through solving queries.
2. **Review user feedback**. Moderator expected to review feedback left by patient or doctor to prevent publishing inappropriate review violating rules.
3. **Check articles**. Moderators expected to check articles being published on a portal by doctors to avoid publishing article with inappropriate language, advertising, etc.
4. **Resolve disputes** between doctor and patient if one arises.

3.2. Functional Requirements

ID	Requirement	Use case	Completed
1	Guest must be able to create account	Signup	✓
2	Patient and Doctor must be able to login into their account using email/password	Logging	✓
3	Patients and Doctors must be able to edit their profile data	Edit profile	✓

ID	Requirement	Use case	Completed
4	Guest must register in system as a Patient or a Doctor	Register Patient account/ Register doctor account	✓
5	Doctor must be able to add work, education related documents on their profile.	Add doctors documents	✓
6	Patients must be able to keep record illnesses and treatments in registry	Edit medical registry	✓
7	Patient must be able to create an entry on illness/treatment with related information	Create illness/ treatment card	✓
8	Patient must be able to edit or delete illness/treatment card	Edit illness/ treatment card	✓
9	Patients must be able to attach documents to illnesses and treatments	Edit medical registry	✓
10	Patients must be able to complete anamnesis	Complete anamnesis	✓
11	Both Guest and Patient must be able to search a doctor based on search query matching	Search a doctor	✓
12	Patients must be able to send a request for consultancy that should include brief description of their problem	Send a request	✓
13	Patients must be able to attach documents to request.	Send a request	✓
14	Doctor must be able to confirm or reject request	Reply on request	✓
15	Patient and Doctor must be able to contact each other through a chat on website	Chatting	✓
16	Patient and Doctor must be able to send documents to each other while there is an active consultancy	Sending documents	✓
17	Doctor must be able to view patients registry and anamnesis only if there is an active consultancy with that patient.	Access registry and anamnesis	✓

ID	Requirement	Use case	Completed
18	Both Patient and Doctor must close the dialogue to confirm that consultancy was provided	Closing dialogue	✓
19	Patient and doctor must be able to write review about each other upon closing consultancy	Writing reviews	✓
20	All users must be able browse articles with several confirmations from doctors	Read an article	✓
22	Doctor must be able to publish an article on website	Publish an article	✓
21	Doctor must be able to confirm an article written by another doctor and not yet published on website	Confirm article	✓
23	Doctors and Patients must be able to respond on others review under the article	Respond on review	✓
24	Doctor must be able to remove his/her article from website	Remove article	✓
25	Doctors and Patients must be able to leave feedback on an article	Leave a review for article	✓
26	Patients must be able to add doctors to a „Favourite“ list	Add favourite doctor	✓
27	Doctor and Patient must be able to add articles to a „Favourite“ list	Add favourite article	✓
28	Upon request or consultancy action an email notification to participant must be sent.	Email notification	✓

Table 1: Functional Requirements List

3.3. Non-Functional Requirements

ID	Requirement	Completed
1	System should respond to user interaction within reasonable amount of time without major delays caused by software related issues.	✓
2	User passwords should be hashed before being stored in database	✓

3	Users should be able to access application and use all features from devices with operating system Linux, Windows, MacOs, Android, IOS	
4	Website pages should adapt appropriately under different screen sizes	
5	System must have intuitive design	

Table 2: Non-functional Requirements List

3.4. Design Pattern

For this web application, I have decided to make use of model view controller(MVC) architecture displayed on Figure 1. The advantages of the MVC pattern is that it facilitates the adoption of the separation of concerns principle. It allows to separate Model(Databases) from Controller(Business Logic) from View(HTML template)(Dragos-Paul & Altar, 2014). Separating concerns across different modules allows to avoid excessive dependence of one block from another and therefore minimise the risk of failure in one block to disable the entire system.

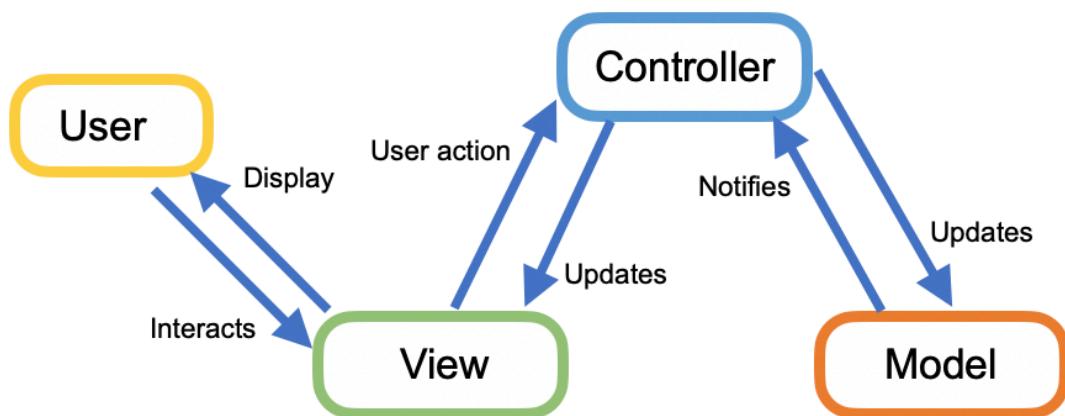


Figure 1: Model-View Controller scheme(Doherty, 2020)

3.5. System design principles

During system design and implementation, I have been followings various essential design principle in software engineering. 4 different principles have many overlapping points and have similar aims. The following one supports the following of another one as well.

1. DRY(Do not Repeat Yourself) - „...says that every piece of system knowledge should have one authoritative, unambiguous representation“(Miller, 2007). Following the DRY principle means for us to aim developing single representation of a logic in the system. Following DRY principle allows to minimise the size of code as well as a

- lower chance of an error during the development and simplify the volume of code to be tested.
2. Separation of Concerns(SoC) - on implementation level refers to separation of blocks of code addressing different concerns(Hursch and Lopes, 1995).
 3. High Cohesion - an element of the system should include only that business-logic that relates to each other(Stevens, Myers and Constantine, 1974a). If we have low cohesion then it means that a function carries on a very broad range of responsibilities that are not necessarily have any relation to each other.
 4. Low coupling - system has low coupling if various components interconnectedness is minimised(Stevens, Myers and Constantine, 1974b). Therefore potential influence of one component's change on another's functionality is also lowered.

3.6. Development approach

During the development process, I will adopt a waterfall model since it is more suitable for cases when the developer has a clear understanding of what should be developed and how it should be developed(Wysocki, 2009, p.300). In this case, I have a clear understanding of what this platform should do and how to realise the proposed functionality. I do not need to overlook constantly the system design and implementation approach. However, in the end I will have to consider some improvements once user feedback on a prototype is received. But feedback sessions will be held upon completion of prototype development with most of the functionalities developed and tested.

3.7. Development methodology

Overall prototype development will be separated into 3 major phases. Phase 1 is the development of webpage structure and design. Identifying the structure and layout of the webpage will be helpful to understand how to implement business logic since Phase 1 will already facilitate thinking about user interaction with the page directly. Moreover developing a webpage template can be seen as a visualisation of expected behaviour but without an implemented logic. Phase 2 is a design of database architecture, that requires the identification of database tables and attributes for each table. Without database development, I cannot develop business logic that is expected to work tightly with the database. Phase 3 is the implementation of business logic for a web application.

Note that for Phase 3 I do not make explicit separation of frontend and backend development of business logic. My development process can be described more like a function-oriented development. That means at a moment I attempt to implement one functionality's frontend(JavaScript) and backend(Python) parts simultaneously. That will allow me to focus on functionality and avoid misunderstanding and confusion of my codes and processes.

Moreover, the development of business logic often requires working with fronted logic(JavaScript) and backend logic(Python) simultaneously since these parts are strongly related in order to provide functionality.

4. Implementation

The current chapter describes an implementation of some core functionality. Features described below are those required for providing consultancy.

Scenario: Patient searches doctor through the main page, sends a request to doctor. The patient attaches documents to request. The doctor accepts the request and provides consultancy. Doctor attach a document to consultancy. The patient closes consultancy and leaves a review about the doctor. The same is done by a doctor. Consultancy ended.

4.1. Languages used

This section describes technologies used in order to implement prototype.

4.1.1. Web-framework and Language for Backend

For this application, we will use the Django web framework. The first reason to select the Django framework is that it adopts the MVC architecture that we planned to implement. Second, it allows to minimise dealing with low-level network technologies and provides many API making development process quicker and more effective. Moreover, it provides ORM allowing to manage database without getting into SQL by this minimises the risk of database-related bugs. The third Django framework provides inbuilt security features. These are input sanitisation preventing from some type of injections attacks and cross-site scripting. Also, the use of ORM in Django prevents SQL injections. Django provides hashing and salting of user passwords based on the SHA-256 algorithm. Finally, Django is built on Python, a programming language providing versatile features and various ready-made methods.

4.1.2. Frontend Logic Technologies

For frontend logic, we will use JavaScript language and jQuery library. These allow making webpage dynamic and interactive. jQuery provides a rich set of shorthand functions allowing it to save time during the development. Moreover, jQuery allows dealing with interpretability across the browser. One of the most important jQuery features is that it provides AJAX functionality.

AJAX plays a crucial role in developing an effective system since AJAX allows to conduct client-server interaction in the background without updating the entire page. This allows to minimise traffic load on the server and improve user experience. Since the user can continue using other feature provided on the webpage while a specific element is awaiting for AJAX request to be done. Therefore for functionalities that are possible to be provided without a page refresh, we will use AJAX.

4.1.3. Frontend Structure and Design

For page structure and design, we have to use HTML and CSS respectively. These are core and fundamental for webpage development. However, we will also make use of the Bootstrap framework since it provides a Bootstrap grid

that facilitates and simplifies development responsive to the screen size design of an entire page.

4.2.Database setting

The class diagram in Appendix E is also representative of various objects interactions. It helps to get an overall understanding of system architecture. For the development of the database(DB) architecture, we will orient on a class diagram.

Each class in diagrams will be converted into a database tables with relative attributes. The only outstanding exception is that the Many-to-many relationship between a pair of tables in Django is controlled through creating a new table with „Entry ID“, „ID of the first element“, „ID of second elements“. These many-to-many tables are not stated in the class diagram, since these are made on the background of Django, and does not affect much on system.

As it was stated previously Django web framework provides in-built Object Relational Mapping(ORM) to manage Database. Therefore we won't make direct use of Structured Query Language(SQL) for project instead management is done through Python.

Django by default creates a Database based on SQLLight Database Management System(DBSM). Since we do not have any reason to use other DBSMs we will stay with SQLLight.

4.3. Core functionality

Login/Signup

To make use of this system users have to sign up. The slider in the signup form allows switching between patient and doctors account signup. If the switch turned on, it will add extra fields specific to doctor accounts. One of these is the dropdown specialisations list coloured green.

Once an account is created user should log in. Without authentication, a patient cannot request consultancy. For MedApp I Django's in-built authentication system was used, that requires to use Django's „User“ class. Django authentication is providing hashing of user password and authenticating session value on the client-side.

Figure 2: Screenshot of signup form

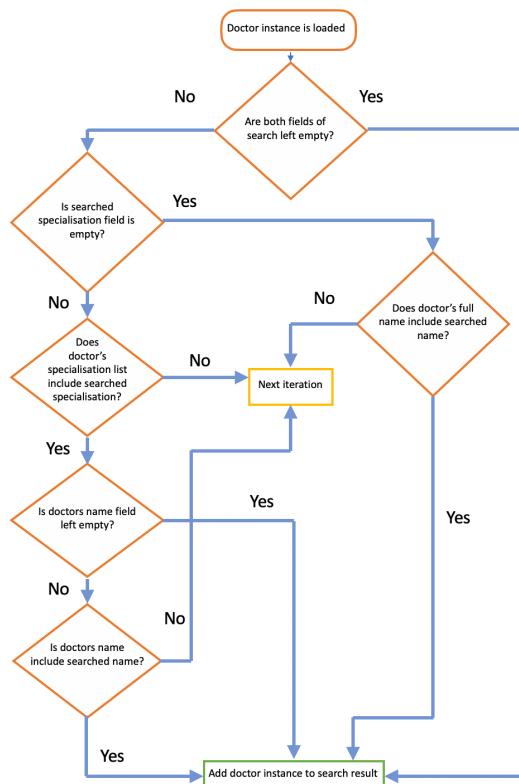
The screenshot shows a blue header bar with the text "MEDApp Home Articles" on the left and "Login Sign-up" on the right. Below the header is a white form area with a "Log In" title at the top center. There are two input fields: "Username" containing "aBagirov" and "Password" which is empty. Below the password field is a red error message: "This pair of username/password was not found!". A blue callout box with a rounded rectangle and a thin border points to this error message, containing the text "Error message displayed if authentication failed". At the bottom of the form is a blue "Log-in" button.

Figure 3: Screenshot of login page

Search doctors

After authentication patient is expected to search doctor. Figure 5 shows an example of search output on speciality „Surgeon“ and doctor's name „Akshin“. The screenshot shows the overall search results structure. For each doctor matching the search query, there will be present a card giving overall information about the specialist.

Search algorithm giving overall understanding about how result output is present on Figure 4. User has to input speciality of doctor and input name of a doctor to search. If we leave both fields blank, the search will return every doctor in the database. If we have at least a speciality or name inputted then the doctor matching query will be returned. If both fields are entered, the only doctor matching both queries will be sent to the client-side.



SignUp

Complete form to sign up

Do you want to create doctor account?

Male
 Female

Email address
bagirov@mail.ru

We'll never share your email with anyone else.

Telephone
+79151257689

Username
aBagirov

Akshin

Second Name
Bagirov

Date of Birth
ДД.ММ.ГГГГ

City Name
Moscow

Country Name
Russia

Work place
Bagirov clinics

Education
Sechenov 1-st Med

About me
I am ...

Experience: when did you start working?
ДД.ММ.ГГГГ

Password

Confirm Password

Select your specialization

Submit

Submit

If there is any error on sign up form: username exists, passwords do not match, etc an error message is displayed to user and submit button is disable until error is fixed.

Figure 4: Search doctor algorithm



Search your doctor at MedAPP

Surgeon Akshin

Here is search result!

Akshin Bagirly
Moscow , Russia
Age: 54 Years | Male

Work: Bagirovs Clinic No 1 in Moscow, Russia!
Education: Sechenov University
About me: Hello, I am Akshin surgeon from Russia!

31

Doctors experience in years

Upon click modal displayed to send request to a specific doctor. Modal shown in Figure 6.

Request consultancy

Figure 5: Screenshot of search doctor result

In chapter 2.2.2 we assessed existing softwares „Select doctor“ use case in order to identify one that requires minimum number of steps to make a request to doctor.

Below we have comparison of MedApp's „Select doctor“ use case implementation and Medved's one that is one of the quickest from 5 existing systems.

Medved:

Prerequisite: User is logged-in

Steps: Enter a homepage > Select specialisation > Select a doctor from a list>Select available time for an appointment>Pay for consultancy

MedApp

Prerequisite: User is logged-in

Steps: Enter a homepage > Enter search query>Send request.

For MedApp it takes only 3 steps to request consultancy if the user already knows the target doctor's name. That is mainly achieved through enabling user not only to input a specialisation but also doctors name that make it possible to find required doctors quicker. While on Medved site even knowing the needed doctor's name does not allow to search doctor as quick since a user can only select a specialisation and then search needed doctor in the list manually.

However, even if the user does not know the exact doctor's name at MedApp it will take only 4 steps to request consultancy since the „Select Doctor“ step is added. While for Medved user must select an appointment time.

Request

Once the doctor is selected user has to send a request for consultancy. Consultancy request consists of two parts. First is a modal with the message send to specific doctors shown in Figure 6 and second is a modal accessible from the profile page where extra features like adding documents to request are available, shown in Figure 7.

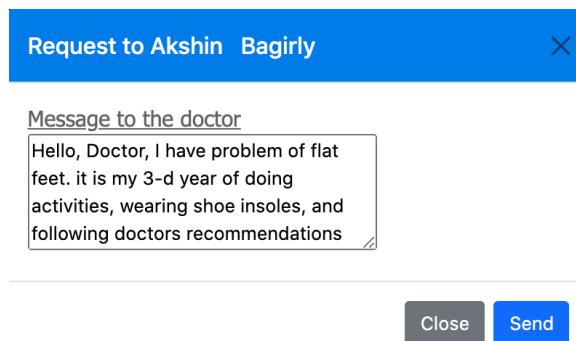


Figure 6: Screenshot of modal appearing to send request to doctor.

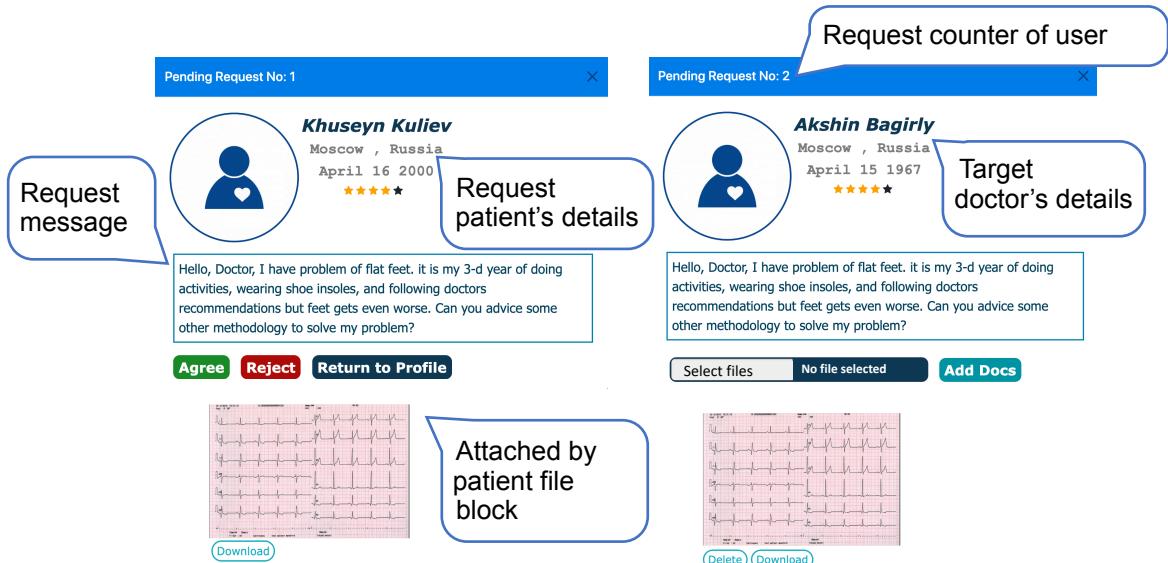


Figure 7: Screenshot of modal appearing to viewing request detail. Doctor's view on the left, Patient's View one the right.

Upon sending a request message a request object is created in the system and the doctor will receive an email notification with text example:

"Dr Akshin, Patient Khuseyn Kuliev has requested consultancy!"

Patient is from: Moscow

Patient tel: +79856742323

Patient email: ec18266@qmul.ac.uk"

Profile Pages

Except dealing with request profile page also allows users to keep track of their detail and control their interaction. Doctors and patients have an overall similar layout of profile pages. In Figure 8 and 9 we can see the components of each page and identify differences.

In each block on the page, there is an instruction written in form of a list that will be present to users once they signup. Instructions are explaining the purpose of each block and will disappear once a user gets at least 5 entries in each block. If the user has less than 5 entries, then instruction will still appear but after the entries.

The screenshot shows a patient profile page with the following annotations:

- Profile card with main data about user**: Shows the user's name (Khuseyn Kuliev), birthdate (April 16, 2000), location (Moscow, Russia), gender (Male), and a 5-star rating.
- Features available only to patients**: Includes links for "Edit Profile", "Registry", and "Anamnesis".
- Favourite articles and doctors block with links to actual article or doctor**: Displays sections for "Favourite articles" and "Favourite doctors" with placeholder text.
- Hello, Khuseyn**: Greeting message.
- Current Consultancies**: Placeholder text for active consultancies.
- Requests**: Placeholder text for requests.
- Past Consultancies**: Placeholder text for ended consultancies.
- Logout Profile Hello, Khuseyn!**: Navigation links.

Figure 8: Screenshot of patient profile page

The screenshot shows a doctor profile page with the following annotations:

- Profile image can be changed through „Edit Profile“**: Shows the user's name (Akshin Bagirly), age (31 years), birthdate (April 15, 1967), location (Moscow, Russia), and a 5-star rating.
- Files block**: Placeholder for files, showing "My files! Click to show!" and three certificate thumbnails.
- Block with links to consultancy pages**: Placeholder text for active consultancies.
- Block with links to open request modals**: Placeholder text for requests.
- Current Consultancies**: Placeholder text for active consultancies.
- Requests**: Placeholder text for requests.
- Past Consultancies**: Placeholder text for ended consultancies.
- Articles for approvals**: Placeholder text for articles pending approval.
- Hello, Doctor Akshin**: Greeting message.
- Logout Profile Hello, Khuseyn!**: Navigation links.

Figure 9: Screenshot of doctor profile page

Consultancy

If the doctor accepts a request through a modal window(Figure 7), a consultancy object is created with data from the request object moved into the consultancy object, this includes files attached to the request. While constancy between Patient and Doctor is active, Doctor will have access to patient registry and anamnesis.

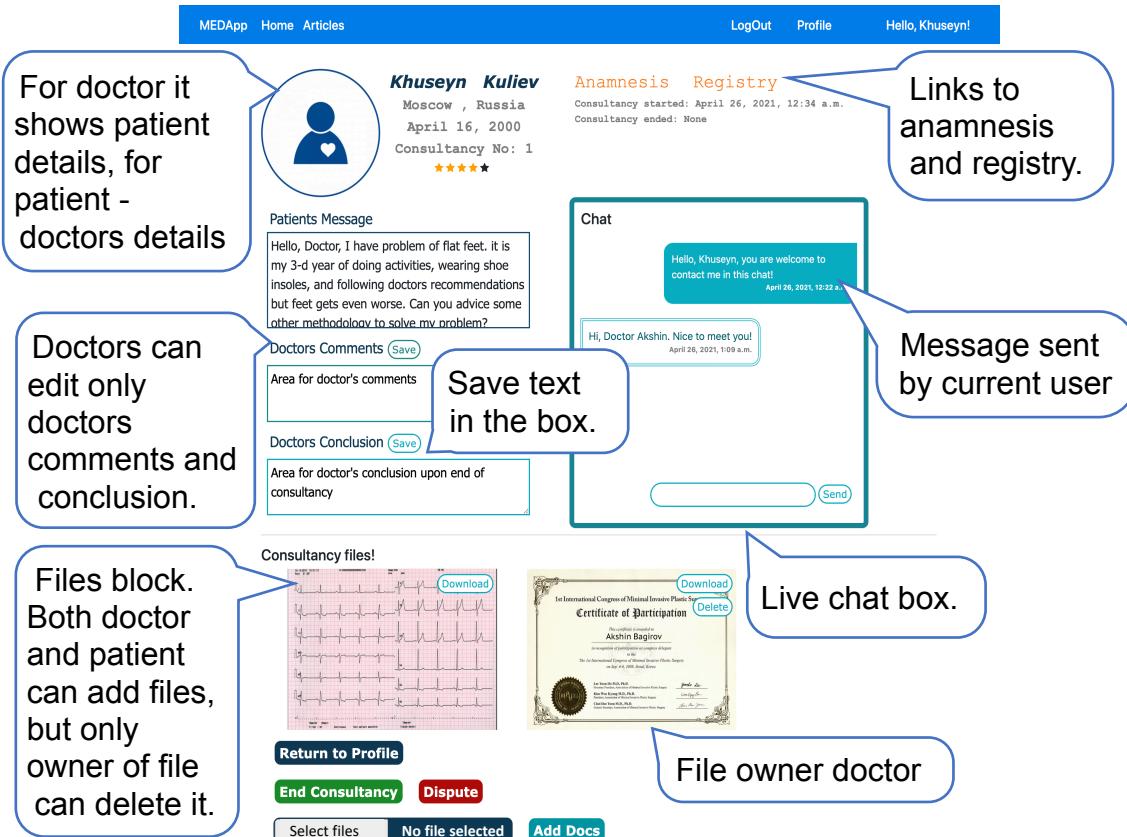


Figure 10: Screenshot of consultancy page(Doctor's view)

To access the consultancy page user has to click the relative link in the „Current consultancy“ block. The consultancy page is shown in Figure 10.

When the user ends consultancy a review modal shown in Figure 11 is displayed where a review about another user can be left. Review about the doctor is published in doctor's full card Figure 15. Review about patient won't be published, but the rating will be changed.

Please give a review about doctor... ×

Rate doctor:
★★★★★

Give a review about doctor:
Doctor Akshin, was very helpful. He informed me about new solution to flat feet. Soon I am going to try it, since I don't have my other option. I advice Akshin to write an article about

Please give a review about patient... ×

Rate patient:
★★★★★

Was patient problematic? :
Khuseyn was friendly and polite. No issues to report!

Close **End consultancy**

Close **End consultancy**

Figure 11: Screenshot of Review modal

Once consultancy is closed by one of the sides a reminder message appears on is consultancy page and user that has closed consultancy won't be able to add any more documents, but still can delete them.

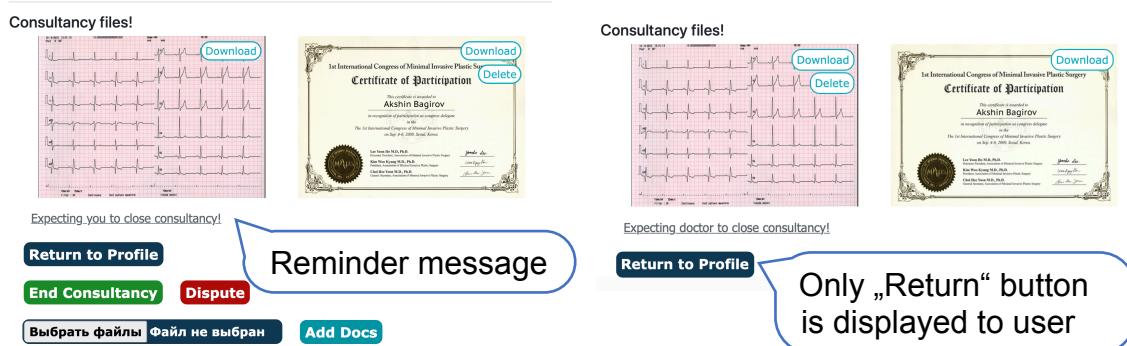


Figure 12: Patient has closed consultancy(Doctor's view on the left, patient's view on the right)

When consultancy is closed relative message displayed and close time is a recorder. Chat remains available even after closing consultancy for some urgent issues. Closed consultancies' page shown in Figure 13.

The screenshot shows a web application interface with a blue header bar containing 'MEDApp', 'Home', 'Articles', 'LogOut', 'Profile', and 'Hello, Khuseyn!'. Below the header, a profile section for 'Khuseyn Kuliev' (Moscow, Russia, April 16, 2000, Consultancy No: 1) is displayed. A message box states 'End date is set'. The main content area includes a 'Patients Message' box with text about flat feet, a 'Doctors Comments' box with text about shoe insoles, a 'Doctors Conclusion' box with surgery recommendations, and a 'Chat' box with a message from the doctor. A 'Consultancy files!' section at the bottom shows an ECG image and a certificate of participation for 'Akshin Bagirov'. A text box indicates 'Status is closed'.

Fields are non-editable

Status is closed

Figure 13: Screenshot of closed consultancy page(Doctor's view)

Request/Consultancies block

Figure 14 shows how block is changing through out the normal flow of consultancy and requests.

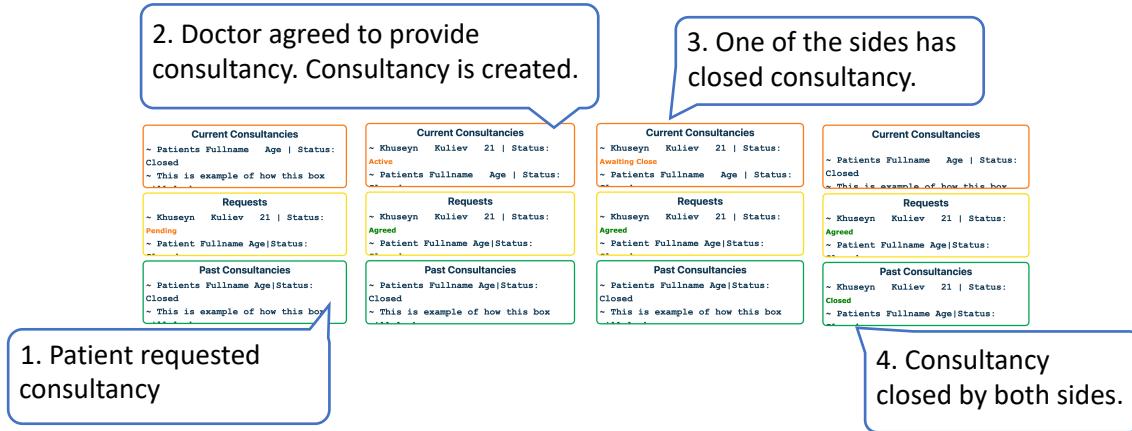


Figure 14: Screenshot of consultancy/request block(Doctor's view)

Doctor's Full Card

As it was stated above once a patient leaves a review about a doctor, that review is shown in the doctors full card. The full card is one being displayed upon clicking on the doctor's full name on the search result page(Figure 5). Compared to search result doctor card, the full card contains reviews and files uploaded by doctors from the profile page. We expect doctors to publish some certificates, awards, diplomas that will enhance patients trust to doctor.

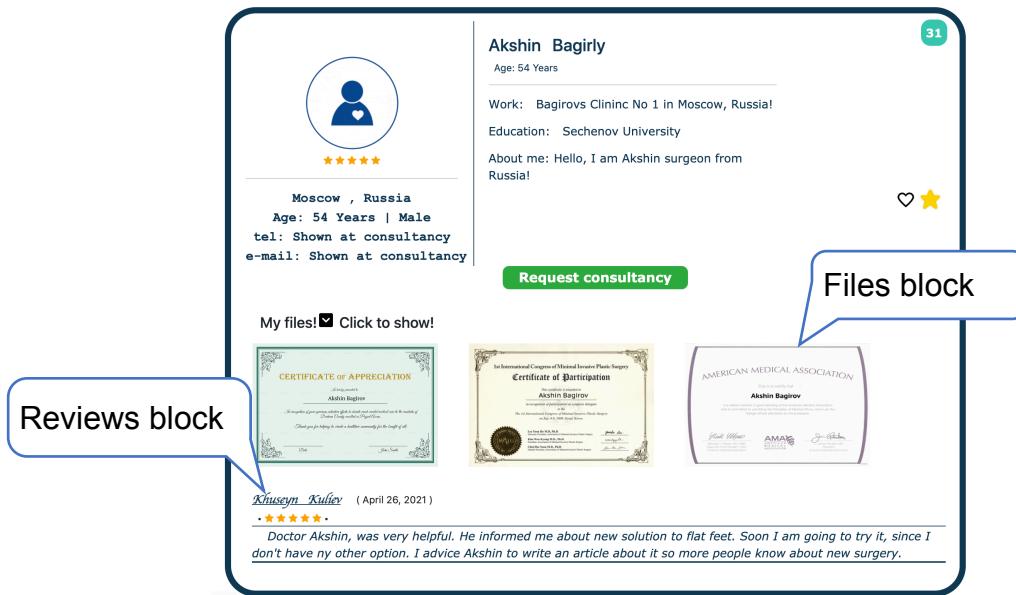


Figure 15: Screenshot of doctors full card page

Anamnesis

In medicine anamnesis is representing a set of information giving some background understanding of a patient(Grüne, 2015a). According to interviewees doctors anamnesis often helps to make right prediction of possible causes of illnesses. Usually anamnesis is formed by doctors interviewing patients(Grüne, 2015b). For this platform patients themselves will complete

fields. Each field has a placeholder representing standard question used to be asked by doctors. For this platform we have asked one of the doctors to help us to form set of questions to be asked for anamnesis.

Each patient has an anamnesis assigned to them at sign up. Anamnesis fields are initially empty. As shown in Figure 16 patients can enter data into fields and save each field independently. Upon clicking different save button same method is triggered that uses button's ID to identify on the back-end field to be updated and updates fields value.

Designing separate save buttons for each field allows to minimise volume of data sent to the server upon change made to anamnesis a field. Since I do not send every field of data to the server. However having only one save button for the entire page would mean that even if only one field was changed, each field's data would be sent to the server even without any updates.

MEDApp Home Articles LogOut Profile Hello, Khuseyn!

Khuseyn's anamnesis

Hello,

- Here you can fill fields of your Anamnesis!
- These are optional, you may leave them incomplete.
- This is for your doctor to view in case of consultancy.
- It will be available to doctor while you have active consultancy with that doctor!

<u>Place of Birth</u>	Where were you born?	Save
<u>Parents Age at birth</u>	At what age were your parents when you were born?	Save
<u>Description Of Birth</u>	Here we expect description of your birth...	Save
<u>About EAT</u>	Do you usually eat home made food or prefer eating outside?	Save
<u>Premorbid State</u>	You pre-illness conditions!	Save
<u>Peculiarities In Development</u>	Did you have any peculiarities in development?	Save

Instruction explaining user purpose of page

Each fields placeholder is an example question doctors ask patients in order to form Anamnesis of patient. But in this case patients themselves form anamnesis.

Figure 16: Screenshot of anamnesis page.

Registry

One of the key features of MedApp platforms is a registry that allows patients to keep a record of illnesses and treatments patients has experienced. Patients

are expected to add illnesses and treatments themselves. Figure 17 displays an example of a registry page.

The screenshot shows a web application interface titled 'Khuseyn's Registry'. At the top, there is a navigation bar with links for 'MEDApp', 'Home', 'Articles', 'LogOut', 'Profile', and a greeting 'Hello, Khuseyn!'. Below the navigation bar, the main content area is titled 'Khuseyn's Registry' with buttons for 'Add Illness' and 'Add Treatment'. A callout box labeled 'Instructions displayed upon button click' points to the 'Add Illness' button. The main content is divided into two sections: 'Illnesses' and 'Treatments'. The 'Illnesses' section contains a card for 'Immunologist, Covid-21' from April 1, 2021, with a status of 'Active'. The card includes a text box with symptoms ('I had cough and headache 😊 . Tried to cure myself with Paracetamol, but it didn't work out.'), a small image of a medical document, and a delete button. The 'Treatments' section contains a card for 'Immunologist, Antivirus medicine' from April 2, 2021, with a status of 'Solved'. The card includes a text box with a message about taking antivirus medicine to cure COVID-19, a small image of a medical document, and buttons for 'Edit Treatment' and 'Delete Treatment'. At the bottom of both sections are buttons for 'Select files', 'No file selected', and 'Add files'.

Figure 17: Screenshot of registry page

Upon clicking „Add Illness“, „Add Treatment“, Edit Illness, „Edit Treatment“ relative modal block is displayed to user. Example are below in Figure 18 and Figure 19.

The screenshot shows two side-by-side modals. The left modal is titled 'Add Illness' and the right one is titled 'Add Treatment'. Both modals have a close button in the top right corner. The 'Add Illness' modal contains fields for 'Select specialization of Illness' (dropdown), 'Diagnose' (text input with 'flu'), 'Description' (text input with 'I had a cough'), 'When was illness noticed first?' (date input with placeholder 'дд.мм.гггг'), and two status buttons: 'Active' (red) and 'Solved' (green). At the bottom are buttons for 'Select files', 'No file selected', 'Close', and 'Send'. The 'Add Treatment' modal contains fields for 'Select specialization of Treatment' (dropdown), 'Methodology' (text input with 'New calex technology'), 'Description' (text input with 'Surgery put in my leg...'), 'When the treatment took place?' (date input with placeholder 'дд.мм.гггг'), and three status buttons: 'Solved' (green), 'In process' (yellow), and 'Didn't work' (red). At the bottom are buttons for 'Select files', 'No file selected', 'Close', and 'Send'.

Figure 18: Screenshot of modal appearing to add illness/treatment in registry.

The image shows two side-by-side modal windows. The left window is titled 'Edit Illness' and the right window is titled 'Edit Treatment'. Both windows have a blue header bar with the title and a close button ('X'). Below the header are form fields.

Edit Illness:

- Select specialization of Illness**: A dropdown menu.
- Diagnose**: An input field containing 'Covid-21'. Below it is a small note: 'We'll never share your data without your permission.'
- Description**: An input field containing 'I had cough and headache 😷. Tried to cure myself with Paracetamol.' Below it is a small note: 'We'll never share your data without your permission.'
- When was illness noticed first?**: An input field containing '01.04.2021' with a calendar icon.
- Status**: Buttons for 'Active' (red), 'Solved' (green), and 'In process' (yellow).
- Buttons**: 'Close' (grey) and 'Send' (blue).

Edit Treatment:

- Select specialization of Treatment**: A dropdown menu.
- Methodology**: An input field containing 'Antivirus medicine'. Below it is a small note: 'We'll never share your data without your permission.'
- Description**: An input field containing 'I take antivirus medicine to cure myself from COVID19. Fortunately, it worked.' Below it is a small note: 'We'll never share your data without your permission.'
- When the treatment took place?**: An input field containing '02.04.2021' with a calendar icon.
- Status**: Buttons for 'Solved' (green), 'In process' (yellow), and 'Didn't work' (red).
- Buttons**: 'Close' (grey) and 'Send' (blue).

Figure 19: Screenshot of modal appearing to edit illness/treatment in registry.

File system

In MedApp's class diagram Appendix E we can see that in each class where we need files to be attached we have a relative class that is used to store files in the system. When users upload files these are stored on server side storage but database entries store the URL to file rather than the file itself.

5. Testing

In current chapter we test our software to ensure that MedApp functionality works as expected.

5.1.Core features unit testing.

Unit testing is the testing of components. In the case of software, a component can be a single function or set of functions. For my web application, I have test core features on various behaviours. Test cases, expected results and test results are displayed in Appendix F.

Application components have passed all conducted tests therefore can be considered ready for user acceptance testing to ensure that application as a whole works as expected and supports user actions.

5.2.Browser Compatibility Test

Web applications main advantage is their ability to be used through various devices without a link to the operating system. However, sometimes web application's front-end technologies like JavaScript do not work properly on every browser due to browser-specific settings, software, etc. That was one of the reasons we regularly made use of the jQuery library. We have tested Safari, Chrome, Mozilla Firefox, Microsoft Edge.

Web application successfully provides whole functionality on every mentioned browser. No browser-related failures were identified.

5.3.User Acceptance Test

The purpose of the user acceptance test(UAT) is to check whether developed software corresponds to the needs of users. At this stage, we have to ensure that users can make use of the core system feature implemented without any errors or software breaks.

In UAT we asked pairs of patient and doctor to interact through MedApp application to ensure that system can provide core features as expected. To test core feature we have given them 2 scenarios they had to follow, but without detailed step by step actions.

Scenario 1 - Consultancy:

Doctor and Patient signup and authenticate in the application.

Patient search doctor on the website and select specialist account corresponding to our test doctor. A doctor has to accept a request and both

users should interact through the consultancy page. Interaction includes the use of chat, file exchange and completing fields „Patient Message“, „Doctors Comments“, „Doctors Conclusion“. Upon the end of consultancy, users have to close it and leave a review about each other.

Scenario 2 - Articles:

Doctor and Patient authenticate in the application.

The doctor publishes an article and approves other articles. Doctor and patient browse articles, they read the full version of our test doctor's published article and leave a review, responding to each other's comment or deleting them.

4 pairs of users have followed the two scenarios described above. During test use of the system, no errors were identified. Users stated that they were able to use the system without any interruption. Therefore we can consider that UAT was successful and has been passed.

6.Evaluation

In this chapter, we asked users to evaluate MedApp system from various aspects and evaluate its strengths and weaknesses.

6.1.User evaluation

After developing the system we have asked potential users to assess system's core feature through completing a questionnaire present in Appendix G. Figures 27, 28, 29 in Appendix H display questionnaire result where we asked potential users to asses how comfortable is doing various tasks one MedApp platform.

Results show that overall users are satisfied with the core features of application. Communication through chat has got the highest rates from both patients and doctors. That is crucial since effective communication between patient and doctors is core to MedApp system and concept.

Respondents find webpage design and navigation intuitive and simple. But results show that there are still some difficulties to understand the application design. The reasons might be a lack of instructions explaining various features.

According to respondents weakest feature of this system from the comfortability, the point is leaving and editing reviews on articles. The user finds that part quite confusing and difficult to use.

When it comes to doctor related core features, these are rated high, each point has got on average more than 8,5. While patient-oriented features were rated lower overall. The minimum value is 7,1 for keeping a record of illnesses and treatments. The reason is the current case could be lack of instructions, on the other hand, we also have to consider the fact that patient users just do not have experience of recording their illnesses and treatments since in clinics it is doctors who fulfil medical registry, not the patients. However, we have to notice that 7,1 is still a high rate, so the overall registry feature can be considered successful alongside other functionalities.

For further development, we are better to interview our respondents to receive some feedback on features that should be added or edited to consider some improvements.

6.2.Weaknesses and strengths

In current chapter we attempt to analyse developed system.

6.2.1.Strengths

The main strength of developed system is that it is an attempt to solve the problem of inappropriate medical services being provided from both sides:

patient and doctor. The application allows the patient to freely communicate with various doctors as well as allows doctors to share their knowledge and experience with their colleagues.

The registry is one more strong point of MedApp platform that can be seen as a diary for the patient to keep a record of illnesses and treatments as well as become that supportive material helping the doctor to understand the patient's problem better. It is a strength not only because existing applications do not have the analogy of MedApp registry feature but also because doctors we interviewed find it very important. Same as with registry is anamnesis that is not provided by competitive software although according to interviewed doctors is very helpful for providing medical services.

The next strength is the profile page. Often we can see profile pages that are used only for account setting and personal detail management, like changing contact details or full name. In MedApp user can access and control any aspect from a profile page. Profile page gives patient and doctor control over requests, consultancies, articles and personal detail. The patient may also access their registry and anamnesis, while doctors may access article editor, approve articles, and publish their ones. The only thing users cannot do through the profile page is to browse a doctor or article.

One more important aspect is the articles section that is an important addition to the entire platform. Implemented system facilitates experience, knowledge share that is crucial to improve overall doctors literacy and therefore improve medical services provided. Articles role goes even beyond the knowledge share, it also acts as content that will hold patients inside the platform as well, since they can also learn a lot of new stuff from articles. We have articles review feature. Unless doctor review, article reviews can be edited or deleted, users can respond to each other's reviews and participate in discussions. Moreover, articles are expected to be published and approved by MedApp platform doctors that makes MedApp unique.

An important point should be made about the design of web pages overall. Each page has a window size responsive design layout. Therefore on various devices like mobile phones, tablets, laptops, PC website pages stay clear.

From a security perspective, application provides various security feature is supported by the Django web framework. These are hashing and salting password on the database, use of ORM to deal with a database that prevents SQL Injection attacks. Also, Django provides CSRF token issue feature, that allows to assign unique tokens to each client and prevent cross-site request forgery attack.

From a performance perspective, application makes use of Asynchronous JavaScript and XML(AJAX) where it is possible to avoid unnecessary page refreshes. By this, we minimise the volume of data exchanged between client-server and makes web application to respond user interaction quicker. One

more performance-related aspect of MedApp is putting JavaScript and CSS of the application separate from HTML template documents. This enables the browser to cache JS and CSS files and reuse them. Therefore server would not be required to send JS, CSS data every time a page requested and make a quicker response.

6.2.2. Weaknesses

Besides strengths system also has some weaknesses.

The main weakness of the application is the lack of instructions to support user learning of new system features. Although the system has some instruction, still many features should be explained concisely. One of the effective ways to solve this problem might be using a small pop up message displayed near a button or field upon mouse over element explaining the feature and how it works. An early example can be seen in the articles comment section, where a message displayed when the user hovers over the „Save“ button on editing the comment, shown in Figure 20.

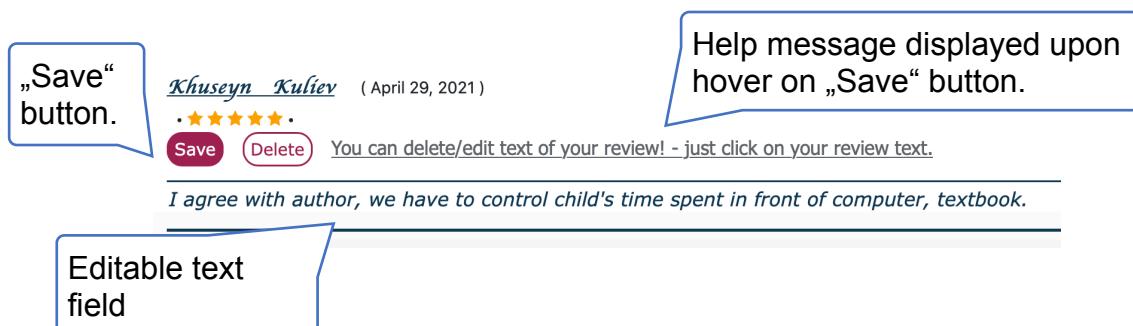


Figure 20: Example of help message on comments section.

Another weak point is that search output for both doctors and articles does not provide rich options for filtering, sorting out, etc. Although on start it is not critical, for further development that should be implemented.

One more weakness that may strongly improve user experience is the lack of chat features. Although we have fully working live chat, we lack supportive features like an indicator of a user being online, an indicator of a message being read by another user, the ability to delete messages, etc.

Another weak side of MedApp is that application lacks some secondary feature that was not implemented in the first version but still can positively impact on user experience. An example of such feature is doctor's achievements system, that was mentioned by interviewee doctors as an important addition to motivate doctor.

From a security perspective, MedApp's does not require a user to create difficult password including A-Z, a-z, 0-9, +-# and other characters that would make guessing password or exhaustive search even more time consuming.

Overall these weaknesses are a question of feature implementation, not fundamental concept weaknesses. Therefore these weaknesses can be simply solved without intervention into the fundamentals of the platform.

6.3.Legal aspects

Due to the nature of MedApp application, in order to make use of application users have to share some personal data with system. According to European Commissions there are 7 principles of GDPR(General Data Protection Regulation) that must be followed when processing users personal data(European Commission, 2019). MedApp platform attempts to follow these principles.

MedApp:

1. Processes data in transparent manner.
2. Collects and processes only personal data that is necessary to provide platform functionality.
3. Doesn't hold user's personal data to third party.
4. Processes personal data only for original purpose.
5. Attempt to prevent unauthorised access to users personal data stored on server-side.
6. Gives users control over their personal data that can be easily removed from the system.

7. Conclusion

At this chapter I summarise project achievements and describe challenges faced. Moreover, I identify further development phases and conclude the project.

7.1. Project achievements

Overall we have achieved set objectives and project aim.

We have developed a system supporting online medical consultancy as well as facilitating knowledge and experience-sharing through the articles section. The developed product is solving the problem of inappropriate or ineffective treatment methodology. We provide the patient with an opportunity to reach any doctor distantly and learn other specialists opinion about their problem or the proposed solution. We allow doctors to publish articles and share knowledge.

Web application acts more like a platform where users can spend their time rather than just visit for one-time service. Both patient and doctor are expected to participate in consultancy, but if there is no consultancy they can spend time learning new things from articles, discussing them and contributing to the overall growth of public literacy about medicine.

On the other hand, there are issues we couldn't implement although they are considered important. An example of such is the achievements system that interviewees mentioned as an important addition to a doctor's account. Also, I couldn't implement search output sorting by rating or any other characteristic.

Although the project was challenging it helped me to improve myself strongly. First of all, I have strongly improved my web development skills, which includes dealing with frontend and backend issues. During the project, I improved my analytical thinking especially during the designing of the system architecture. I have improved my ability to manage time properly in order not to be late with my work. However, due to the illness stated in chapter 7.2, I missed my time plan slightly. In future, I am better to consider possible delays due to external factors. Most important is that at the end of a project I have improved my ability to learn and adopt new things quickly that allows me to develop in any new area effetely.

7.2. Difficulties and challenges

The first major difficulty was the lack of web development experience. Although I have studied web-development module in my third year still I didn't have much programming experience on Python and with Django. During the initial phases of a project where I conducted interviews and background research, I was learning the fundamentals of Python and Django and practising these

technologies. During the development process, I still was learning many new things on go, since many issues were arising during the development.

The second major difficulty was organised interviews with doctors. Although I know many doctors still it was difficult to arrange a time for interviews and post-development user evaluation with them. Every doctor I requested to participate in was happy to help but only with 4, we could allocate a time due to the specialist's very tight schedule. These 4 doctors were also participating later in the user acceptance test.

Another challenge is related to my health that was monied in risk assessment(Appendix A). From the start of April, I was infected by COVID-19. Although I did not have any symptoms later I found out that my vision started to drop immediately and it was caused by inflammation that COVID-19 indirectly triggered. The problem aggravates with fact that even glasses or lenses couldn't work properly to help me. During that problem I was continuing to work on the project however I had to slow down significantly. Simultaneously I was treating my eyes. Unfortunately, I am still treating my eyes and at the moment of project submission still cannot see properly even from a close distance.

7.3.Further development

Although the developed platform achieves set aims and contributes to the solution of the stated problem still there are improvements that need to be made. Further development can be set into phases.

Phase 1: Based on user-provided feedback some improvement and secondary features would be implemented into an existing application. Some of the features to be developed are an indicator of a user being online, a doctor achievements system, implementation of pop up instructions explaining a feature to the user, language selection. At the end of Phase 1 application will be deployed to an online server to allow a wider range of users to participate in the system and provide feedback.

Phase 2: Implementation of Moderator actors features. Moderator is expected to play the role of third party dealing with user disputes, moderating articles, reviews, etc. We expect the development of a profile page from where the moderator can execute his/her responsibilities.

Phase 3: Major update of web application design aiming to make it unified and more intuitive. Moreover, major feature updates should be conducted as well to improve software effectiveness. The phase will start from analysing existing features and identification of improvements to be done.

Each phase would end with a user feedback session to identify further improvements to be made.

7.4. Overall conclusion

Overall a substantial product was developed in response to the stated problem. I have succeeded in building a deliverable that addresses both patient's and doctor's side of the problem and leaves an open perspective for further developments.

Proper user investigation was made prior to development through interviews conducted along side with analysis of existing software aiming to deal with similar problem but with slightly different approach.

Project has become a great way to apply my knowledge as well as develop new skills and get into new technologies. The project is a start of real-life application that will be implemented further and made available for real life interaction.

References

- Doherty, E. (2020). *MVC Architecture in 5 minutes: a tutorial for beginners*. [online] Eduative: Interactive Courses for Software Developers. Available at: <https://www.educative.io/blog/mvc-tutorial> [Accessed 4 May 2021].
- Dragos-Paul, P. & Altar, A., 2014. Designing an MVC Model for Rapid Web Application Development. *Procedia Engineering*, 69(2014), pp. 1172-1179.
- European Commission (2019). *What data can we process and under which conditions?* [online] European Commission. Available at: https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/principles-gdpr/what-data-can-we-process-and-under-which-conditions_en [Accessed 4 May 2021].
- Grüne, S. (2015). Anamnese und körperliche Untersuchung. *DMW - Deutsche Medizinische Wochenschrift*, 141(01), pp.24–27.
- Hursch, W. and Lopes, C.V. (1995). *Separation of Concerns*. [online] CiteSeerX. Available at: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.5223&rep=rep1&type=pdf> [Accessed 4 May 2021].
- Miller, J. (2007). *The Don't Repeat Yourself Principle and the Wormhole Anti-Pattern | Jeremy D. Miller*. [online] Codebetter. Available at: <http://codebetter.com/jeremymiller/2007/03/22/the-dont-repeat-yourself-principle-and-the-wormhole-anti-pattern/> [Accessed 4 May 2021].
- Stevens, W.P., Myers, G.J. and Constantine, L.L. (1974). Structured design. *IBM Systems Journal*, 13(2), pp.115–139.
- WHO (2011). *Opportunities and developments TELEMEDICINE in Member States Report on the second global survey on eHealth Global Observatory for eHealth series -Volume 2 2010*. [online] WHO, p.8,9. Available at: https://www.who.int/goe/publications/goe_teemedicine_2010.pdf [Accessed 22 Oct. 2020].
- WHO (2016). Report of the third global survey on eHealth Global Observatory for eHealth Global diffusion of eHealth: Making universal health coverage achievable. [online] WHP, pp.57–60. Available at: <https://apps.who.int/iris/bitstream/handle/10665/252529/9789241511780-eng.pdf?sequence=1> [Accessed 23 Oct. 2020].
- Wysocki, R.K. (2009). Effective Project Management: Traditional, Agile, Extreme. John Wiley & Sons Incorporated, p.300.

Appendix A - Risk assessment

Risk assessment

Description of risk	Impact of Risk	Likelihood rating	Impact rating	Preventive actions
Poor time management	Development of prototype might not be finished in time	Medium	High	Regularly revise timetable and ensure following it
Loss of work done	May lead to loosing all the files related to current project	Low	High	Regularly save all files backup versions
Connection error during test use	Website may work inappropriately because of low internet connection of user evaluating website	Medium	Medium	Ensure that evaluating users will have stable connection
Personal health	Due to risks of getting infected project development may be completely stopped	Medium	High	Avoid visiting popular public places and use personal safety masks and gloves
Cancellation of project presentation	Due to COVID-19 there is a risk of events related to project presentation or evaluation to be cancelled at all	Low	High	This risk is totally out of my control, best thing is to have ready project.

Description of risk	Impact of Risk	Likelihood rating	Impact rating	Preventive actions
Lack of testing	We may miss testing some critical issues in a software	Medium	Medium	Regularly check list of things should be tested.
Security problems	An error in software may lead to lack of security in a system	Medium	High	Test security functions to work appropriately
Website speed problems	Website may work slowly due to high volume of data being loaded by browser	Low	Medium	Ensure browser download pages of website fast enough. Otherwise improve code.
Browser compatibility	Some browsers may not support full functionality of my system	Medium	High	Test system on different browsers

Appendix B: Interview structure and questions

Introduction: Explaining to interviewee what kind of system is expected to be developed.

Part 1

Questions to doctors

1. What do you think of this kind of system to be developed?
 - Will it be demanded and why?
 - What is necessary for this kind of system?
2. Will a regular doctor have enough time for this kind of work to be done?
3. What will motivate doctors to be a part of this system?
4. Can you propose a feedback mechanism that will allow people to select best doctor out of alls based on some kind of reviews leaved?
5. Is there a mechanism in medical area to ensure that a person has been graduated and received medical qualification to ensure he/she is not a fake doctor?
6. Can you as a doctor state that it is possible to provide some initial consultancy based on documents that a person can remotely send you? So is it possible for doctor to provide consultancy to someone online without physical meeting for at least basic consultancy purposes?

Questions to patients

1. Have you ever faced problem related to lack of access to information about an illness, treatment?
2. Have you ever faced problem related to lack of knowledge of a doctor related to your problem?
3. What do you think of this kind of system to be developed?
 - Will it be demanded and why?
 - What is necessary for this kind of system?
4. Will you as a patient mind to share your problem with doctor online?
5. What might make you as patient to trust to a doctor?
6. If the system will require you to pay a specific amount to receive consultancy how likely you will pay for it?
7. Did you have some treatment process or any medical service provided that could be done remotely?
8. Will it be important for you to get an opportunity to contact foreign doctors as well?

Part 2

Questions for each interviewee

Explain two challenges being addressed:

- For patients it is lack of opportunity to reach different doctors independently from patients location.
- For doctors it is to lack of access to experience and knowledge of other specialists.

1. Should it serve only one purpose or it will make sense to create something bigger?
2. What are most essential features do you think this system should have?

Conclusion, finalising interview

Appendix C - Existing systems analysis structure

1. How focused is a website on providing online consultancy?
 - Based on information on main page and easy access to the online consultancy section
 - Based on set of services provided
2. How strong does it engage direct communication between patients and doctors?
3. Type of consultancy implementation: online chat, video call, other
4. Profiles
5. Review system
6. Interesting/unique features
7. Select doctor Use case steps
8. Take aways(Conclusion)

Appendix D: Existing systems home page screenshots

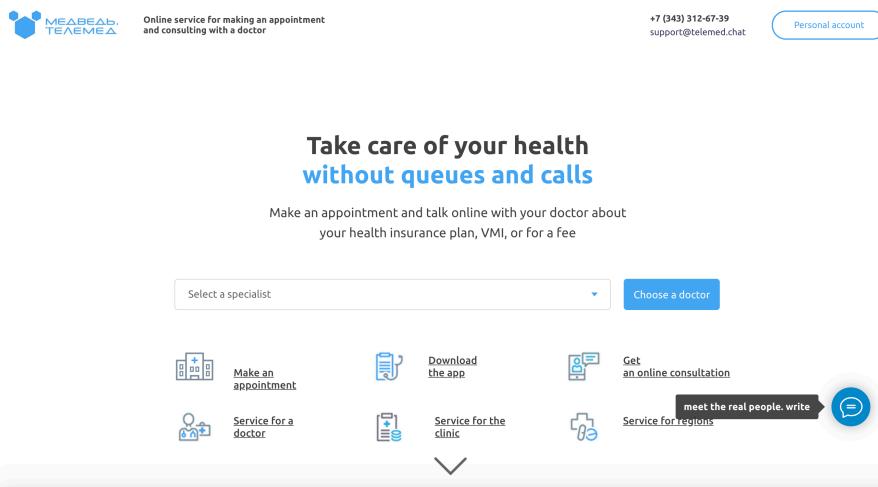


Figure 21: Medved web service's home page screenshot

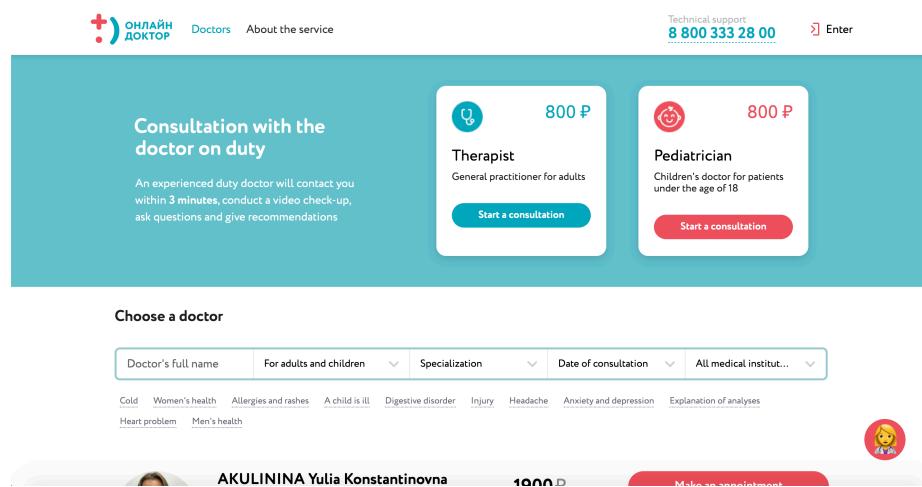


Figure 22: OnlineDoctor web service's home page screenshot

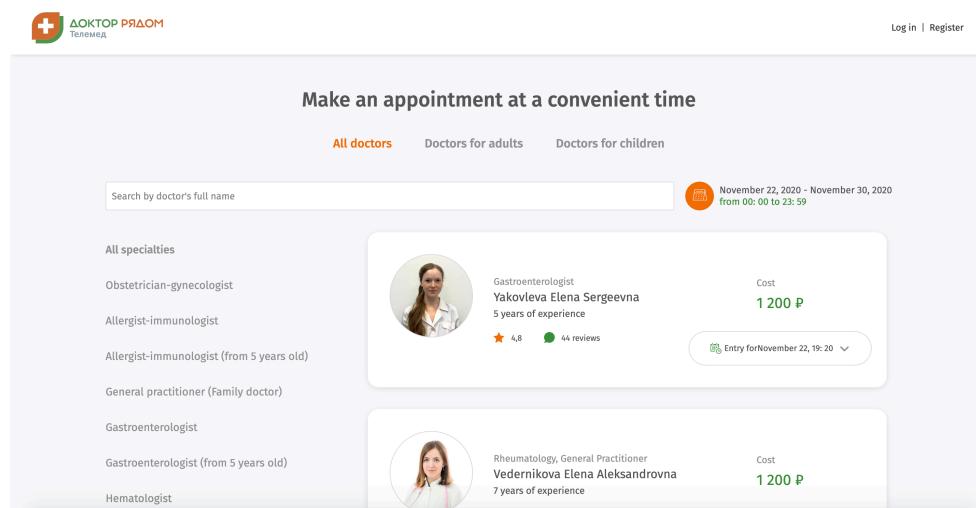


Figure 23: Dr.Clinic web service's home page screenshot

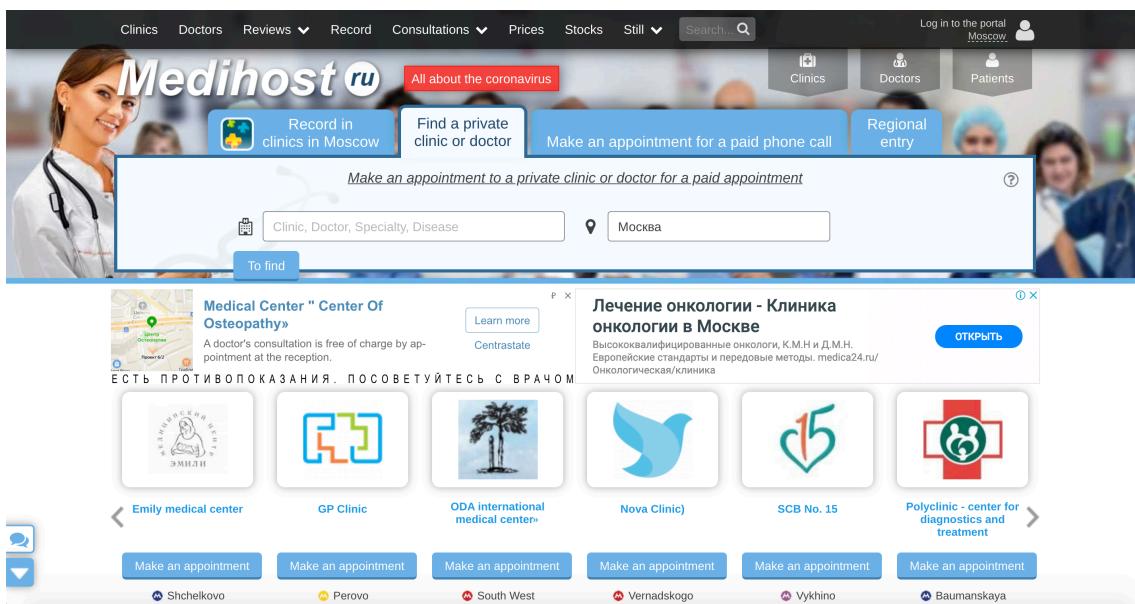


Figure 24: Medihost web service's home page screenshot

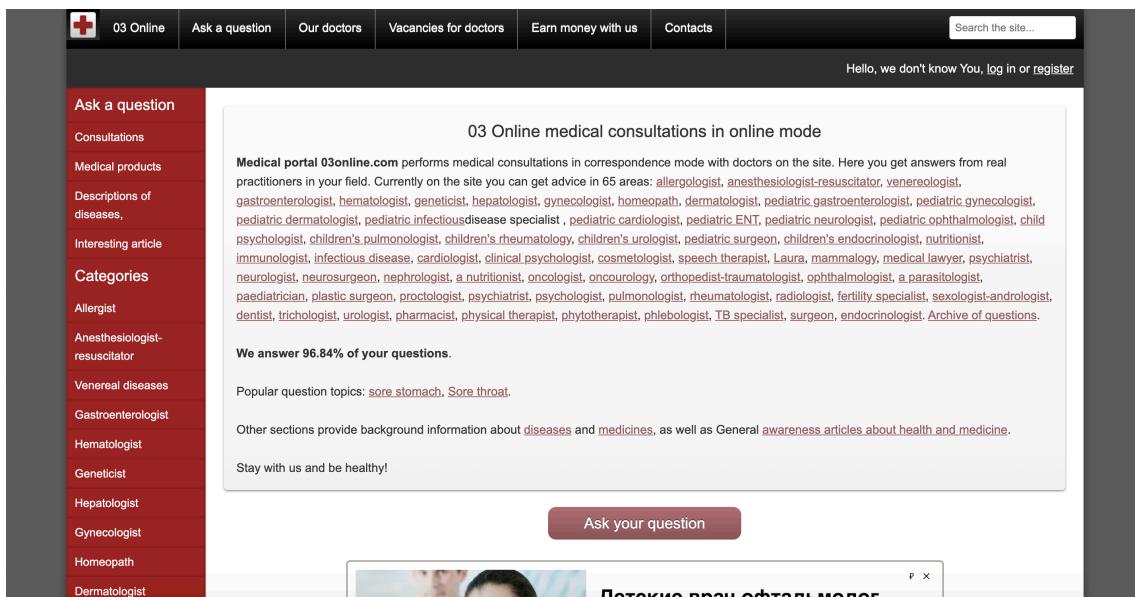


Figure 25: 03Online service's home page screenshot

Appendix E: UML Class diagram

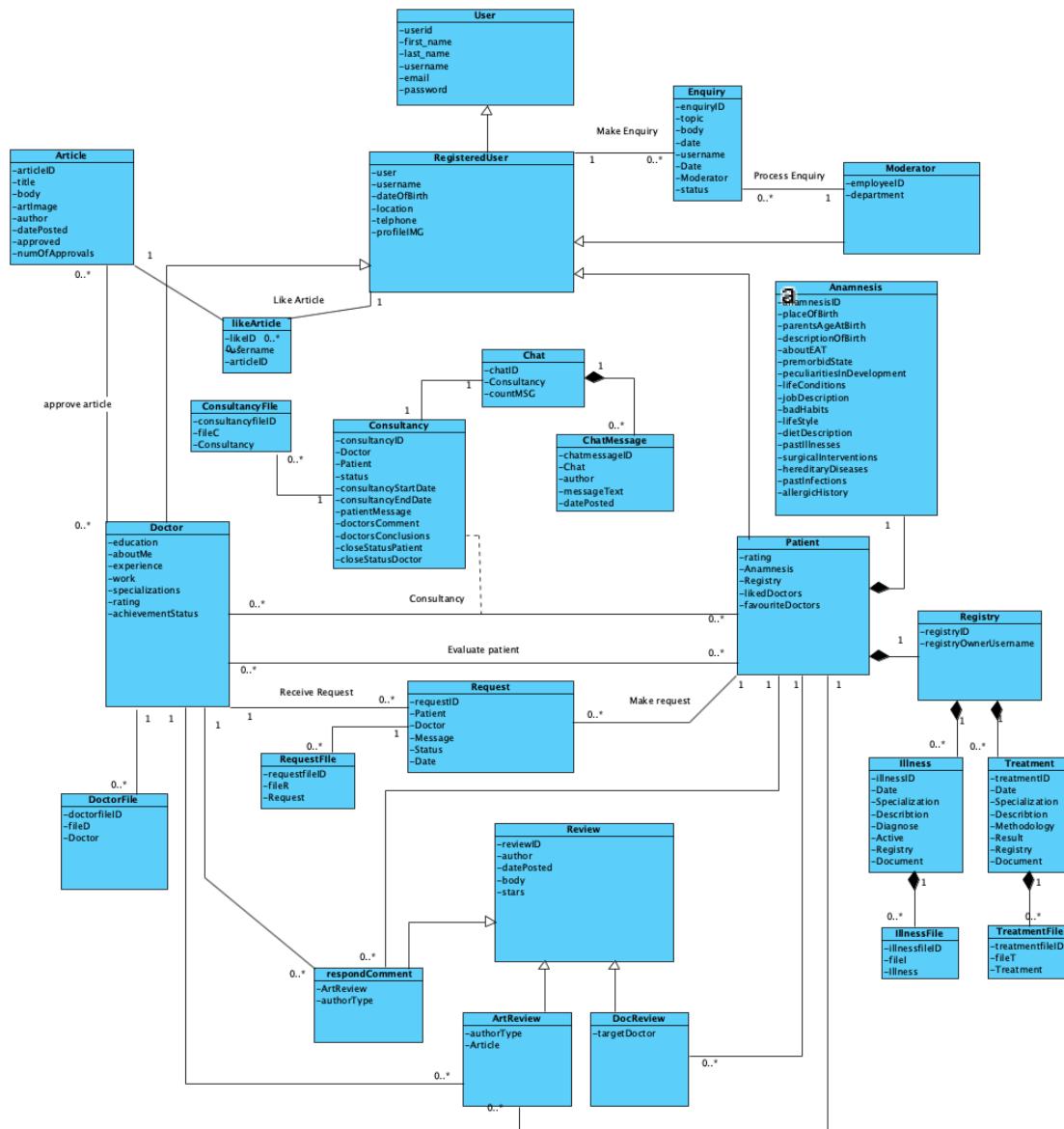


Figure 26: MedApp UML Class diagram

Appendix F - Summary of unit testing

Component	Test case	Expected output	Success
Login	User inputs correct login credentials	User is taken to homepage	✓
	User inputs incorrect login credentials	An error message displayed on login page	✓
Sign up	User inputs unacceptable value in a signup fields	Relevant error message displayed and submit button disabled	✓
	User select to create doctor account	Extra fields for sign up are displayed	✓
	User submits sign up form successfully.	User is taken to home page.	✓
Edit profile	User edits profile data.	Related save button indicates successful edit.	✓
	User uploads new profile image and saves it.	User is taken to profile page and new image displayed on profile.	✓
Add profile documents	Doctor uploads documents through profile page.	Uploaded documents displayed on profile page and doctors full card.	✓
Registry of illnesses and treatments	Patient creates new illness/treatment through registry page	New illness/treatment displayed on registry page	✓
	Patient edits illness/treatment through registry page	Updated illness/treatment displayed on registry page	✓
	Patient attached document to illness/treatment	Illness/treatment card displayed with new documents	✓
	Patient deletes illness/treatment through registry page	Registry page displayed without deleted illness/treatment	✓
Complete anamnesis	Patient inputs new value on anamnesis page and saves it.	Related save button indicates successful input save.	✓

Search doctor	Patient inputs specialisation and name into search fields upon search request.	Doctors with matching name and specialisation are displayed	
	Patient leaves specialisation and name fields empty upon search request.	Each registered in the system doctor is displayed.	
Send request	Patient send request consultancy with message to doctor.	Successful request is indicated to user and request entries are created at patient's and related doctor's profile page. Related email notification send to target doctor.	
	Patient attaches documents to request consultancy.	Profile page displayed to user and new documents are shown on request modal.	
Doctor respond on request	Doctor accepts request consultancy through profile page and request modal.	Request status set to „Agreed“ and consultancy entry in involved users profile page is created with status „Active“ and data transmitted from request. Related email notification sent to patient.	
	Doctor reject request consultancy through profile page and request modal.	Request status set to „Rejected“. Related email notification sent to patient.	
Consultancy	User clicks consultancy entry on profile page.	Consultancy page shown with related data.	
Chatting	User sends a message through chat block on consultancy page	New message displayed in chat block and other user receives message within 1 second.	
Attach documents to consultancy	User uploads document through consultancy page.	New document displayed on consultancy page.	

Access patient registry/anamnesis	Doctor accesses patient's registry/anamnesis through link on active consultancy page	Relevant registry/anamnesis page displayed in „read“ mode without editing features.	
Close consultancy	One of the sides closes consultancy on his/her side through consultancy page.	Consultancy status is set to „Awaiting close“. Relevant close message displayed on consultancy page. Buttons „End consultancy“, „Dispute“ and upload document elements are hidden. Relevant email notification sent to patient and doctor. Review modal displayed.	
	Another side closes consultancy through consultancy page.	Consultancy status is set to „Close“. Relevant close message displayed on consultancy page. Buttons „End consultancy“, „Dispute“ and upload document elements are hidden. Consultancy end date is displayed on consultancy page. Relevant email notification sent to patient and doctor. Review modal displayed.	
Send review about user	User leaves review about another user upon closing consultancy.	Reviewed users rating is updated. Doctors review is published on doctors full card.	
Search article	User inputs search query into search article field.	Articles with title or text matching search query are displayed.	
View full article	User clicks „Read Article“ button on search article result page.	Page displayed with relevant article's details and reviews.	
Publish article	Doctor completes article fields and clicks publish button.	Article added to the list of articles awaiting approvals.	

Confirm article	Doctor clicks on article entry awaiting approval and clicks approve.	Article approval count incremented. Approved articles entry is coloured green in „Awaiting Approval“ list for doctor who approved that article.	
Review article	Authenticated user first time leaves review about specific article through a displayed modal.	Successful review sent message displayed. Full article page displayed with new review added. Article rating modified.	
	Authenticated user second time leaves review about specific article while one review is already published from this user.	Warning message displayed. Review not sent.	
Comment article review	Authenticated user responds on article review through displayed input field.	Successful comment sent confirmation message displayed to user. Comment displayed on full article page.	
	Authenticated user that is author of review/comment article, changes its content and clicks save.	Successful modification confirmation message displayed to user. Review/comment displayed with new content.	
	Authenticated user that is author of review/comment article, clicks „Delete“ button.	Review/comment removed from webpage and database.	

Table 3: Summary of unit test

Appendix G - User evaluation questionnaire

1. Overall questions

1. How intuitive is webpage design overall? Rate from 1 - 10
2. How simple is navigation across different section of application? Rate from 1 - 10
3. How closely arch article result matches your search query? Rate 1-10
4. How comfortable is leaving and editing reviews on articles? Rate from 1 - 10

2. Patient related questions

1. How closely search doctor output matches your search query? Rate from 1 - 10
2. How informative is doctors card at search result? Rate from 1 - 10
3. How simple is it to send consultancy request? Rate from 1 - 10
4. How comfortable is communication with doctor through consultancy page chat? Rate from 1-10
5. How comfortable is exchanging file on request or consultancy? Rate from 1-10
6. How comfortable is keeping track of illnesses and treatments in system? Rate from 1-10
7. How comfortable is managing your anamnesis? Rate from 1-10

3. Doctor related questions

1. How comfortable is management of consultancy requests though profile page? Rate from 1 - 10
2. How comfortable is providing consultancy through consultancy page? Rate from 1 - 10
3. How comfortable is communication with patient through consultancy page chat? Rate from 1 - 10
4. How comfortable is access to patient's registry and anamnesis through consultancy page? Rate from 1 - 10
5. How comfortable is publishing/edit articles? Rate from 1 - 10
6. How comfortable is approving articles? Rate from 1 - 10

Appendix H - Questionnaire results graphs

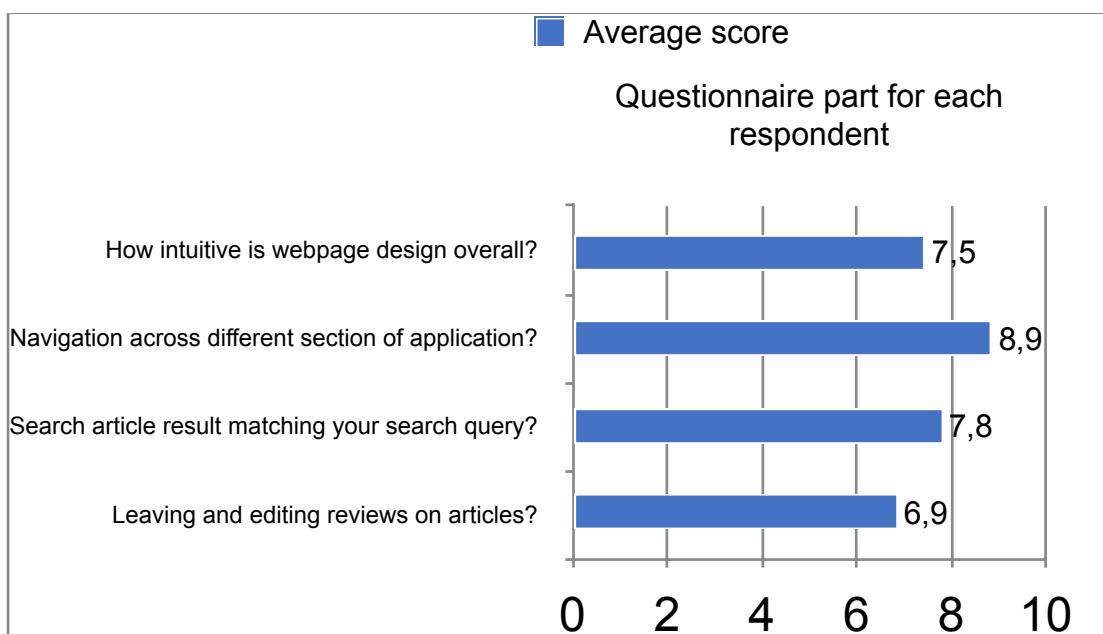


Figure 27: Questionnaire results for 20 potential users include both doctors and patients

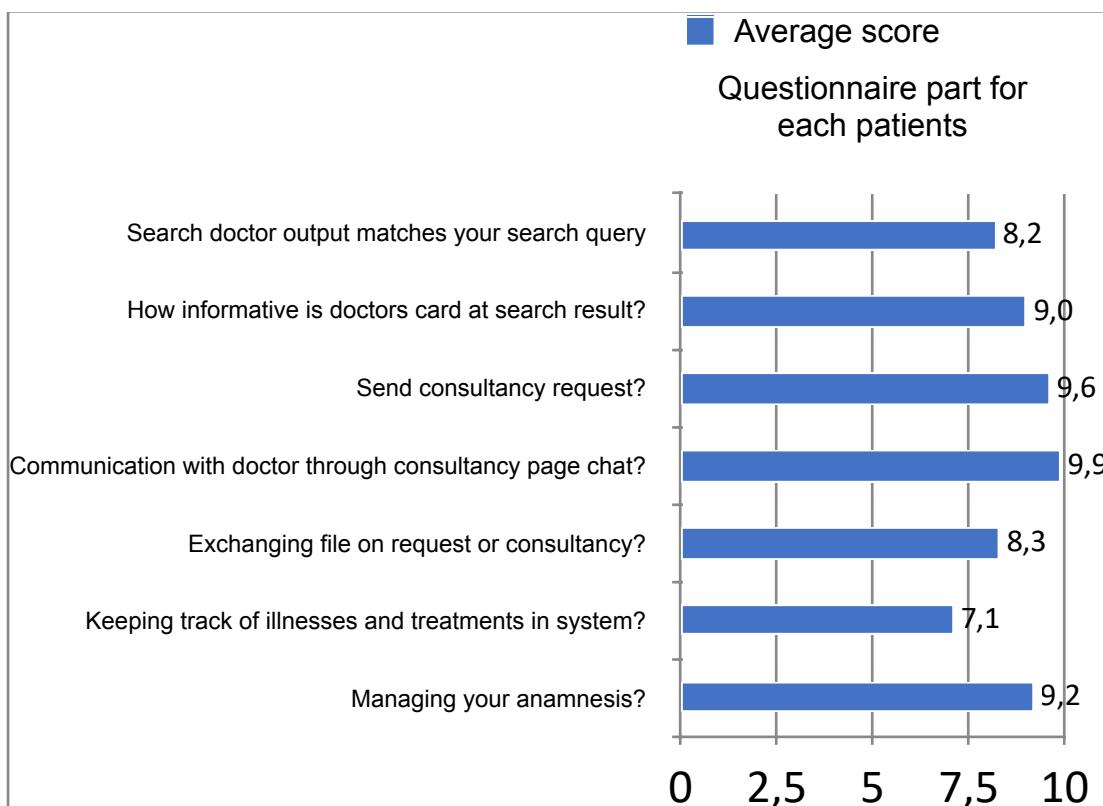


Figure 28: Questionnaire results for 10 potential user patients

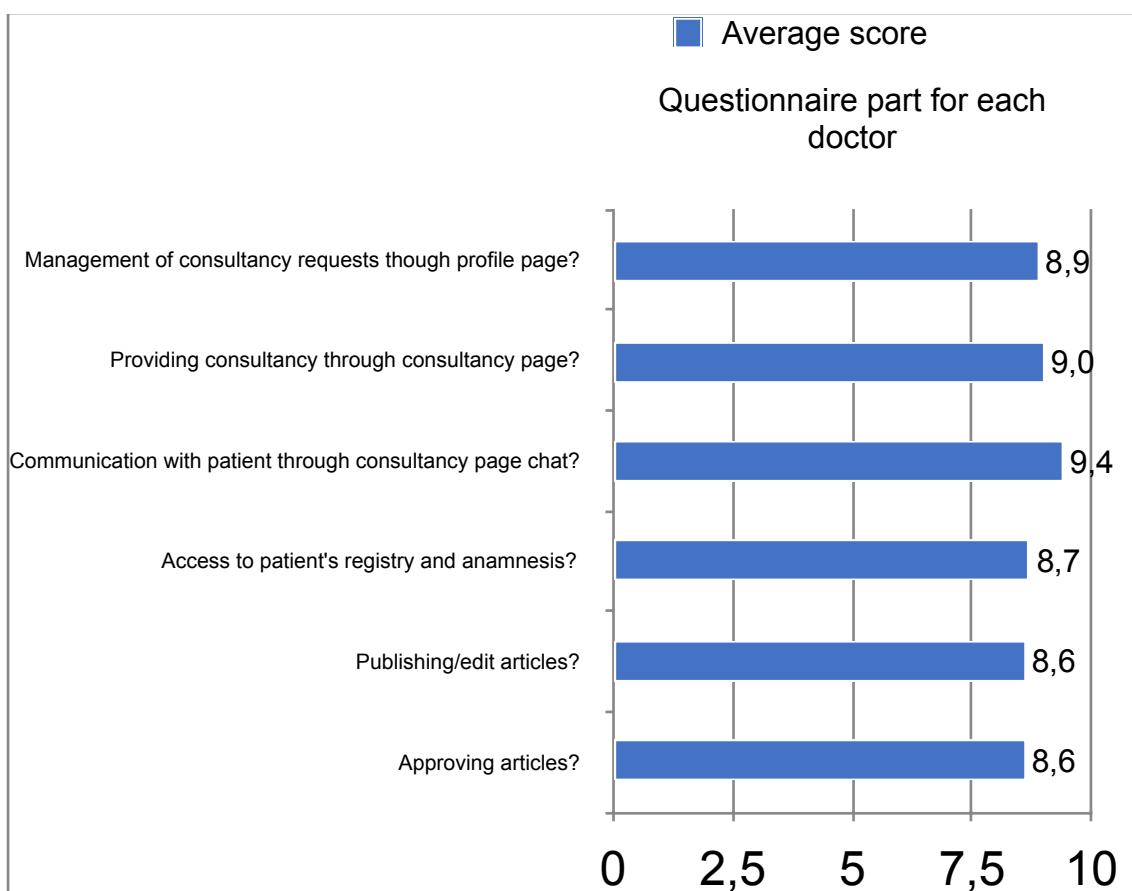


Figure 29: Questionnaire results for 10 potential doctor user