1 THEORETICAL PART

* 1. Introduction

Today, in the field of pharmaceutical and medical services, there is a lot of scientific and economic data for the general presentation of these areas. Since the project solves the problems of these areas, various solutions and analysis of already existing problems must be translated in order for the project to be able to meet the requirements of public and market demand. And also, the need to analyze already existing software solutions in order to obtain the necessary information about the existing order of things that have developed in the development of projects with a similar or similar topic. However, it is possible to create new solutions, which can increase the demand for the project and add some uniqueness to it over other solutions. Following these criteria, the development team compiled an analysis and review of all literature related to the project topic, as well as a comparison of existing software solutions.

* 1. Literature inspection of problem
     1. Introduction

There are many scientific works related to the field of pharmaceuticals and medicine. It contains a lot of data related to problems in society, as well as examples of how to solve these problems. Therefore, applying these scientific works, the development team will have an idea of various problems and the ability to add their own solutions and internalize these solutions to the project.

1.2.2 “ФАРСАР” review

First is a “ФАРСАР” study **[n]** in Russian cities. This study examines the problem and prevalence of self-medication among various age groups of the adult population in Russian cities and their attitude to self-medication. The study was conducted in the form of a survey of the population of cities.

Self-medication is spreading in modern society due to the dissemination of information via the Internet and the media. Self-medication in case of simple situations like headache, heartburn or heartburn may be warranted. But in the case of clinical symptoms and chronic diseases, it can be uncontrolled, inappropriate use of drugs and neglect or incorrect treatment can lead to major health problems. However, self-medication all plays an important role in health care because as knowledge increases in the population, it becomes a necessity in some cases.

In result, 3798 respondents were interviewed, among whom women predominated - 65.8%. 35.9% of respondents were under the age of 25, 37.9% - 25-55 years old and 26.2% - over 55 years old. Self-medication rates averaged 63.1%, and of these, 2/3 were partially or completely successful. OTC drugs use was 62.1%, non-steroidal anti-inflammatory drugs were 34.2% and herbal remedies were 24.9%. Gastrointestinal problems were common reasons for self-medication was tract, headache, preventive taking drugs, acute respiratory viral infections and pains of various localization. The majority of respondents studied information about drugs independently, using sources such as the Internet and the media, and the percentage of people who took into account the advice and recommendations of doctors is 0.8%.

This indicates that self-medication is common in the population of Russia and also possibly in the population of most CIS countries. Therefore, in order to lessen the harmful effects of self-medication in population, the project tries to provide the most truthful information that is available in the public.

1.2.3 “Analysis of Drug Name Confusion in Medicine Labels” review

Second is a study “Analysis of Drug Name Confusion in Medicine Labels” **[n]** done by Vincent Huff and Dan Nathan-Roberts from San Jose State University, California, USA.

Purpose of the research is to study the problem of labeling and names due to confusion and misunderstanding in some names and also the problems of the human errors that increase the misunderstanding as well as possible methods and solutions to this problem.

According to statistics by the Institute of Medical Health in 2000 reported that at minimum 44,000 to upwards of 98,000 Americans die each year in hospitals from medical errors each year. The widespread use of prescription drugs in the United States has grown from people using one or more drugs to using five or more drugs. In 2010 about 2.6 billion drugs prescribed in the US and it will only increase. In 2001 the Joint Commission on Accreditation of Healthcare Organizations estimated that 10,000 patients were suffered each year due to drug label name confusion.

This is one of the reason why it is necessary to increase the fight against human errors and pay special attention to this on the part of health care. For instance, if a person uses 2 or more drugs and one of them is not correctly prescribed, then this can lead to health complications or even death. Usually people do not pay attention to which prescribed drugs they are using. Because of this, many patients do not know how to handle situations when an error occurs. Healthcare professionals do not have appropriate protocols to prevent and inform patients about an error that can lead to death. There are many ways these errors appear. For example, it may be a misspelled drug name or the name was written in a very incomprehensible handwriting. Furthermore, different medicines may have the same name, which may lead to the fact that the patient, instead of taking painkillers, takes psychotropic drugs. The error mitigation must be a priority for healthcare. Reducing the number and impact of errors can save both industry money and patient health.

On December 5, 2014 a 65-year-old woman named Loretta Macpherson passed away due to the fact that she confused the names of the drugs. She underwent brain surgery and was prescribed the wrong medication by the hospital staff. Instead of Zarotin which is an anticonvulsant medication, she took Zemuron which is a paralyzing muscle relaxant, which led to the mixed up of drugs. This is a good example of how a small misspelling of a drug can be fatal. Because of this, the hospital suffered losses in money and her family members also lost her. This problem did not arise simply because one person made a mistake in choosing two drugs. The main reason is that the current hospital system allows this to happen.

As noted above, drug naming problems are a very dangerous thing, as they can lead to fatal outcomes. However, it is possible to mitigate this problem. From the research, the development team decided to pick up the idea that the name is very important not to be confused. Therefore, the project will rely on the accuracy of the spelling of the names of drugs and also enables the medical staff from hospitals to use our site as links to certain drugs in order to reduce the possible occurrence of errors in the spelling of drugs and, if possible, show additional information about the drug to the patient.

1.2.4 “Health information behavior of rare disease patients: seeking, finding and sharing health information” review

Third is a study “Health information behavior of rare disease patients: seeking, finding and sharing health information” **[n]** by Snjezana Stanarevic Katavic from Croatia, University of Osijek.

The aim of study is to describe distinctive aspects of health information behavior of rare disease patients and specific challenges they face when seeking health information. The methods of study is conducting semi-structured interviews among fifteen respondents that suffering from three different rare diseases.

The prevalence of information technology has made health information and drugs more accessible. However, there are groups of people with rare diseases who find it difficult to find information about their diseases and the drugs for them. Rare diseases are diseases where the number of patients does not exceed 5 patients out of 10,000 in Europe and 5% of patients out of 6,250 in the United States.

Lack of medical knowledge is a common problem. However, patients with rare diseases are more likely to seek and value non-medical advice used in everyday life. Patients need an independent search for medical information, as the information received from medical staff was often insufficient. Patients in search of the necessary information faced the difficulty of the search for more accurate and useful information. Pauer et al. found that the quality of information on rare drugs on the internet is rather poor. However, the support groups had very valuable information. Rare disease information retrieval analysis is showing increasing interest in clinical research and people want to supplement and exchange information with healthcare providers to make it easier for patients to search.

The study results revealed a lot of things about finding information about rare diseases.

First, the highest search intensity was after getting the diagnosis.

Second, the problem is to find more specific information related to the disease, since most of the information was generalized and scarce in detail. Lack of information about the condition of the disease in everyday life.

Third, the ability to find new information is not easy, since possible sources containing information can be found in specialized medical sources to which ordinary people do not have access. There is also the problem of finding information in Croatian.

Last, difficulty in finding drugs and lack of information from doctors. Moreover, much of the information was not available in Croatian and there was no description of side effects and drug interactions.

To sum up, this information from the study was useful in order to clarify the situation with people suffering from rare diseases and the problem of finding information about diseases and drugs. Moreover, it shows the problem of inaccessibility of information on the example of the Croatian language, thus the team would like to provide all kinds of information in different languages, such as Kazakh, Russian, and English for a complete understanding in Kazakhstan.

1.2.5 “The use of internet and social media for drug information services in pharmacies in Yogyakarta province: a study if Asthma care” review

Fourth is a study “The use of internet and social media for drug information services” **[n]** done by Fajar Ira Juwita, Aris Widayati and Enade Perdana Istyastono from Universitas Sanata Dharma, Yogyakarta, Indonesia.

The aim of the study is to investigate pharmacists' perceptions of using the internet to provide drug information services and deliver these services to asthma patients. The research method was interviews with confirmed professional judgment.

ICT use has risen rapidly. The development of ICT has led to the emergence of e-health, which is also rapidly developing. Furthermore, this affected pharmaceutics where ICT is mainly used in the procurement and storage of drugs and this is called e-pharmacy. Therefore ICT can improve patient adherence to the correct regimen. For example, Canada used social media to connect patients with each other with the help of a healthcare professional as a guide.

For people with chronic conditions, the internet can improve their quality of life. People with chronic illnesses require special attention and treatment, they generally take more than two drugs during long-term treatment. Therefore, the Internet can be used to monitor the use of medicines by patients. Asthma is an increasing chronic disease in Indonesia, but the use of ICT in the treatment of the disease has been rare.

As a result of the interview, many themes emerged.

First theme, the capabilities of pharmaceuticals using the Internet to maintain services. All respondents answered that they use social networks, especially WhatsApp, Instagram and Facebook communicate with patients. In addition, they also use them to find and improve knowledge by finding specific information from other pharmaceuticals in other countries.

Second theme difficult in transformation of roles of pharmacists in e-pharmacy. Some respondents say that patients are more likely to search for information on the Internet than to turn to professional pharmaceuticals, however, they consider this a disadvantage, since information on the Internet can be unreliable and can also lead to big delusions. Moreover, they noted that the pharmacist is obliged to ensure that information on the Internet must be accurate and up-to-date.

Third topic is the relevance of regulation in e-pharmacy. The majority of respondents stated that online pharmacies sell not only over-the-counter drugs, but also prescription drugs, which is an example of the lack of supervision of authorized persons to control drug sales. And the fact that the regulation of such pharmacies is a necessity.

Fourth theme is related to ICT improvement. The respondents who worked at the Primary Health Center said that the main obstacles to the introduction of ICT and the Internet are unstable Internet connections and suboptimal work of services. Additionally, many rural areas are not connected to the internet.

Fifth theme about contribution of pharmacist in on people’s e-health literacy. The respondents argue that pharmacists can increase the literacy of people using the internet and social media. They expect that by using the Internet, the patient will become easier to treat and increase knowledge about drugs.

To conclude, this study allowed the team to look at the situation of the prevalence of ICT in other countries as well as the problems and possible benefits of Internet interconnection for pharmacists and their patients. The team believes that the project can partially provide information for patients and pharmaceuticals who need drugs, especially for people with chronic diseases. In addition, this work pushed the team to implement a system into the project that will show the location of drugs in local pharmacies, which will help increase user satisfaction.

1.2.6 “How Do Patients Expect Apps to Provide Drug Information?” review

Fifth is a study “How Do Patients Expect Apps to Provide Drug Information?” **[n]** done by Anton Grube, Tobias Dehling, Ali Sunyaev from University of Kassel, Germany.

Patients seek information about drugs from a variety of sources. And one of them is mobile apps. When using mobile applications, patients experience a lack of functionality that is more useful than that provided by healthcare professionals.

This study identified 33 features that patients expect to see in mobile apps. Patients are more interested in personalizing the information provided by applications. The current situation in health information systems shows an attachment to big data. The generalization of patients' personal information leads to a decrease in the display of the personal needs of patients. However, healthcare is shifting more towards personalization and patient-centeredness rather than generalization of all information. Patients' lack of adherence to treatment results in increased treatment rates, increased disease rates and poor treatment outcomes. Patients may refuse or forget to take medications altogether. To prevent this, actively remind the patient of the medication regimen and provide more information about the medication. This is critical to the success of the treatment.

The development of mobile technologies is increasing the personalization of information for patients. The availability of information has led to a decrease in non-adherence to medication. For example, this information can be transmitted both through databases and by text messages or reminders. But at the same time, application requirements are based on the expectations of healthcare professionals. Therefore, it affects developers who develop applications based on these requirements. But this leads to the fact that patients are not satisfied with the information that mobile applications provide them with, as well as the presence of unnecessary functionality.

As a result, the research has brought out 33 features that can help increase usability for patients and also show developers what functionality should be in their applications. These functions are divided into four groups: information provision features, graphical user interface features, search and sort features and features providing additional functionality:

1. Check drugs for adverse drug reactions;
2. Well-arranged presentation of drug information;
3. Search drugs by name;
4. Display dosage information based on other factors than age;
5. Simple user interface design;
6. Provide information whether a drug unit can be split;
7. Display dosage information for a drug in different units;
8. Compare two or more drugs;
9. Search drugs by application area/medical condition;
10. Provide information on drugs which may be taken to alleviate side effects;
11. Provide information on how to enhance the effect of a taken drug;
12. Filter and search for side-effects;
13. Provide information if a drug can be dissolved in a fluid;
14. Filter search results by users' personal characteristics;
15. Provide the possibility to store drugs users are taking;
16. Provide a tutorial on how to use the application;
17. Provide information how a drug acts in your body;
18. Provide the ability to store users' personal information;
19. Search drugs by active ingredient;
20. Allow users to comment on their experiences with a drug;
21. Provide additional information for the technical terms or abbreviations;
22. Provide functionality to identify drugs;
23. Provide information on follow-up drugs;
24. Provide functionality to print out selected information;
25. List all other drugs with same active ingredient as currently displayed drugs;
26. Search drugs by a unique local identifier;
27. Filter for requirement of prescription;
28. List all other drugs for the same application area/medical condition as currently displayed drugs;
29. Integrate images/pictograms to illustrate drug information;
30. Search drugs by company/manufacturer;
31. Provide image of drug package;
32. Provide information on homespun remedies;
33. Customizable user interface.

The first five functions are essential, as without them it is impossible to create an adequate application for patients. The sixth to twenty-seventh functions are functions that patients would like to have as they increase the usability and personalization of information. And functions from twenty-eighth to thirty-third are convenient features that would be nice to have in an application, but still they are not so important to most users.

To sum up, this research is very useful information for developers who are developing applications related to information about the drugs. This will make it possible to make a more user-oriented web application, which will undoubtedly increase its need and demand. The development team will rely on this information to provide specifications for the project being developed.

1.2.7 Conclusion

In conclusion, most of the information from these scientific papers and studies has proven useful to the development team. Moreover, it has determined the direction in which web applications will be developed, as well as determine which functions will be in demand for users.

* 1. Market analysis
     1. Introduction

Market analysis is a very important part of choosing a project direction, analyzing user satisfaction and information about the current state of the selected market. This subsection provides brief information about the pharmaceutical market, how valuable the market is and how much it affects the public. In addition, it provides an analysis of market research for its main features and possible resulting market problems. It is also necessary to identify large companies that greatly influence on the market. In addition, after analyzing the development of the market, it will be possible to see what the market opportunities are for increasing or decreasing its size.

* + 1. Brief analysis from the internet

Today, the pharmaceutical market is estimated at around 1.3 trillion [n]. The key companies in the pharmaceutical market are [n]:

1. Johnson & Johnson;

2. Pfizer;

3. Roche;

4. Novartis;

5. Merck & Co.;

7. Sanofi;

8. AbbVie;

9. Takeda;

10. Shanghai Pharmaceuticals Holding.

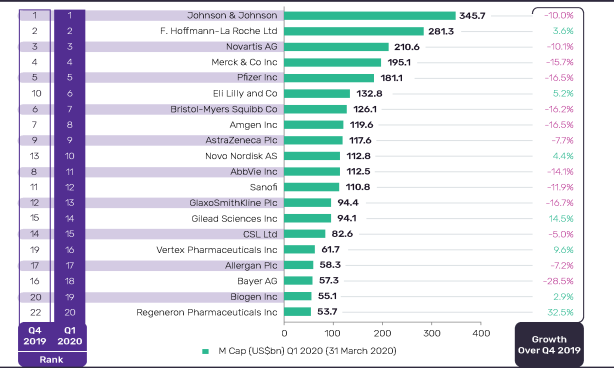


Figure 1.1 [n] – Market cap of biggest pharmaceutical companies in Q1 2020.

As we can see in Figure 1.1, many companies declined their market caps between Q4 2019 and Q1 2020. COVID 19 has caused a lot of damage to the global market. However, this did not have a strong impact on the pharmaceutical market as in other markets.

According to statistics from www.statista.com[n] in global international market of pharmaceutical products grown from 2001 when value of market was 390 billion dollars to 2019 where value of market was 1.25 trillion dollars.

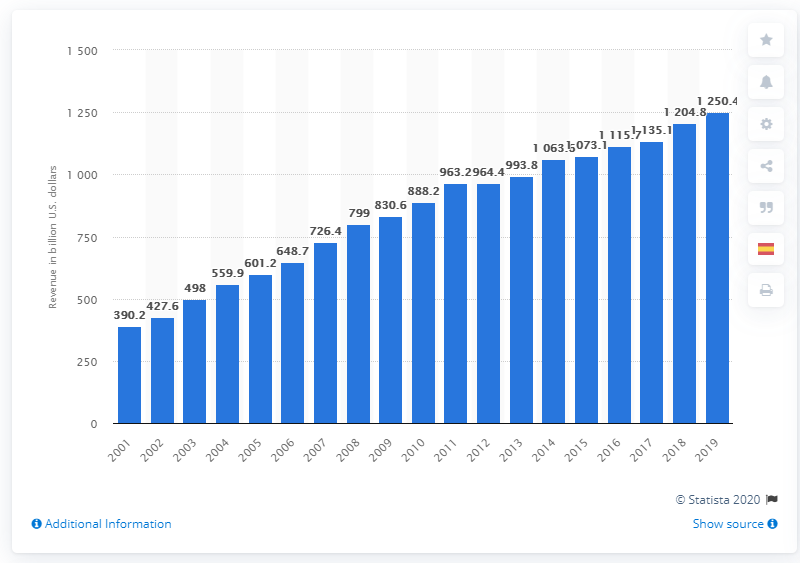


Figure 1.2 [n] – The pharmaceutical market development from 2001 to 2019.

In Kazakhstan, the volume of the pharmaceutical market is 180 billion tenge [n]. In April 2020, about 203 million packages of medicines were sold.

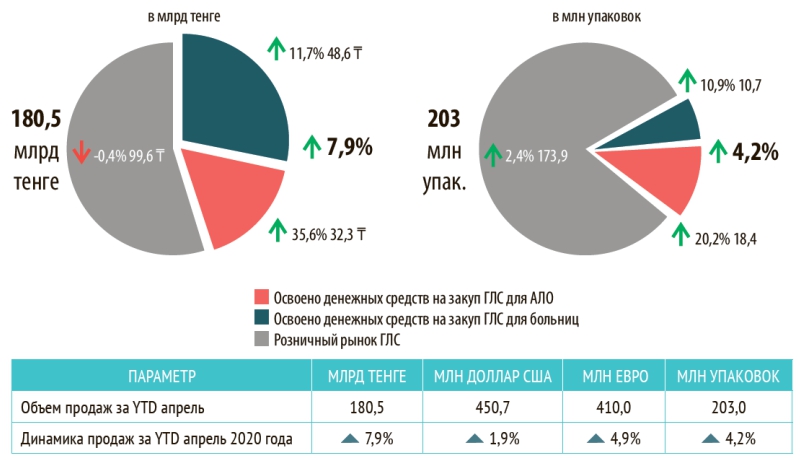


Figure 1.3 [n] – The volume of the retail drug market of Kazakhstan in April 2020.

As noted above, the pharmaceutical market is a very large and emerging market in the global economy. COVID 19 has greatly affected the economy of the entire world for the worse. However, some pharmaceutical companies are doing well and are seeing increased profits.

* + 1. Research on the Pharmaceutical market

Following the research "Main tendencies of the Global Pharmaceutical Market" **[n]**, the pharmaceutical market is a dynamic and fast developing market, especially in developing countries. This is due to the rapid increase in the range of different drugs. Between 2013 and 2017, the market volume increased by 2-6% every year. The chart below shows the market size for each inhabited continent and Japan.

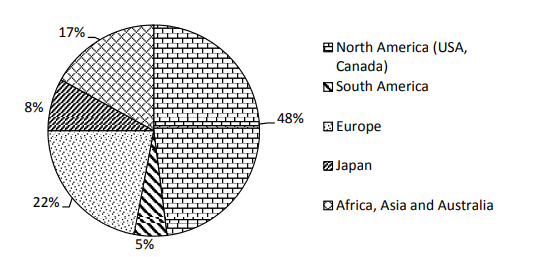


Figure 1.4 - Drug sales in the World Pharmaceutical Market

As shown in Figure 1.3, North America has the largest volume of market is 48%, second is Europe with 22% of market, third is Africa, Asia and Australia together with 17% of market, Japan with 8% and South America with 5% of market. North America is largest because 25 million people in this region suffers from diabetes and 80 million in prediabetes stage. Europe is second due to the spreading of infections in 28 countries, with 76% of local transmission of infection.

The main threat in the pharmaceutical market is the expiration of many patented drugs and can lead to a so-called "patent cliff", which means that patented drugs with large market sales after patent expiration can drastically reduce overall market profits.

* + 1. Conclusion

To conclude, the pharmaceutical market has a large volume in the global economy, and at the same time it is increasing every year. But due to the COVID 19 virus, many pharmaceutical companies lost profits, however, some were able to increase their profitability. This information is useful for studying the state of the market, which will later be used for project design.

* 1. Analysis and comparison of existing software solutions
     1. Introduction

For the chosen theme for the project, there are software solutions that to some extent solved this problem. By analyzing these solutions, the development team will be able to pick up ideas for the project being created and also see what other ideas have not been embodied in the software that we can provide for users. Check what projects exist for specific regions and countries. In addition, the analysis can help with what patterns already exist for creating drug dictionaries and in what form they are best embodied.

* + 1. MedElement review

MedElement [n] is a project aimed at improving the quality of medical services for the population through various online services. For patients has a search for the service of doctors and clinics from Russia, Kazakhstan, Belarus and Uzbekistan. For doctors and specialists, it has a drug dictionary and standards for the diagnosis and treatment of diseases.

When visiting the directory, we see the main page on which there is an input field for searching for a medicine by name, as well as a filter by one parameter, which includes four options. Also on the same page, there is a tab ATC classification, which contains the Anatomical-therapeutic-chemical classification by application such as cardiovascular system, dermatology, etc. The description of the medicine itself contains all the information about the medicine, as well as the field where you can send stuck files about the medicine to your email. One useful thing about information accessibility is switching the language from Russian to Kazakh and vice versa. The approximate purchase price per unit of drug is also indicated.

* + 1. Vidal review

Vidal [n] is a reference guide to drug information for workers and students in healthcare services.

This site has a search engine that includes such parameters as name, alphabet, ATC, Nosology index, clinical and pharmacological index, disease, manufacture, pharmacotherapeutic groups. Drug description consist of general description of drug, contradictions, age and pregnancy restrictions, availability and related drugs, package information, type of pill and etc. Furthermore, details contains order the drug from “Ютека” where prices is provided.

The site contains a medical encyclopedia, which presents medical devices, types of various diseases and disease prevention. Moreover, there is additional information for specialists.

The site has registration and login for specialists. The site has only Russian language support.

* + 1. Drugs.com review

Drugs.com[n] is a drug dictionary site aimed at providing accurate information about medicines for medical personnel in the United States. They want to provide up-to-date information that will be trusted resource of drug information.

It also have a mobile application associated with the site.

The site represented as drug dictionary for USA and English language users. Site are divided into four sections: drugs, side effects, disease and condition, and pill identifier. In addition, site provides the news from pharmaceutical field in USA and worldwide.

Sections of site:

Drugs section is the main section where there search by drug name is presented as well as search in alphabetical order which divided in consumer leaflets and professional monographs. Drug details have information such as description of a drug, warnings, dosage, brief side effects, interactions and popular FAQ. Moreover, details contains could contain images of drug, classification of drug, related drugs, manufacturer, drug status which shows availability of drug and pregnancy restrictions, users reviews of a drug. Manufacturer could have a details that shows information about manufacturer and drugs which him produce. In addition, site has drug interaction where you can input the drugs and system will show that the drug components will mixed which lead to poisoning.

Side effects section is a more informative continuation of information about the side effects of certain drugs. Search by side effects is present only in the form of searching by name. Side effect details contains description, a list of side effects that do not require attention and that require medical attention and additional information for professionals.

Disease section has information about various diseases, allergies and conditions. Disease could be searched by name and by alphabetic order. Disease details contains information such as general description of disease, health guide, disease references, care notes, medication list and Q&A.

Pill identifier is search of drug by pill characteristics, such as shape of the pill, color and number printed in the pill.

Site has authentication system where user can register and login to system. User profile has elements as add illness or allergy that is change the filter on searching drug. Furthermore, user profile has reminder of which pill you need to consume and in which time.

* + 1. rlsnet.ru review

rlsnet.ru [n] is a site of register of medicines in Russia, which has various services and information for specialists and encyclopedic knowledge. The site has a search for all products by name and alphabetically. The site is divided by six sections: drugs and substances, biological active additives, pharmacies and prices, news, product and services, library.

Drugs and substances section contains subsections by manufacturer, ATC classification, disease dictionary, pointer of drugs and etc. Drug details contains general description of drug, manufacturers, quantity of drug in the package, composition, etc. In addition, the user can leave a comment on the drug.

Biological active additives section is a repository of information about biological active additives for dietary, preventive and cosmetic effects.

Pharmacies and prices section contains the regional prices for specific drugs in Russia and list of pharmacies in Russian cities.

News involve last news from the field of Russian pharmacy and news of site.

Product and services section contains information and prices of services of this site such as database of drugs, books and software. In addition, it contains shop where you can order some books and software from register of medicines in Russia.

All information of this site is provided only in Russian language.

* + 1. i-teka.kz review

i-teka.kz is a medical portal for cities in Kazakhstan, providing services and information about pharmacies, doctors, clinics and drug delivery and drug information. The site supports two languages: Russian and Kazakh. The site is intended for patients and consumers. The site has five sections such as pharmacies, drugs, medical centers, doctors and medical services.

Section of drugs involves the list of various drugs and searching of drug by name. Drug details contains instructions for use, similar drugs, drug content and prices in pharmacies. Apart from this, it is possible to order the delivery of the drug from the selected pharmacy.

Section of pharmacies contains the information about pharmacies from Kazakhstan cities and map where is the nearest pharmacy is located. In pharmacy details shown the schedule, contacts and location of pharmacy. In addition, it is possible to search for the drug in the selected pharmacy.

Section of medical centers contains the list of medical centers and searching them by name. In addition, it is possible to filter the search by category of medical center. Each medical center has information about the location of center, list services it is provides, schedule and list of doctors working in it.

Section of doctors involves the list of doctors with their location, search by name and search by specialties of a doctor. Each doctor has information about schedule, location and contracts.

Section of medical services shows the list of services and search by name. Each service has location where it can be provided.

* + 1. rxlist.com review

rxlist.com [n] is a medical dictionary of prescription drugs. All medical terms from MedTerms are written by US pharmaceuticals. The site is provided in English language.

The main page of the directory site has an alphabetical search for drugs and a search by name. There is also a list of popular medical terms. From the main page you can go to the following sections: Drugs A-Z, Pill identifier, Supplements, Symptom checker, Diseases, Dictionary and Media.

Drug A-Z page is involves an alphabetical search for medicines and a list of all medicines grouped in alphabetical order. Each drug has professional description, consumer description and side effects. Professional description has detailed information of the drug as the chemical composition of the medicine with formulas, indications, dosage, warnings and etc. Consumer description contains the general information about the drug such as what is a drug, how consume, images, side effects and what avoid when consuming the drug. Side effects description has an additional information about the drug’s side effects and their divided in common effects and effects that required medical help.

Pill identifier page contains drug search by shape, color and number printed on the drug and drug search by brand. Moreover, there is a slider with images of popular drugs by pill, the top 10 prescription drugs and drugs by side effect in the USA, and an example of commonly abused prescription drugs.

Supplements page is a section about vitamins, herbs and dietary supplements. Page contains search by alphabetic order and list of popular supplements. Supplement details have general information.

Symptom checker page has the symptomcheker by WebMD, list of common and popular symptoms.

Disease page section contains detailed information about diseases, conditions and their classifications. Disease page has search by alphabet and list of diseases classifications. Classification page has filled with details of classification, for example, allergy definition. Disease details has facts, causes, treatment of disease.

Dictionary page is a list of general terms in medicine and search by name of term. Media page has images of different diseases, abuses, conditions and advices. Furthermore, it has quizzes of medical terms.

In addition to the listed pages, there are also tools for the user. First, is a tool for drug interactions, which shows the do the two drugs can mixed composition and if it would be poisonous. Second, is a tool which shows the location of pharmacies in USA.

* + 1. Results

Based on the results of the analysis of six projects, a table 1.1 was made on which the functional features of each project were indicated. Each project is checked for the following functionality: support for two or more languages (Different languages), search and its complexity (Searching), login and registration (Login), user profile, drug information (Drug information), the presence of a product store on the site or a list of pharmacies (Shop and/or list of shops), availability of drugs in other stores (Product in other shops) and prices for drugs (Prices).

Table 1.1 Comparison of projects

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Different languages | Searching | Login | User profile | Drug information | Shop and/or list of shops | Product in other shops | Prices |
| drugs.com | - | +++ | + | +++ | +++ | - | - | - |
| Vidal | - | + | + | - | ++ | - | - | + |
| MedElement | + | + | + | + | ++ | + | - | + |
| rlsnet.ru | - | + | - | - | ++ | ++ | - | + |
| i-teka.kz | + | + | + | + | + | + | + | + |
| rxlist.com | - | ++ | - | - | ++ | + | - | - |

Following table 1.1, the development team decided that more in the priority of functionality and information content, we would follow the drug.com project. However, we also want to add the function of presenting goods in pharmacies and also allow the user to change the language. From the unrealized functionality, the team decided to implement the search for drugs by regional and international names. This can help users find the medicine they need, since in a region such as Kazakhstan there may be names of medicines that do not coincide with international names, which can lead to confusion.

2 SOFTWARE DESIGN PART

2.1 Introduction

Software design is one of the most important and key stages of software development. It includes planning, an abstract representation of the problem and a set of methods that can be used to solve the given problem. Following the theoretical analysis given in the previous part, the development team will use this information to present the problem in the form of various diagrams, textual descriptions and flowcharts. In addition, the team will draw up a development plan where a time frame will be set for each important development stage. Furthermore, this part will analyze the choice of tools, the choice of framework and the main programming language.

2.2 Formation of requirements

2.2.1 Requirements gathering

Requirements gathering was carried out through the analysis of scientific works related to the topic of the project. Furthermore, the requirements gathering was carried out by reviews of existing projects related to theme of the project. However, the team conducted a survey among users related to the color palette used for the interface on the site. This will help to find out which colors are most suitable for the drug dictionary interface and which ones are preferred by our target user.

2.2.2 Roles

The project must have roles to define functionality for certain individuals. The project should have four roles:

1. User. It is the default and all users have it. Its functionality is limited within the framework of using the functionality on the site as a search for various drugs or viewing information about them. To open additional opportunities, the user can register on the system. That will give him the opportunity to get the role of a registered user;
2. Registered user. The role assigned to registered users in the project system. In addition to the capabilities of an unregistered user, he may have additional functionality such as writing a review for a drug or adding chronic diseases or allergies to the user's list of diseases;
3. Moderator. It is the site maintenance role. The capabilities of the moderator include operations on the database such as creating and modifying tables. In addition, the moderator can “mute” users who violate the rules on the site;
4. Administrator. This is the main role of project control. The administrator has the ability to fully manage the database through the admin panel. Can assign moderators among registered users. Moreover, has all the features as all the previous roles;

2.2.3 Highlighting the target user

To designate a specific target user for a project related to information about medicine is a little puzzling since all segments of the population use medicines. However, the development team can bring potentially incoming groups of people to the target users:

1. Patients in hospitals or other healthcare facilities;
2. Doctors in hospitals or other healthcare facilities;
3. People with chronic diseases or allergies;
4. Self-medicating people.

2.2.4 Functional requirements

The project should be a site on which there should be four main sections, an admin panel and pages with additional information.

Sections:

1. Drug section contain search by drugs and details of drugs.
2. Side effects section where user searching and see the details of side effects.
3. Diseases section where user searching and see the details of side effects.
4. Manufacturers section contain descriptions for drug companies and search by manufacturers.
5. Admin panel contains pages for adding, changing and deleting information from the database and offices over users registered on the site
6. Pages with additional information contain information about the site and the project like a page about us, contacts, instructions, etc.

1. Drug section:

1. Site has authorization system where user can login and register;
2. User can add in profile his information such as user diseases and/or allergies;
3. Searching specific drug search through an alphabet of one or two letters;
4. Searching by name or description text of a drug;
5. Searching could filter drugs by user allergies;
6. Search algorithm should capable to search some drugs by using its alternative or local names;
7. Drugs details consist titles of drugs, description, general side effects, diseases, age and pregnant restrictions, dosage, acceptance restrictions, special instructions, drug interactions, composition;
8. Shows related or familiar drug by drug group, drug disease propose and drug category;
9. Shows the presence of drugs in certain shop;
10. User could write review for specific drug, could delete written reviews;
11. Each review contains rating of drug and comment;
12. Moderator and admin could delete comments;
13. Moderator and admin could “mute” user for a certain time;

2. Side effects section:

1. Site has searching algorithms that capable searching by: side effect by text in name and description;
2. Side effect details consist title of side effect, description, drug contain this side effect;

3. Diseases section:

1. Site has searching algorithms that capable searching by: side effect name in alphabetic order, drug by text in name and description;
2. Disease details consist title, description, list of drugs contain this disease, category, related diseases by category;

4. Manufacturer section:

1. Site has searching algorithms that capable searching by: manufacturer name in alphabetic order, manufacturer by text in name and description;
2. Manufacturer details consist title, description, year of foundation, year of termination, list of drugs produced by manufacturer;

5. Admin panel:

1. Admin panel consist the pages database control for drug, disease, manufacturer, links for drug, drug categories, disease categories;
2. Drugs database control page should has list of drugs with pagination, drug search by name, category filter, link for create page, link for edit page and delete page;
3. Create page for drugs consist of form where admin or moderator could add titles for a drug, choice a category for the drug, choice a disease for the drug, choice a drug which is related to the drug, checkbox if drug has contradictions for children, checkbox if drug has contradictions for pregnant, list of diseases for contradiction, list of manufacturers, links for pharmacies;
4. For each drug admin or moderator could add images;
5. Each drug could has description, composition, side effects, dosage, composition, special instructions, drug interactions;
6. Each drug in create page could has side effect with description, general effect and doctor attention effects;
7. Each drug in create page could has contradiction with description;
8. Admin or moderator could edit each drug with same parameters as in create page. Admin could delete drug;
9. Diseases database control page should has list of diseases with pagination, diseases search by name, category filter, link for create page, link for edit page and delete page;
10. Create page for diseases consist of form where admin or moderator could add titles for a drug, choice a category for the diseases;
11. Admin or moderator could edit each disease with same parameters as in create page. Admin could delete disease;
12. Manufactures database control page should has list of manufactures with pagination, manufactures search by name, link for create, edit and delete pages;
13. Create page for manufacturers contains form where admin or moderator could add title, description, year of foundation, year of termination and description of manufacturer;
14. Edit page for manufacturer has a form with same parameters as in create page form. Admin could delete manufacturers;
15. Disease and drug categories database control pages should have list of categories of diseases and drugs, links for create, edit and delete pages;
16. Create page for diseases and drugs must contain a form with parameters such as title of category.
17. Admin or moderator could edit each categories with same parameters as in create page. Admin could delete categories;

6. Pages with additional information:

1. Page “About us” has information about the project, its purpose, main functions and information about the members of the development team;
2. Page “Contacts” contains information about contacts of a project such as phone numbers, location, links to social media and form to contact development team;
3. Page “Pharmacies” has information about online pharmacies that used in project;
4. Page “Instruction” contains instructions on how to use the basic functionality of the site.

2.2.4 Non-functional requirements

The project must contain the following non-functional requirements:

1. Security. The project should provide an acceptable way to protect user information and the project itself. In addition, ensure the site's performance in case of hacker attacks;
2. Performance. The project should provide a quick response to user requests and also provide a quick response;
3. Localization. The project should support multiple languages and allow users to change the language. There should be at least three languages: Russian, Kazakh and English. Moreover, consider multiple languages for use in the admin panel;
4. Personalization. The project should have personalization of search algorithms through the user's choice of their diseases and allergies.
5. Accuracy of information. The information in the project must be satisfactorily accurate and reliable for use by both ordinary users and specialists.
6. Simple User interface. User interface of the project should be simple enough to use and at the same time pleasant for users.
7. Ease of testing and finding bugs. The project should be easy to use testing for errors and also output all possible errors.
8. Robustness. The project should handle some possible scenarios where there may be errors in the use of the functionality by the user. Moreover, handle database errors when adding information to it.
9. Usability. The project should provide convenient use of the functionality by both ordinary users and administrator.

2.3 Risks

2.3.1 Risks analysis and mitigation

Risks are one of the strongest reasons for the failure of many projects. Risks are potential problems that can arise during development, implementation, and use. Risks vary from projecting risks to business risks, but the main ones for us are predictable and unpredictable risks. Predictable risks are risks that we can detect early and can be reduced the likelihood of their occurrence. Unpredictable risks can no longer be detected earlier and therefore they can only be solved during the very appearance.

To highlight many of the risks that may appear during the development of our project, an Excel table was compiled on which the risks, their characteristics and possible solutions were indicated. Each member of the team put forward possible risks with which it is possible to meet and a solution to prevent their occurrence. Each risk has its own level of importance and difficulty of elimination, techniques for mitigation, and whether the risk can be reduced or not.

2.3.2 Results

During the discussion, the development team put forward the following possible risks that may appear as during development. The results for each risk are given in the form of the following parameters: risk name, description, importance in points from 1 to 5, difficulty in points from 1 to 5, risk level in sum of importance and difficulty divided by 2, methods of mitigation. There are five potential risks:

1. Risk name: Parsing data from various online drug shops;

Description: this risk may appear during development and it is connected with whether it will be possible to implement this functionality. Parsing information from various sites is demanding on the load of the system and on the response time, since additional pages are being loaded from other sites during this process;

Importance: 4;

Difficulty: 3;

Risk Level: 3.5;

Methods of mitigation:

1. Use libraries for GitHub or others public resources. The team can use publicly available libraries with functionality to solve this problem.
2. Use the cURL class. cURL is a ready-made class in the PHP programming language and is used to parse pages from sites. If the team chooses the PHP programming language as the main language, then it is possible to use it.
3. Risk name: Searching algorithms for drugs;

Description: due to the specifics of the project and its requirements, there is a risk of the problem of implementation of the search algorithm for drugs due to such features as the support of several languages in the project and changes in the search query when used by a user with chronic diseases;

Importance: 5;

Difficulty: 3;

Risk Level: 4;

Methods of mitigation: absent because the specifics of the project do not allow to take ready-made solutions. Therefore, this risk will require a lot of time from the development team to solve this problem.

1. Risk name: Deadline risk;

Description: The time frame can put the team on the assumption that the project may not be completed in time. Because of this, it is possible that some functionality may not work or even be removed from the project;

Importance: 3;

Difficulty: 2;

Risk Level: 2.5;

Methods of mitigation: With good planning and division of work among all members of the development team, as well as timely completion of tasks will help to avoid this risk.

1. Risk name: Poor security;

Description: Poor security can cause many problems when maintaining a project, such as leaking user information, disrupting the site's performance and losing data from the database. The problem of implementing good security is that the team does not have specialists trained in system security and has no experience in creating protective systems;

Importance: 3;

Difficulty: 3;

Risk Level: 3;

Methods of mitigation:

1. Use libraries for GitHub or others public resources. The team can use publicly available libraries with functionality to solve this problem;
2. Using frameworks with implemented security will help avoid security problems and make things easier for the development team.
3. Risk name: Lack of experience;

Description: This is the first time the development team is engaged in projects with such a topic, and therefore there are problems with the implementation of some functionality as well as with the description of the necessary components for the project. In addition, lack of experience can lead to an increase in the occurrence of the risks described in this subsection;

Importance: 4;

Difficulty: 2;

Risk Level: 3;

Methods of mitigation: Additional training on working with the creation of sites, as well as a detailed analysis of the literature associated with the creation of projects on a similar theme can reduce the likelihood of this risk.

1. Risk name: Bad optimization;

Description: bad project optimization can lead to poor functionality of the functionality, which will entail great inconvenience in using the project;

Importance: 3;

Difficulty: 4;

Risk Level: 3.5;

Methods of mitigation:

1. When developing the program code, focus on optimizing the code execution and checks for execution time;
2. Use of ready-made solutions with good optimization.

2.4 Choice of software development life cycle

Software development life cycle are business methods and standards used to plan, develop and implement software using various techniques. Usually, the stages of the life cycle are divided into six to eight parts: planning, requirements, design, build, document, test, deploy and maintain. Table 1.1 shows the examples of life cycles, and their advantages and disadvantages.

Table 2.1 Life cycle comparisons

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Description** | **Advantages** | **Disadvantages** |
| Waterfall | A model where all stages of development go one after the other and there is no way to back. | 1. Easy to understand and well documented;  2. Can be easy to planned;  3. Good for small projects. | 1. Hard to go back in last stages. |
| V Model | A model in which each stage has a corresponding testing. | 1. Good for small projects;  2. Good quality control. | 1. Hard to go back in last stages. |
| Iterative | A model in which the development and specification of a program is done in part and supplemented during development. | 1. Create high-level design of application before building the product.  2. Good track of defects. | 1. Good for big projects. |
| Prototype | A model where we create working software with limited functionality and test with the end user. | 1. Quick feedback.  2. Good for online systems.  3. Quick implementation of functional software. | 1. Incomplete application may not be used in a full-fledged project. |
| Table 2.1 continuation | | | |
| RAD | A model in which priority is given to the collection of requirements and user testing and the project itself is divided into modules. | 1. Quick development time.  2. Good feedback. | 1. Require a high skilled developer;  2. Require a strong team. |
| Incremental | A model in which the development of program divided into modules that are developed consistently and increasing the functionality of the program. | 1. Easy to risk mitigation.  2. Feedback of customer for each module.  3. Easy to test. | 1. Need good planning.  2. Cost is higher than waterfall. |
| Spiral | Model which similar to waterfall model but has a very high emphasis on risk analysis. | 1. Better risk management. | 1. Complex management.  2. Large documentation. |
| Big Bang | A model where  developers do not follow specific processes or procedures and make a program without a specific requirement and without in-depth analysis. | 1. Simple usage.  2. Require less resources.  3. Flexible. | 1. High risks and uncertainty.  2. Bad for complex projects. |
| XP | Model that aims to produce higher quality software, and higher quality of life for the development team. | 1. Good quality of software.  2. Good for continuous projects. | 1. Require a good skilled developer;  2. Depend team coach.  3. Bad prediction and planning. |
| Table 2.1 continuation | | | |
| Kanban | Model which requires real-time communication of capacity and full transparency of work. | 1. Good quality of software. | 1. Require a good skilled developers;  2. Depend team coach.  3. Bad prediction and planning. |
| Scrum | Model where projects progress via a series of sprints. In keeping with an agile, sprints are time boxed to no more than a month long, most commonly two weeks. | 1. Good quality of software. | 1. Require a good skilled developers;  2. Depend team coach.  3. Bad prediction and planning. |

According to table 1.1, development team decided to take waterfall model, because it is good documented which help us to work out the system even before the start of implementation and it will be easy to plan.

2.5 Development tools and program languages

2.5.1 Introduction

This subsection lists the various tools for software development like text editors and compilers, as well as a listing of the programming languages used in the project. In addition, the team compared the frameworks intended for the development of websites in order to determine which one is most suitable for the project.

2.5.2 Comparison of frameworks

When developing software, it is very important to choose an appropriate main programming language as this can affect all further development in terms of time frame and complexity of implementation. In turn, it is important to adequately evaluate them according to the requirements and specifications of the project itself, since if the programming language does not meet the requirements, this can lead to problems in the implementation of requirements in the program code, as well as a lack of functionality and in other numerous aspects. In addition, the framework for the selected programming language is in a great importance, since the framework can solve some of the problems of the language itself and make it easier to use. The experience with familiar program languages of the development team itself is also important because it will be easier for them to develop in already known programming languages.

The development team carried out a thorough analysis of existing frameworks in various programming languages related to web development and, as a result, will decide which programming language they prefer to solve this problem. Each framework was evaluated according to the following criteria: advantages, disadvantages, team's skill with program language and team's skill with framework.

The following frameworks are evaluated there: Spring on Java, Django on Python and Laravel on PHP:

1. Spring

1.1 Program language: Java

1.2 Team’s language skill: middle;

1.3 Team’s framework skill: none;

1.4 Description: Spring is a Java framework that facilitates development for enterprise applications. It supports many scripts making it noticeably easier to use in large projects. Spring allows you to build applications from Plain Old Java Objects (POJO) and non-invasively apply enterprise services to POJOs. There is also support for cloud programming.

1.5 Advantages:

* Using scripts significantly reduces development time.
* There is a flexible configuration of frame settings.
* Injectors make testing easier.
* Easy to use database connectivity.
  1. Disadvantages:
* It's hard to learn how to use a lot of the framework's functionality.
* Complex framework structure.
* Hard to learn the main concept of framework.

1. Django

2.1 Program language: Python

2.2 Team’s language skill: middle;

2.3 Team’s framework skill: middle;

2.4 Description: Django is an MVT (Model-View-Template) web framework used to build web applications. Reliability and simplicity makes it possible to write very effective applications on it. It is one of the most famous frameworks in the world and it is used in web applications like YouTube, Facebook, and Google.

2.5 Advantages:

* Since Django is written in Python this gives it a lot of convenience in code writing and good optimization.
* Easily scalable.
* Uses the best ones safety measures.
* Well-documented.
* There is a ready-made admin panel.
* Easy database management.
  1. Disadvantages:
* Not for smaller projects.
* Uses regular expression for URL.
* Monolithic.
* Too heavy, some of the functionality may not be used.

1. Laravel

3.1 Program language: PHP

3.2 Team’s language skill: high;

3.3 Team’s framework skill: more than middle;

3.4 Description: Laravel is a free open source PHP framework designed specifically for developing web applications. The framework is based on the concept of MVC (Model-View-Controller) which makes it more modular for development. There is also a modular packaging system with dependencies, utilities for deploying the application. In addition, Laravel connected to a lot of ecosystems such as Lumen, Forge, Nova and etc.

3.5 Advantages:

* The Artisan CLI helps you build your application and also use repetitive commands.
* Using Composer makes it easy to implement ready-made libraries.
* Well-documented.
* Reverse routing system and easy to create and maintain routes.
* Easy database management.
* Integrated with email systems.
* Easy to learn and have free learning courses.
  1. Disadvantages:
* Limited inbuilt support.
* Complex for starters.

As a result, the development team made a table to compare these frameworks according to the following criteria: easy to learn, team's experience, security, optimization, build-in admin panel, complexity, well-documented, database connectivity, easy to integrate libraries, easy to routing, fit to the project requirements.

Table 2.2 Frameworks comparisons.

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Spring | Django | Laravel |
| Easy to learn | - | + | ++ |
| Team's experience | - | + | ++ |
| Build-in admin panel | - | + | - |
| Simple  complexity | -- | + | + |
| Optimization | + | ++ | + |
| Security | + | ++ | + |
| Easy to integrate libraries | - | + | + |
| Well-documented | + | + | ++ |
| Database connectivity | + | + | + |
| Fit to requirements | - | + | + |
| Easy to routing | - | - | + |

Following the comparison in Table 2.2, the development team decided to use Laravel as the main framework for developing the project. The development team will try to reduce all the disadvantages of this framework by using more optimized ready-to-use libraries. Below will be given all the other programming languages used in the project, as well as tools for writing code.

2.5.3 List of additional programing languages and libraries

When developing web applications, in addition to the main programming language, many additional programming languages are used to create the visual part of the web application. For example, languages like HTML, CSS and JavaScript are mainly used. Furthermore, the use of libraries with ready-made solutions for the main programming language is also important as it simplifies project implementation and reduces development time. Below are all additional programming languages that will be used to create the user interface and libraries for the Laravel framework with descriptions and examples.

1. HyperText Markup Language (HTML)

HyperText Markup Language is the markup language for web pages used on the worldwide Internet. This language is interpreted through web browsers through which they are displayed in the form of a graphical image or a user interface. In addition, in this markup language, it can be used together with languages like CSS (Cascading Style Sheets) and JavaScript for additional visualization of a web page and adding dynamics in a visual and programmatic way.

HTML page can be built from various HTML-elements. Each HTML-element represents blocks on which objects such as text, images, lists, links can be visualized. From the HTML5 version it is also possible to use video playback. HTML elements are highlighted in the document by the tags.

Many elements are used along with opening and closing tags in order to indicate the boundaries in which this element is used. All open tags are composed of a specific name surrounded by "<" and ">" shapes. To close the element, use the "/" after the "<" to indicate that this is a closing tag. There are also elements that do not require closing tags, since the element itself points to a single object. For example, the "<image>" tag can be used without an end tag, but you need to put a "/" in front of the ">" sign.

HTML-elements can be categorized into non-semantic and semantic elements. Non-semantic elements are elements that do not have any meaning of content within it. These elements are most of the elements that exist in HTML. Semantic elements have a specific purpose for the content within a given element. For example, the first level heading element "<h1>" has the values of the main page heading. With styling, we can create an element similar to "<h1>" from the "<p>" tag, but it will not have the same meaning to the page as "<h1>". Semantic elements is elements such as “<article>”, “<footer>”, “<header>”, “<main>”, “<nav>”, “<section>”, etc. The main advantage of using semantic elements is that many search engines like Google search and Yandex use priority to semantic elements in their algorithms when searching by text.

Below on figures N.N and N.N is an example of html code and browser interpretation of this code.

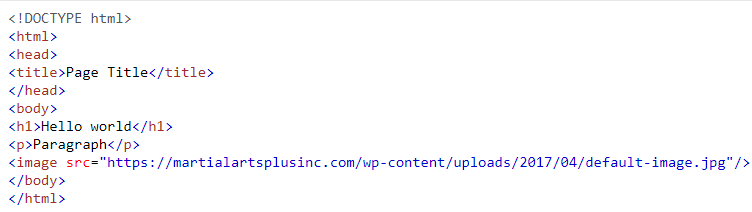


Figure N.N HTML document code example

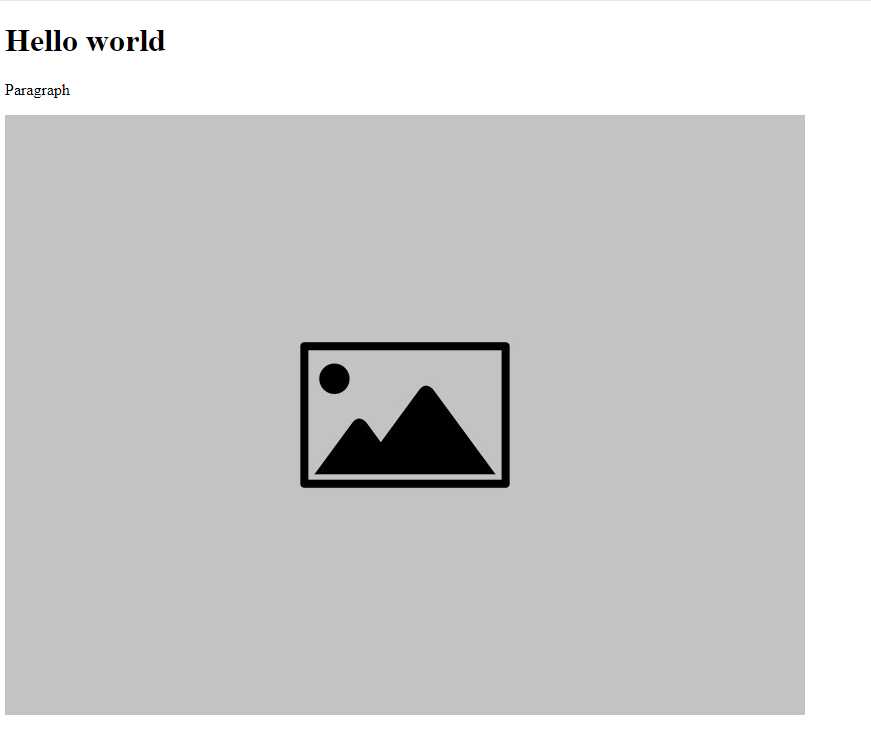


Figure N.N HTML document browser interpretation

2. Cascading Style Sheets (CSS)

CSS adding an effective way to add styles to website pages. In addition, it is a set of formatting rules that determines the appearance of pages, including fonts, colors, layouts, and other visual parameters.

Style sheets let you control the following text properties: font family, font size, text color, background color, text formatting, and link color. In addition, style sheets allow you to determine the display location of block elements, add color to them and move text around, as well as customize fields and borders. A block element is a separate content component that is visually represented as a block. For example, content blocks and panels are also block elements.

Add style elements to the stylesheet, and then apply style elements to the selected page or page element. This method allows you to separate the content from its presentation.

Built-in styles are easier to understand and use, therefore they are recommended for users who have not encountered style sheets before. However, inline styles only support some features of style sheets, as they mix content with its presentation and apply only to individual elements.

If style sheets are used to design a site, users can override the formatting of HTML tags. You can create CSS codes or classes to define the style of individual elements. The CSS class allows you to define and apply style properties to many elements on the page, while CSS code is ideal for styling a single page element. For example, typically in the page structure, codes are used to define header and footer areas, since each page contains only one header or footer, and classes are used to identify duplicate page elements.

CSS contains elements such as selectors. Selectors are used to select specific HTML elements by their type, parameters, position, web browser window size, affiliation, hierarchy, etc. Since a web developer needs to set properties on certain elements on a page, selectors make this task easier. For example, if on a web page you need all paragraphs to have a certain text color, then you can use an element selector “p” and enter all the necessary parameters in it. Table 2.3 below lists most of the selectors available in CSS.

Table 2.3 CSS Selectors.

|  |  |
| --- | --- |
| Universal selector | \* {margin: 0; padding: 0;} |
| Element selectors | p {font-family: Garamond, serif;} |
| Class selectors | .note {color: red;  background: yellow;  font-weight: bold;} |
| Identifier selectors | #paragraph1 {margin: 0;} |
| Attribute selectors | a[href=”google.com”]{  font-weight: bold;} |
| Descendants selectors | div#paragraph1 p.note {color: red;} |
| Child selectors | p.note > b {color: green;} |
| Nursing selectors | h1 + p {font-size: 24pt;} |
| Pseudo-class selectors | a:active {color: yellow;} |
| Pseudo-element selectors | p::first-letter {font-size: 32px;} |

For example, below is figures N.N and N.N showing an example of using CSS in conjunction with HTML.

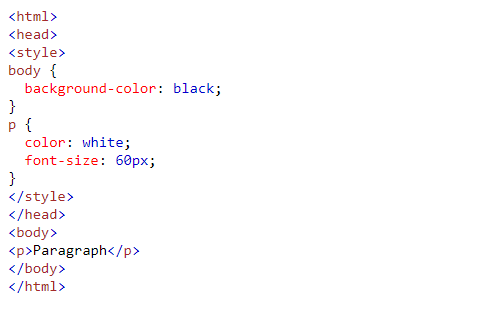


Figure N.N CSS code example

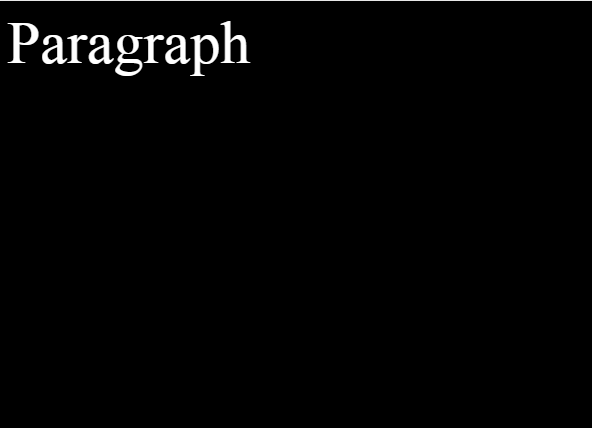


Figure N.N CSS code interpretation

3. JavaScript (JS)

JavaScript is a programming language that supports object-oriented, functional and imperative styles and is based on the ECMA-262 specification. In web development, it is used to create complex graphics on a web page as well as dynamic page interaction with the user. Currently, JavaScript is one of the most used programming language, especially for web development, because almost not one complex and even simple web sites cannot do without using it. All modern web browsers support JavaScript.

Since HTML and CSS are not programming languages in the usual sense, they do not have any arithmetic operations, logical operations, data structures, variables, classes and objects. Therefore, the use of JavaScript is necessary if a web programmer needs to create a dynamically changing page with different choices or graphic interpretation of dynamically changing objects. It is possible to use JavaScript as server-side processing with Node.js, but mostly JS is used in conjunction with other languages like Python, PHP, Java, etc.

One of the peculiarities of using JavaScript in web browser is that it is interpreted at the client level and not at the server level. This makes it possible to reduce the cost of server system resources in some moments when the site does not require access to the database or any confidential information associated with access to authorization. However, you should be aware of information leakage when using JavaScript, so you should not store very important information there.

JavaScript has many features when using it. First, we have the ability to use the storage of information in variables. Due to the fact that JavaScript is a dynamically typed language, when declaring a variable, you do not need to write its type. In addition, when declaring a variable, three ways of declaring a variable are used: through “let”, “var” and “const”. “const” declaration means that the variable will not change its original value during script execution. “let” and “var” declaration is declare variables that can change their value during script execution, however, there is some difference in them. For “var” variables there is no block scope, because of this, by declaring a variable inside some block, it will still be declared globally for the entire script. Unlike let variables, which already accept the restrictions imposed by block visibility.

Second, there is support for many built-in APIs for interacting with different environments, systems or technologies. For example, JS has an API for interacting with the DOM (Document Object Model), which makes it possible to add, change and remove content in an HTML document and change style properties in CSS. Furthermore, there are third party APIs that can add functionality or make JS compatible with third party systems like Google or Twitter.

Third, JS is capable of handling certain events triggered by some action on a web page. For example, there is a button click event in which if the user clicks on a certain button, then this can cause a certain function to be triggered.

Fourth, there is support for asynchronous programming in JS. For example, we need to load a table from the database, during loading it may happen that the table may not be able to be loaded while the code is already starting to perform the following operations, which can lead to errors in the code. Therefore, using asynchrony, it will be possible to eliminate errors and notify that some other side operation has finished successfully.

Below in the figure N.N and N.N is an example of using a JS script in which text is added after the heading in div with ID “p” while click a button with “Add” label.



Figure N.N JavaScript function code example

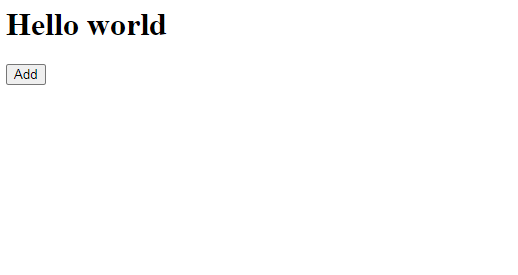


Figure N.N HTML document before button is clicked.

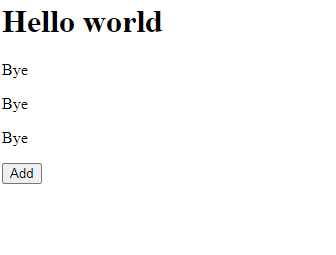


Figure N.N HTML document after button is clicked three times.

4. JQuery

JQuery is an open source JavaScript library that makes it easy to build web applications using JavaScript. JQuery is licensed free of charge under the MIT license. When using jQuery, it is possible to simplify the development of web pages and manipulate them and also have support for many useful technologies.

Installation of this library does not require complex operations, since the library can be loaded through a link to the sources. In addition, one of the advantages is that the library is lightweight, which means that most of the library's functionality will be used to some extent.

JQuery implements three main concepts. First, DOM traversal and manipulation. JQuery has more simplified handling of HTML and CSS elements than regular JS. JQuery has selectors that make it easier to interact with HTML elements. Selectors are denoted by the code “$(‘parameter’)”. By changing the parameter, you can select a specific element according to various criteria: ID, class name, attributes, tag name, etc. By selecting HTML elements through the selector, you can perform various actions with it. For example, using the “hide()” method the element changes display to hidden. JQuery traversal is implemented to select elements through the relationship of other elements to them. For example, through the “parent()” function, you can find the parent of the selected item.

Second, event handling methods. There are many methods in the library to handle specific events. They complement the existing JS event handler and add many other events. Events can be handled by selecting an object through the selector and calling the event method in which the code is written that is executed when the event occurs. For example, in JS there is a problem in which the code can be loaded earlier than the HTML elements of the web page itself. You can solve this problem using the ready () event method, which will be executed when all HTML elements on the page are loaded.

Third, jQuery supports AJAX (Asynchronous JavaScript and XML). AJAX is a concept where the transfer of data from the server to the client side is done in the background. In JQuery, AJAX is capable of handling GET and POST requests to other pages, then fetching the received data to the current page. AJAX processes requests asynchronously, which allows data to be loaded based on response time. For example, by using the get () method, you can get data from another page and use it to display specific data.

For jQuery, there are some libraries that can also complement the functionality of the library itself. For example, there is jQuery UI that adds effects and animations to HTML elements.

The following figure shows N.N, N.N and N.N an example of using JQuery on a web page.



Figure N.N JQuery code example



Figure N.N Page before button is clicked.

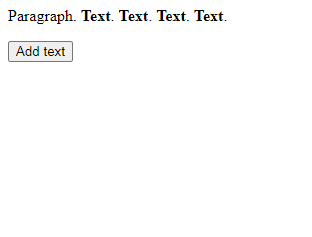


Figure N.N Page after button is clicked.

5. Zebra cURL

Zebra cURL is a PHP library which improve the performance of CURL technology. The library is distributed free of charge under the GNU license[n].

Zebra cURL is mainly based on the functionality of the standard CURL. CURL stands for Client URL and it is also a PHP library. Since PHP 4.0.2, CURL is the standard library.

The purpose of CURL is to receive HTML content via HTTP requests. Roughly speaking, be able to take data from third-party web applications. Requests can be of two types POST and GET. POST requests send some specific data based on the request from the form. GET requests send a request to the server based on the URL link.

To start working with CURL, the initiation of the library class itself is first declared through the “curl\_init()” command. Next, the URL address to which the request will be sent must be described. After that, you can configure various parameters such as the type of connection, encryption algorithm, request timeout, etc.

Main difference between Zebra cURL and CURL is that it is better optimized. In addition, it implements support for multi-threaded page processing, which gives a significant increase in the sending of requests. Furthermore, the library can store a cache of the requested pages, instead of sending new requests, because of the limitations of the hardware of the third-party web applications themselves, which can lead to very long waiting times for requests from web applications.

Below in figures N.N and N.N is an example of using Zebra cURL which displays all links from the main page of the site dl.iitu.edu.kz.



Figure N.N PHP code of using Zebra cURL.

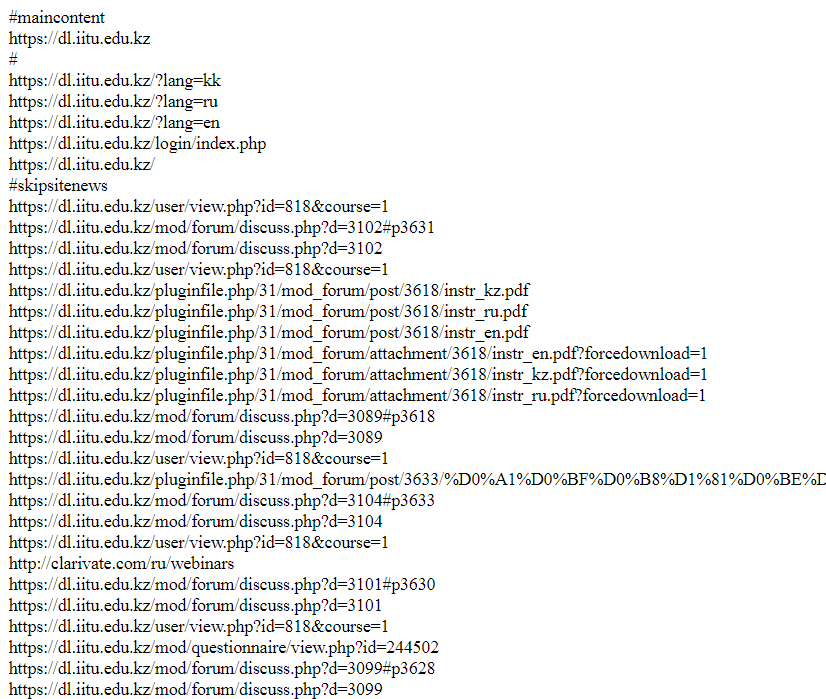


Figure N.N Result of request from dl.iitu.edu.kz.

6. MySQL

MySQL is a database management system program that stores a collection of data. In MySQL each database has interfaces for creating, managing, searching the data. MySQL is based on the concept of relational databases that are divided into tables and their relationships between them.

Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

MySQL can be built and installed manually from source code, but it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions, the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

An additional program php PhpMyAdmin is used to administer MySQL.. PhpMyAdmin allows you to administer MySQL server through a browser, run SQL commands and view the contents of tables and databases. The application uses a very convenient and intuitive interface, where in many cases you do not need to use the SQL language to create and manage databases. There is also detailed usage documentation for users. By using PhpMyAdmin, you can significantly reduce the time to create a database, as well as simplify its management and reading.

There are, however, limits to how far performance can scale on a single server, so on larger scales, multi-server MySQL deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations.

2.6 UML diagrams

2.6.1 Introduction

Unified Modeling Language (UML) is a standardized modeling language for charting that is used in software development. They help in providing visualization of systems, solutions for developers, as well as building documentation describing all the artifacts of the program.

Using these diagrams, the development team will visualize an abstract solution to the problem related to the topic of the project, which in its turn will increase the understanding of how to solve these problems and provide a reliable representation of the system. The following diagrams were used to visualize the project: Use case diagram, Activity diagram, Class diagram, Deployment diagram. To draw the diagrams, the team used the online service draw.io, which contains many elements for drawing UML diagrams.

2.6.2 Use case diagram

To show how our users interact with our system and what functionality is available to a certain type of user, we made a use case diagram. The diagram has elements such as actors who are users of the system or third-party systems, ovals represent the use cases of actors and box represents bounds of system and lines from actors to use case represents the relation.

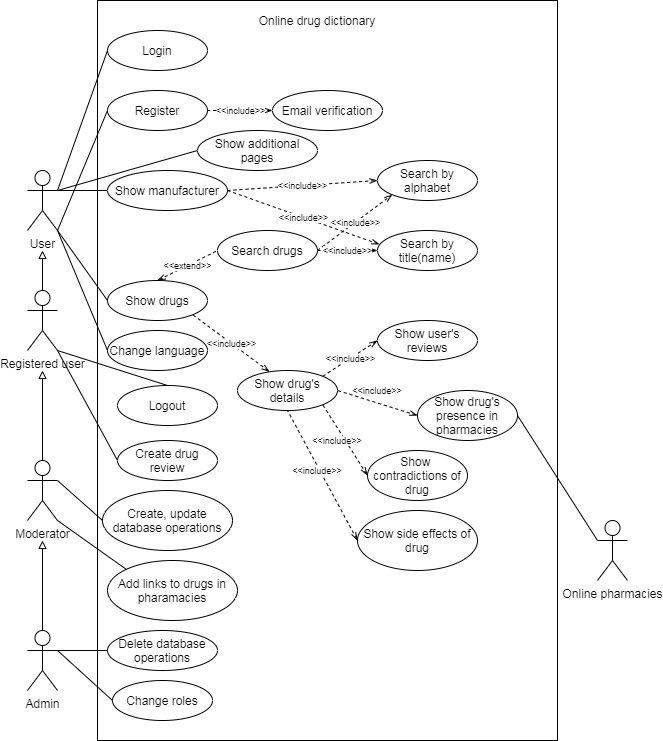


Figure 2.1 Use case diagram

2.6.3 Activity diagram

Activity diagram is used to show a certain process of the program in the form of a sequence of activities. The activity workflow starts from the starting point to the finish point where the activity can split along several or different paths. The starting point is drawn as a black filled circle with red borders. The end point is a black filled circle within a red open circle. Activity is shown as a rounded circle. The flow is shown as arrows from one activity to another. The decision is shown as a diamond. Swim lanes shows the group of related activities.

The team made two activity diagrams:

First diagram is diagram on figure 2.2 of database interaction of admin and moderator in the project site.

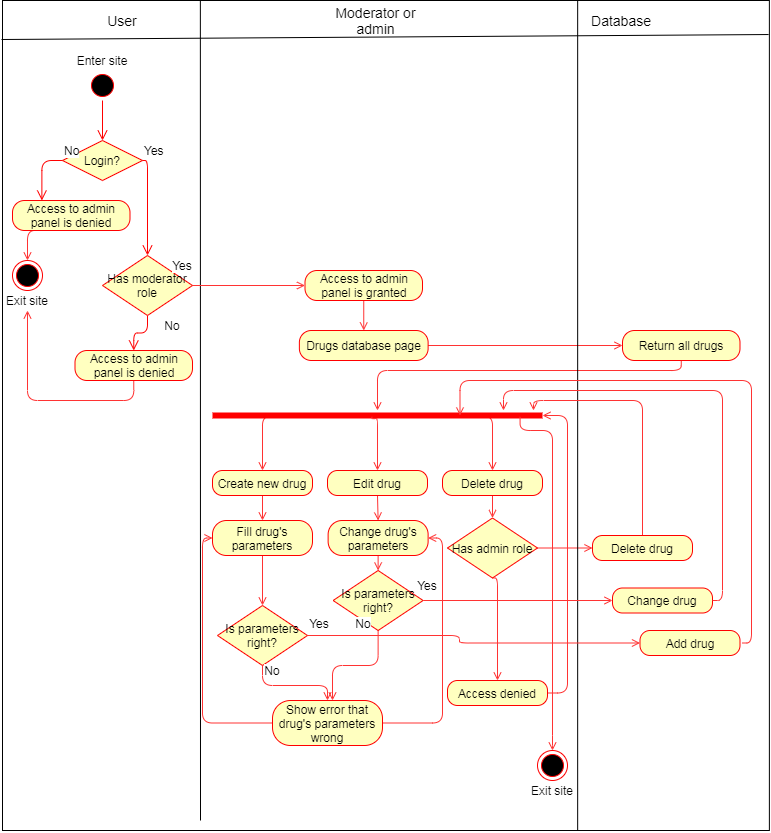


Figure 2.2 Drug search activity diagram

Second diagram on figure 2.3 is diagram of drug search in the project site.

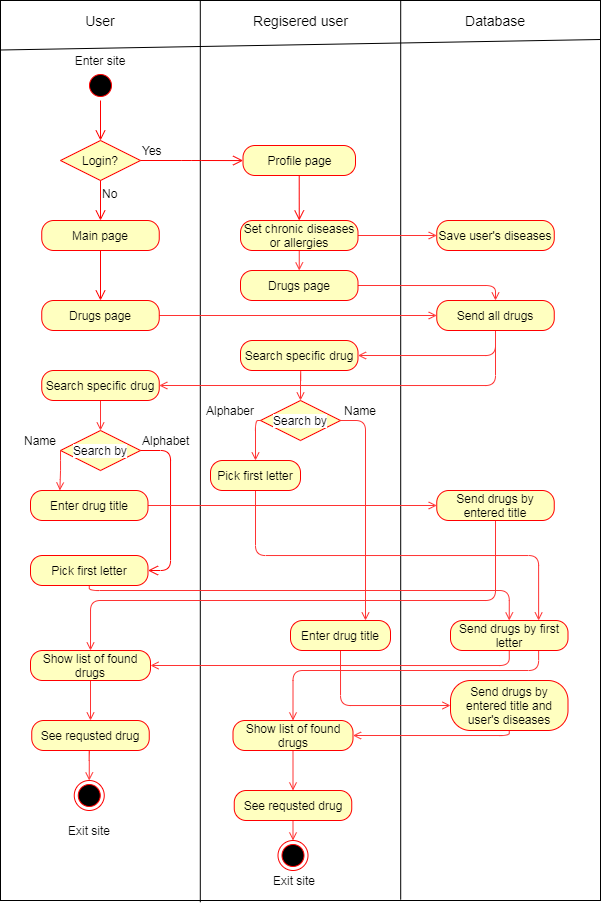


Figure 2.3 Drug search activity diagram

2.6.4 Class diagram

Class diagrams are an object-oriented representation of the system and show the objects of the system in the form of classes, as well as the relationships between them. In addition, class diagrams could be used to model data using the example of non-relational databases. Development team used class diagram to describe the database and class of program language.

Each class has attributes and operations. Attributes define the characteristics of a class such as weight, category, description, etc. In addition, attributes have a name, accessibility such as public, private, protected and data types such as string, integer, float, etc. Operations show the behavior of a class as methods. Each operation has name, input types and output type.

Relations between classes represented as lines with or without specific arrows. On figure 2.4 used only one type of relationship is bi-directional association. With bi-directional association classes knows the existence of each other.

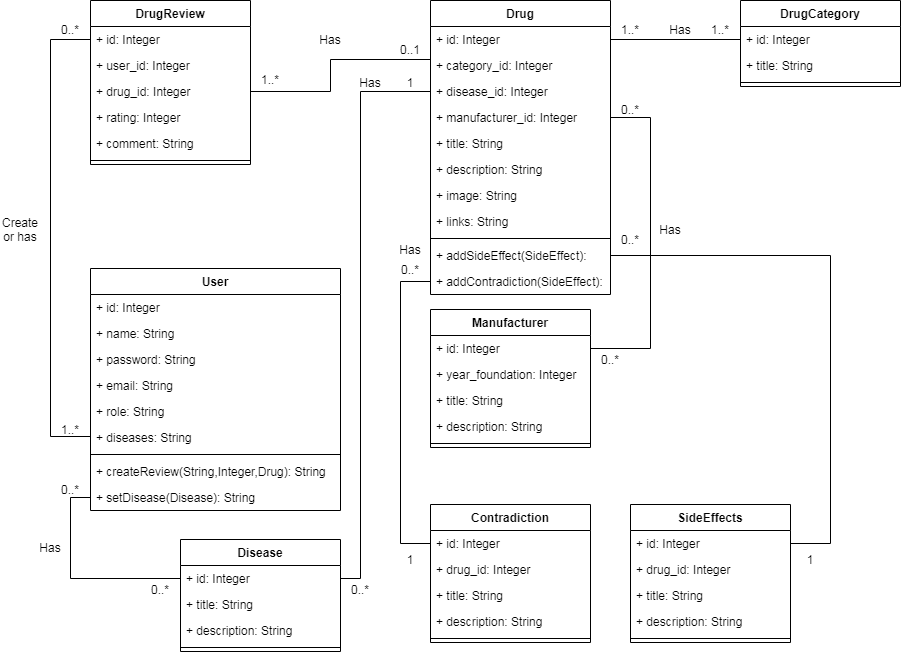


Figure 2.4 Drug dictionary class diagram

2.6.5 Deployment diagram

Deployment diagram is designed to modeling the physical representation of an object-oriented system. Diagram showing system architecture and deployment of software artifacts inside hardware. The diagram contains the following elements: Devices in diagram represented as blocks in that artifacts is located. Artifact is a part of software that are located inside devices. Solid line shows the connection between the hardware.

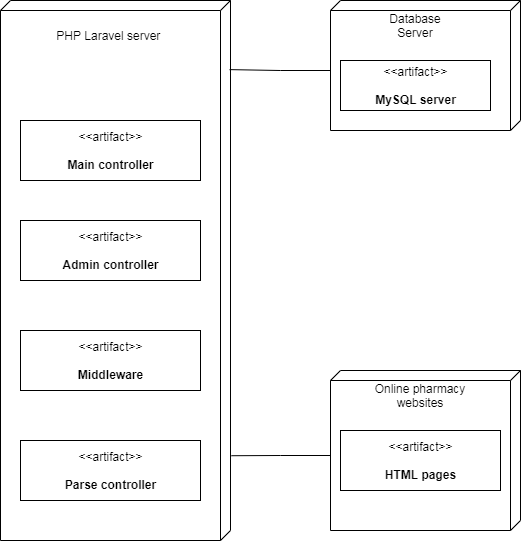


Figure 2.5 Drug dictionary class diagram

As we can see in Figure 2.5 the system has main PHP Laravel server that has a lot of controllers to control the requests. Parse controller handles parsing from online pharmacies. Main and admin controller handle database and main pages requests. Middleware handles filtering the HTTP requests.

2.7 Calendar plan

In this subsection, development team have created a plan that can still be finalized over time. However, the main criteria and processes for developing the project have already been highlighted at this stage. Team made a plan following the SMART concept. S stands for specific which means what goals need to reach. M stands for measurable, for example, team measure speed of page loading that shows quality of our code. A stands for achievable which means that the goals team need to reach is adequate and team can handle them. R stands for relevant, it needs because tasks in our project should be fully usable in the project. T stands for time-based what considered as starts and deadline time of our tasks. Below is our timetable with marking who will do what work. Our team is made up of software developers, so it is easy for us to assign all team members to the same task.

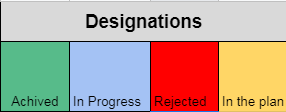


Figure 2.4 Calendar plan designations

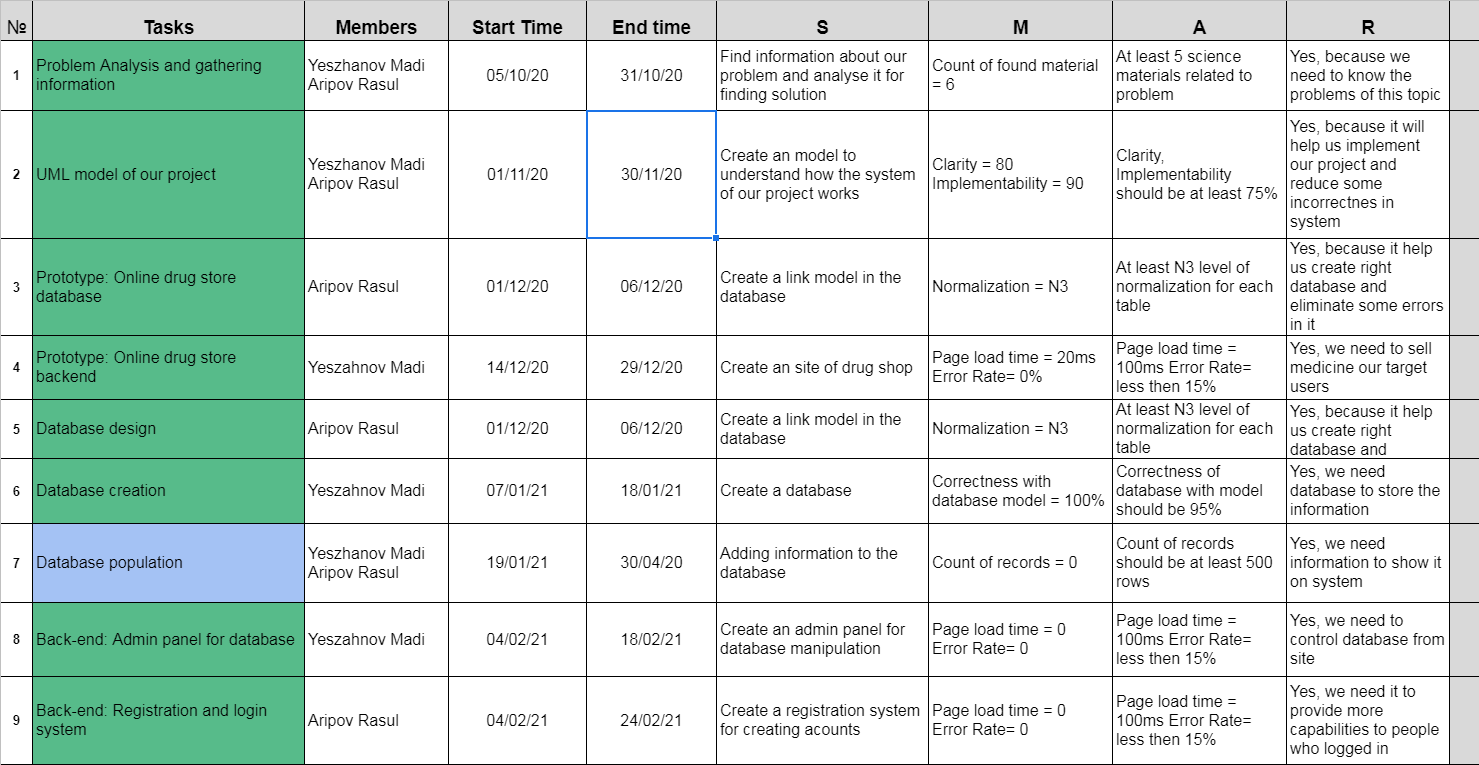


Figure 2.2 Calendar plan first part

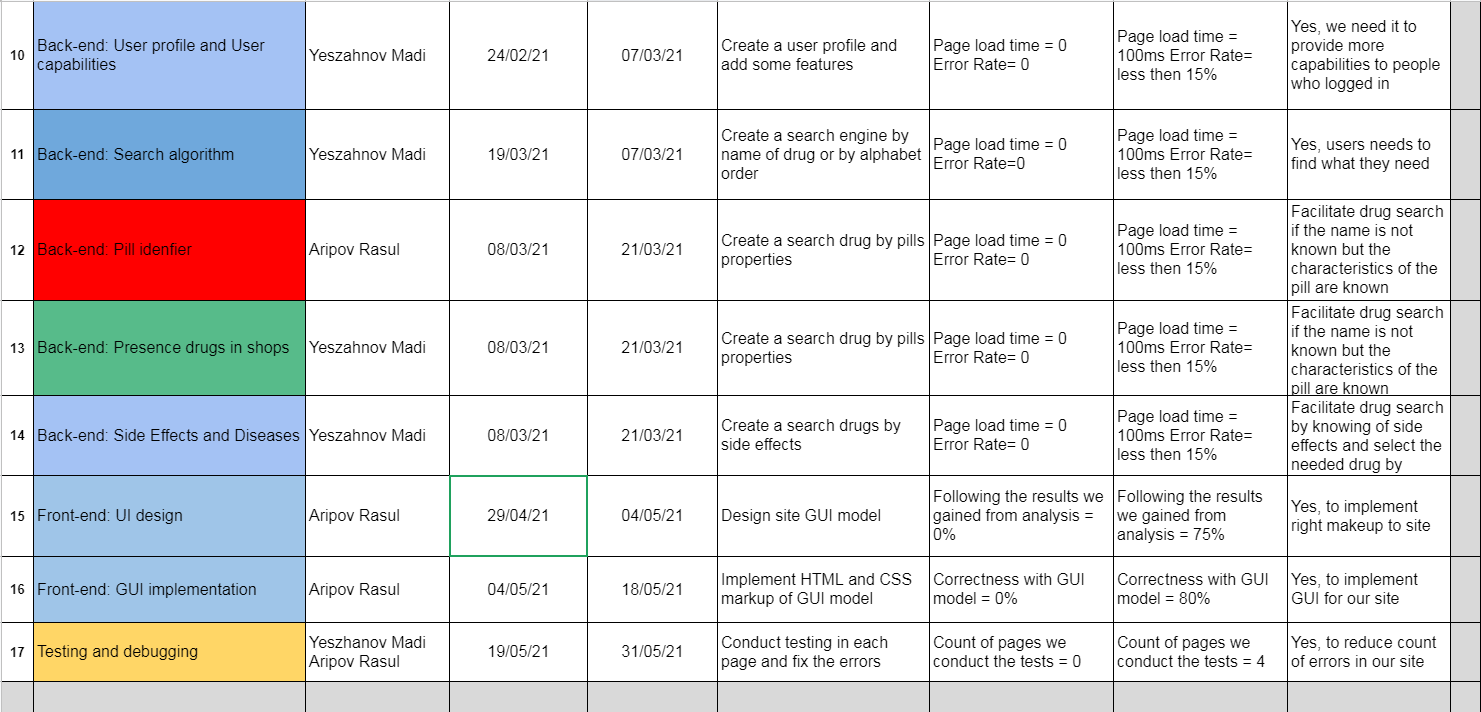


Figure 2.3 Calendar plan second part

2.8 Database design

The MySQL database is the primary storage medium for this project. Therefore, the team developed specifications, characteristics and diagrams for the database with this in mind. In addition, the team developed database based on the theory of creating a relational database, from which using the methods of normalization and design of relationships between entities (tables).

Basic requirements of the database:

* All entities in the database must have a minimum normalization level of N3.
* Localization support for three languages (Russian, English, Kazakh).

Roles in database:

* Administrator. Has full control over the database (read, add, change, delete);
* Moderator. Has partial control over the database (read, add, change, delete in certain tables);
* User. Has read-only access from the database.

In order to represent the relationship between entities, the development team created an ER-diagram. ER-diagram based on Crow's Foot notation. The team used the simplified version due to the fact that there are too many relationships between entities (Drug table has eleven relationships with tables).

The diagram has the following elements:

* Entity (table) - is the main element for storing data by classification data (Drugs, Manufacturers, Categories). Each table has attributes (columns) and records (rows);
* Relationship - represented as a line where the ends are connected between the tables. The ends are responsible for the type of relationship between the tables. Relationship types: solid end - solid end, one-to-one; Crow foot end - solid end, many-to-one; solid end - Crow foot end, one-to-many; Crow foot end - Crow foot end, many-to-many;
* Primary key (PK) - main identifier of the record;
* Foreign key (FK) - it is the identifier of table associated with it.

Below is an ER diagram divided in figures N.N and N.N.

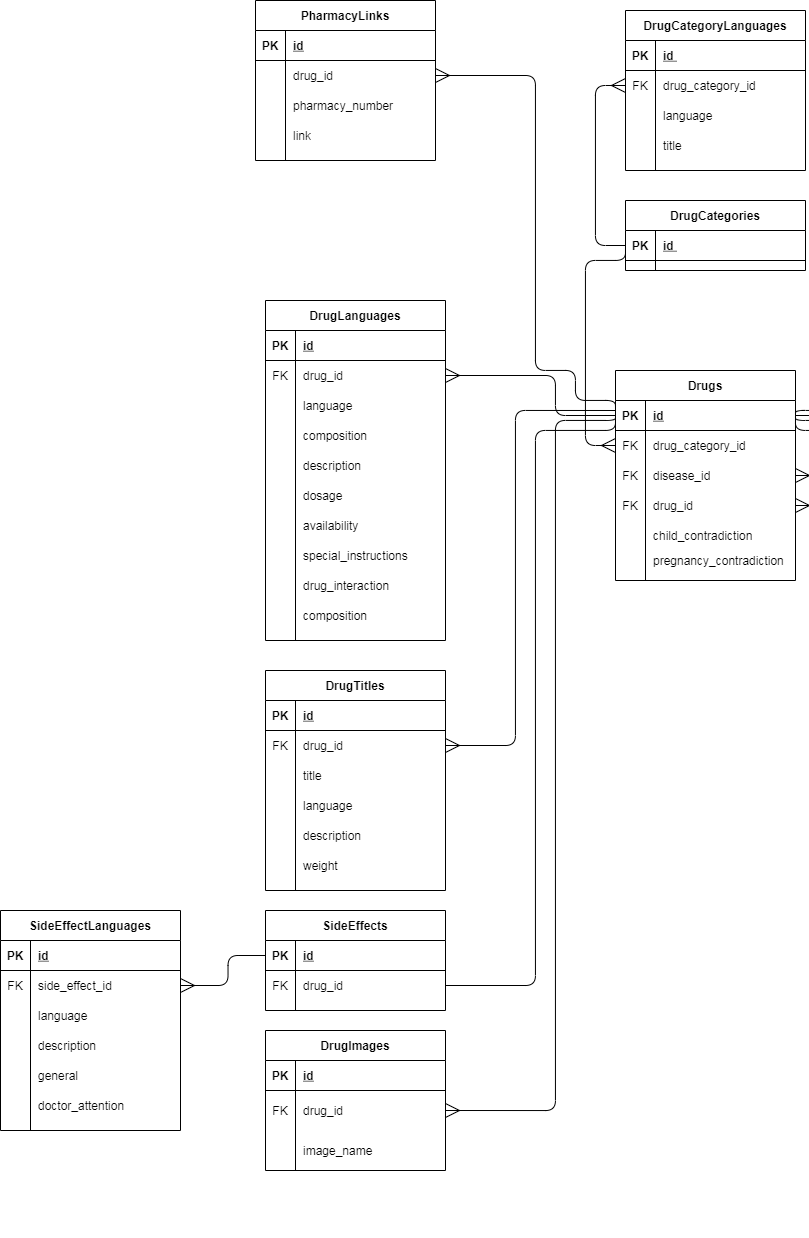


Figure N.N ER-diagram part one.

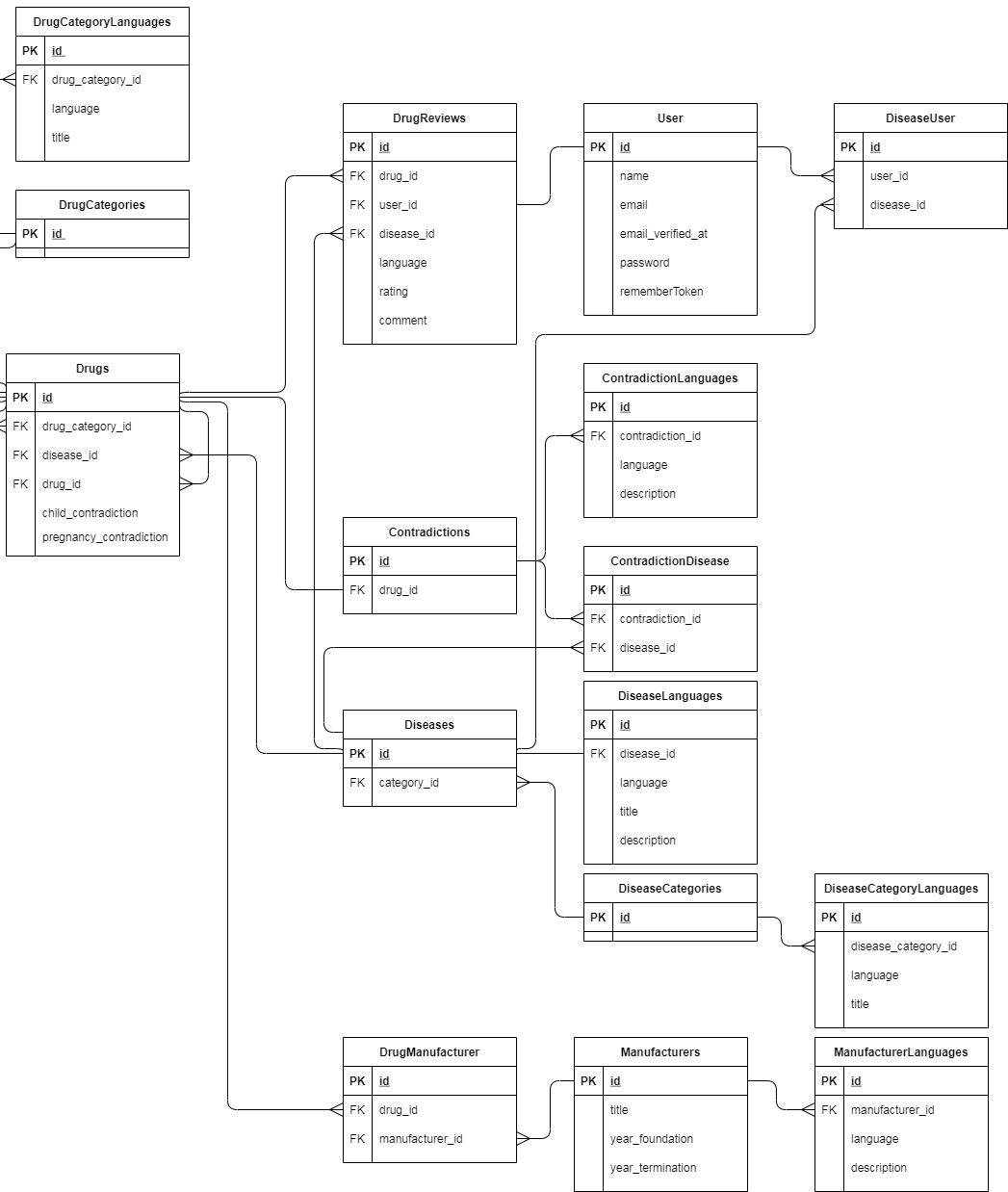


Figure N.N ER-diagram part two.

Each table in the ER-diagram is compiled following 3N normalization. To achieve this normalization for this table, the following requirements must be met:

* Each attribute must have an atomic value;
* Each table must have full dependency of its attributes;
* Each table must have no transitive dependency of its attributes.

To clarify the properties of attributes on each entity, the development team compiled a tables that contains information where each attribute is described in a more detailed way. Each attribute has the following descriptions: attribute name, short description, data type and size, constraints. Some entities are combined into one table because they work together as a whole (Language, Image and Title tables). Below are tables from N.N to N.N. Only the most important tables for the functioning of the project are given. The rest of the tables are described in Appendix N.

Table N.N Drugs and drug related tables.

|  |  |  |  |
| --- | --- | --- | --- |
| Drug | | | |
| **Name** | **Description** | **Data type and size** | **Constrains or notes** |
| id | Id of drug | BigInt | Primary key |
| drug\_category\_id | Category of drug | BigInt | Foreign key |
| disease\_id | Disease of drug | BigInt | Foreign key |
| drug\_id | Id of drug related to | BigInt | Foreign key |
| child\_contradiction | Has contradiction to child | Bool | NOT NULL |
| pregnancy\_contradiction | Has contradiction to pregnant women | Bool | NOT NULL |
| DrugLanguage | | | |
| id | Id of drug language | BigInt | Primary key |
| drug\_id | Id of drug | BigInt | Foreign key |
| language | Language of drug | TinyInt | Range from 1 to 3 |
| composition | Composition of drug | Text | NULL |
| description | Description of drug | Text | NULL |
| dosage | Dosage of drug | Text | NULL |
| availability | Availability of drug (prescription or non- prescription drugs) | Text | NULL |
| special\_instructions | Additional instruction of drug | Text | NULL |
| drug\_interaction | Interaction between other drugs. | Text | NULL |
| DrugTitles | | | |
| id | Id of drug title | BigInt | Primary key |
| drug\_id | Id of drug | BigInt | Foreign key |
| title | Title of drug | Varchar(255) | NOT NULL |
| language | Language of drug title | TinyInt | Range from 1 to 3 |
| weight | Weight of title to show the main title | Unsigned TinyInt | NOT NULL |
| DrugImages | | | |
| id | Id of drug iamge | BigInt | Primary key |
| drug\_id | Id of drug | BigInt | Foreign key |
| image\_name | Path and name of image | Varchar(255) | NOT NULL |

Table N.N SideEffects and SideEffectLanguages tables.

|  |  |  |  |
| --- | --- | --- | --- |
| SideEffects | | | |
| **Name** | **Description** | **Data type and size** | **Constrains or notes** |
| id | Id of side effect | BigInt | Primary key |
| drug\_id | Id of drug | BigInt | Foreign key |
| SideEffectLanguages | | | |
| id | Id of side language | BigInt | Primary key |
| side\_effect\_id | Id of side effect | BigInt | Foreign key |
| language | Language | TinyInt | Range from 1 to 3 |
| description | Description | Text | NULL |
| general | List of side effects | Text | NULL |
| doctor\_attention | List of side effects that requires medical attention | Text | NULL |

Table N.N Contradictions and ContradictionLanguages tables.

|  |  |  |  |
| --- | --- | --- | --- |
| Contradictions | | | |
| **Name** | **Description** | **Data type and size** | **Constrains or notes** |
| id | Id of contradiction | BigInt | Primary key |
| drug\_id | Id of drug | BigInt | Foreign key |
| ContradictionLanguages | | | |
| id | Id of contradiction language | BigInt | Primary key |
| contradiction\_id | Id of contradiction | BigInt | Foreign key |
| language | Language | TinyInt | Range from 1 to 3 |
| description | Description | Text | NULL |

Table N.N Diseases and DiseaseLanguages tables.

|  |  |  |  |
| --- | --- | --- | --- |
| Diseases | | | |
| **Name** | **Description** | **Data type and size** | **Constrains or notes** |
| id | Id of side effect | BigInt | Primary key |
| drug\_id | Id of drug | BigInt | Foreign key |
| DiseaseLanguages | | | |
| id | Id of disease language | BigInt | Primary key |
| disease\_id | Id of disease | BigInt | Foreign key |
| language | Language | TinyInt | Range from 1 to 3 |
| title | Title | Varchar(255) | NOT NULL |
| description | Description | Text | NULL |

Table N.N User table.

|  |  |  |  |
| --- | --- | --- | --- |
| User | | | |
| **Name** | **Description** | **Data type and size** | **Constrains or notes** |
| id | Id of user | BigInt | Primary key |
| name | User is nickname | Varchar(255) | NOT NULL |
| email | Email to login | Varchar(255) | NOT NULL |
| email\_verified\_at | Time of verification of email address | Timestamp | NULL |
| password | Hash mb5 of user’s password | Varchar(255) | NOT NULL |
| rememberToken | Token of last logged device | Varchar(100) | NULL |

3 IMPLEMENTATION PART

3.1 Introduction

In this part, the development team described how and with what methods they tried to implement all the described functionality of this project. Each task has already been assigned during the development of the software, referring to the tasks described in the calendar plan. The team described the implementation for the following tasks: Database creation, Admin panel for database, Registration and login system, Presence drugs in shops and GUI implementation. For each task, implementation process, task goals and results.

3.2 Database creation

**Goal**: Create a database using the Laravel framework utilities.

**Implementation process:**

Creation of models, migrations and factories in Laravel for MySQL database: This implementation method is intended to create a description, migrating tables to MySQL using models in Laravel.

The Laravel framework is based on the MVC (Model-View-Controller) software design pattern. The model is responsible for the interaction of the application with the database in the form of various models (sometimes called classes) and having various attributes like username, email, etc.

Laravel uses migrations to create tables in MySQL. In addition, in the migration there is already a complete description of the attributes and their data types. With the help of “*php artisan make: model [name of model] –m –f*” command we can create a new model, migration which attached to model and factory of the model. Below is an example of creation of model and its migration:

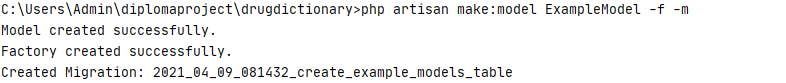


Figure 3.1 Creation of model, migration and factory.

For each model, a corresponding model class is created. The class can describe the relationship between other models as a function with the name of the model having a relationship. Types of relationship correspond to types of relationships in relational databases. However, the class itself does not describe the model attributes, because in Laravel, attributes are added to the model’s migration class.

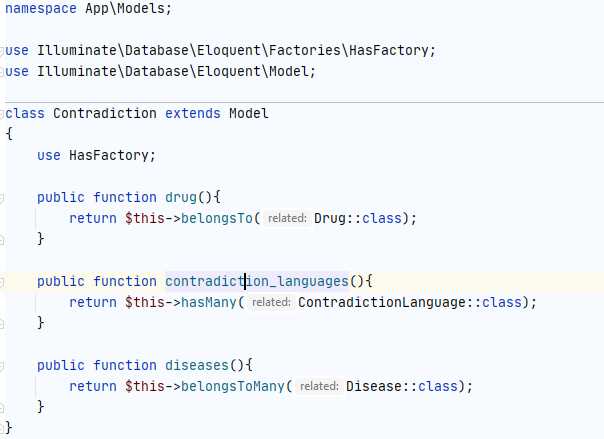


Figure 3.2 Example of Model class.

As mentioned earlier, the migration describes the attributes for the model. To do this, there is an “up” function in the migration class, in which attributes with their name, data type and exceptions are written through the table schema class. In addition, the migration has a “down” function to describe the migration rollback process when a table is deleted from the database.



Figure 3.3 Example of Migration class.

For additional control and testing of models and migrations, the Factory class is can be used. This class is used to initially populate the model tables in the database during the migration. Each Factory class has a model object that it will fill and a “definition” function that returns an array with the model's attributes. To fill in the values in the attributes, an object of the faker class is used in which ready-made phrases, names of places, names, dates are written.



Figure 3.4 Example of Factory class.

To populate tables with Factory classes, the DatabaseSeeder class is used. It declares the Factory classes that must populate specific tables in the database. Furthermore, for each Factory class, you can describe the number of records. However, when filling, you must first fill in the parent tables, because when filling in the child tables, an error may occur that there is no record in the parent table.



Figure 3.5 Example of DatabaseSeeder class.

Now, having created these classes and configured them, you need to migrate the created migrations. To migrate to Laravel, you must use command “*php artisan migrate*”. After entering this command, Laravel checks all added migrations and creates tables in the database following the settings in all classes. If an error occurs, the migration is reset to zero. If the migration has already been added to the database, then Laravel ignores it even if there are changes in it.

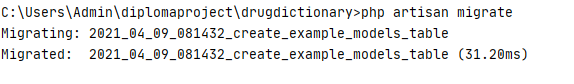


Figure 3.6 Example of migrate command.

Moreover, by adding the “*--seed*” argument to the migrate command, you can immediately populate the database through the DatabaseSeeder after creating the tables. However, you can use “*db:seed*” command if you need to seed the table separately or repeat the seed.

If the migration was done incorrectly or the developer wants to change attributes in migrations or delete it, then there is an opportunity to roll back the last migrations by using the command “*php artisan migrate:rollback*”. In addition, you can configure the number of steps to roll back the migration using the attribute “*--step=n*” where n is a number of steps.



Figure 3.7 Example of rollback command.

**Results:**

As a result, using these Laravel functions, the team created the entire database, models and all tables following the requirements set in the previous part. Below is an example of MySQL database in phpmyadmin GUI.

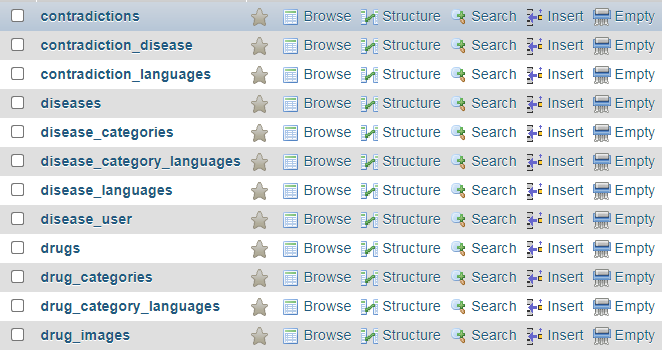


Figure 3.8 List of created tables.

Furthermore, the development team compiled a table 3.1 that shows how the created models and tables meet the requirements and ER-diagram. The table consists of the name of the group of tables (Drug and drug language), the normalization level of the tables and the percentage of compliance with the requirements.

Table 3.1 Level of compliance with the requirements.

|  |  |  |
| --- | --- | --- |
| Name of table group | Level of normalization | Compliance with requirements |
| User | N3 | 100% |
| Categories | N3 | 100% |
| Drugs | N3 | 100% |
| Diseases | N3 | 100% |
| Side effects | N3 | 100% |
| Contradictions | N3 | 100% |
| Manufacturers | N3 | 100% |

3.3 Admin panel for database

**Goal**: Admin panel for database by using Laravel controllers.

**Implementation process:**

Laravel uses classes called Controller to handle client-side requests. Controllers are responsible for the logic of the behavior of the web application itself when a request to the server appears. With the help of controllers, it is possible to process certain requests and display information corresponding to the request. In addition, controllers group logically related requests. For example, requests that are related to the Drug model (create, modify, output).

To create a controller, you could use the command “*make: controller [path]/[controller name]*”. This command automatically creates a controller and binds it to the namespace. Moreover, it is possible to add arguments like “*--resource*” and “*--model=[model name]”.* “*--resource*” argument creates a controller from the template (index, show, create, store, edit, update and destroy). “--model” creates a controller related to chosen model.



Figure 3.9 Creation of controller.



Figure 3.10 Example of controller.

Now that we have created a controller and bound it to the model, we must bind each function in the controller to the request. To do this, we use the web.php file in which we must write the route. Each route is tied to a specific controller and its function, URL address and name. Moreover, the route has several types: GET, POST, PUT and DELETE. GET requests display information depending on the parameters in the URLs entered in web browser. POST requests arise when filling out a form and process information from the entered form. PUT also requires a form however it is only used to change data in the selected row of the table. DELETE is used to delete a record from the database.



Figure 3.11 Example of routes.

Next, to display the processed data on the page from the controller, the View pages is used. They are HTML pages that display information processed by the controller. Basically, the controller will return the view page on a successful request. All view pages are stored in the framework resources.



Figure 3.11 Example of view pages.

**Results:**

Now, using this classes, team can output data to the web browser.

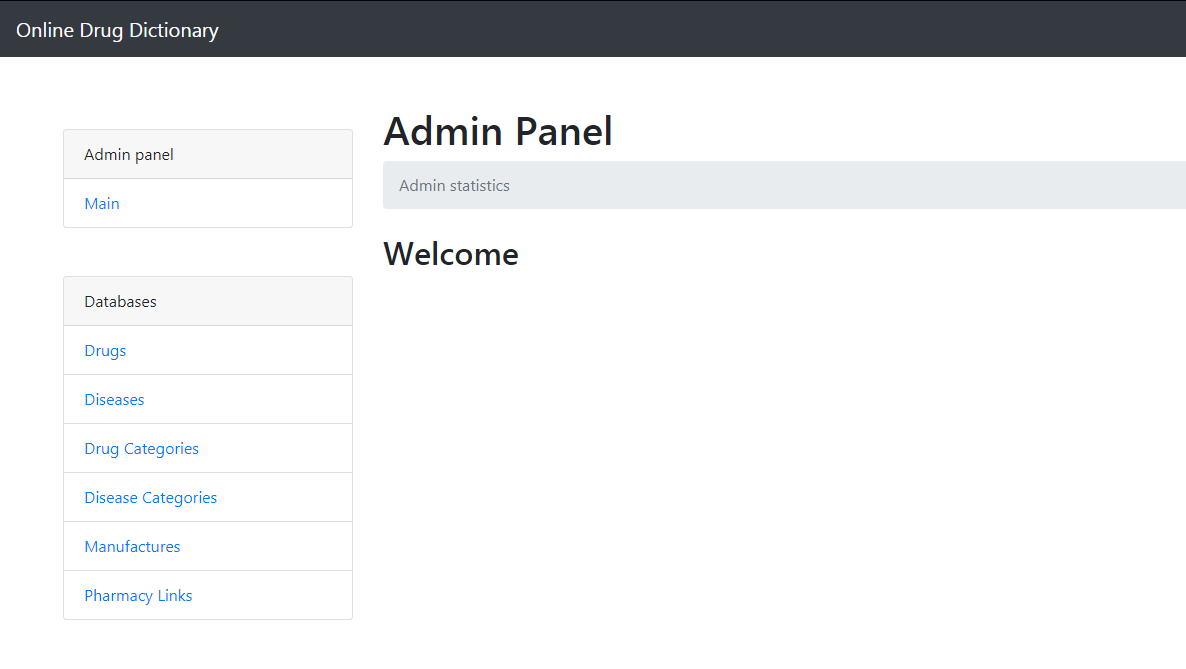


Figure 3.11 Example of admin page.

The team compiled a table of generated pages to measure page working. Each page is scored for parameters such as page rendering speed and the number of errors handled. Handled errors mean those errors that can occur when working with pages (Incorrectly entered data, an error when loading data).

Table 3.2 Pages render time and handled errors.

|  |  |  |
| --- | --- | --- |
| Name of page | Render time (ms) | Handled errors |
| Drugs | 100 | 18 |
| Diseases | 98 | 7 |
| Drug Categories | 106 | 3 |
| Diseases Categories | 116 | 3 |
| Manufacturers | 102 | 6 |
| Pharmacy Links | 107 | 3 |

3.4 Registration and login system

**Goal**: Create registration and login system of the project by using Laravel Facades and Middleware.

**Implementation process:**

Laravel framework has a built-in authentication system due to the fact that creating such system by yourself is time consuming and make security of the project more vulnerable to threats. When you create a Laravel project, by default, it create initial model and migration for users. This model consists of the attributes name, email, password, remember token and email verified. When registering, this model is used to store user data in the system. On login, Laravel creates a session stored in server files. To prevent unauthorized access to user data, Laravel checks if the session is valid both on client and server side.

Next, to use the authentication system in Laravel, there is a facade called Auth. It contains all the necessary functions for interacting with user data as well as with a session. Using the “*Auth::attempt($ credentials, $ rem)*” code, the framework will try to log in using the entered user credentials. Moreover, to check if the user is logged in “*Auth::user()*” code is can be used.

Now, to prevent the logged in user from entering the registration page, you can use Middleware. Middleware is used to filter HTTP requests based on request parameters. If a registered user tries to enter the registration page, Middleware will check if the user is logged in, and if so, it will send him to home page. To include Middleware in a controller, you need to use the constructor function of the controller class. In it we enter the code “$this->middleware([name of middleware])” to activate the selected Middleware.

To check the correctness of the entered data from the user on the registration or login form Validator facade is can be used. Validator checks the correctness of the entered data and existence of data in the POST or PUT request. If the request does not meet the requirements, then the controller sends the user to the same page where he entered the forms and displays an error message.

**Results:**

As a result, using these facades and Middleware, the team developed two controllers for user registration and login.



Figure 3.12 Register controller.



Figure 3.13 Login controller.

3.5 Presence drugs in shops

**Goal**: Create parsing system by using Zebra cURL.

**Implementation process:**

The project uses data from third party websites. In order to use this data, the project uses cURL technology. This technology allows you to extract HTML documents from websites. In the project, this technology allows to retrieve the prices and quantity of a product of a certain drugs on the web page of an online pharmacy. Unlike cURL, Zebra cURL is a more optimized version of this technology. This technology allows you to use multithreading and keep the cache of parsed web pages.

In addition, two built-in PHP class libraries DOMDocument and DOMXPath are used for parsing. DOMDocument translates extracted HTML documents into a special format that can later be used to extract specific elements of an HTML page. DOMXPath using the already translated page and extracts HTML elements according to various characteristics like tag, class, id, etc.

**Results:**

As a result, the development team created a parsing code from five online pharmacies.



Figure 3.14 Parsing function.

The development team compared the page load speed with and without cache:

* Loading speed without cache: 3.6 seconds.
* Loading speed with cache: 0.150 seconds.