

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Аналіз та розробка програмного забезпечення з відкритим кодом для
опрацювання та передачі телемедичних даних по каналу WiFi

Виконав(ла): студент(ка) 6 курсу, групи СНм-61
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Крот В.І.

(прізвище та ініціали)

Керівник

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Мацюк О.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Пастух О.А.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.

(підпис)

(прізвище та ініціали)

« »

20__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

студенту Кроту Вадиму Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз та розробка програмного забезпечення з відкритим кодом для опрацювання та передачі телемедичних даних по каналу WiFi

Керівник роботи Литвиненко Ярослав Володимирович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 29 » 10 2019 року № 4/7-949

2. Термін подання студентом завершеної роботи 29.06.2021

3. Вихідні дані до роботи Формати даних

4. Зміст роботи (перелік питань, які потрібно розробити)

Зміст. Вступ. 1.Огляд існуючих даних. 2.Робота з телемедичними даними. 3.Розробка програми на мові програмування C++. 4.Охорона праці та безпека в надзвичайних ситуаціях. Висновок. Перелік використаних джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Зміст. Вступ. 1.Огляд існуючих даних з телемедицини, завади в Україні та світова практика.

2. Робота з телемедичними даними. Історія та стандартизація. 3. Розробка прототипу програми для передачі телемедичних даних. 4. Охорона праці та безпека в надзвичайних ситуаціях. Висновок. Перелік використаних джерел. Додатки. Презентація (15 слайдів).

АНОТАЦІЯ

Аналіз та розробка програмного забезпечення з відкритим кодом для опрацювання та передачі телемедичних даних по каналу WiFi // Кваліфікаційна робота на здобуття освітнього ступеня «магістр» // Крот Вадим Ігорович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль 2021 // с.140, рис. – 24, бібліогр. – 52, додат. – 2.

Ключові слова: ТЕЛЕМЕДИЦИНА, ТЕЛЕМЕДИЧНІ СИСТЕМИ, ТЕЛЕМЕТРИЯ, ОНЛАЙН КОНСУЛЬТАЦІЇ.

Предметом даної роботи є телемедицина. Принципи впровадження, застосування та проблеми. Ціль роботи – дослідження існуючих телемедичних рішень, сучасних стандартів даних та створення прототипу власної телемедичної програми.

В першому розділі дипломної роботи розкрито поняття телемедицини, історія виникнення галузі, законодавство цієї галузі, теоретичні основи роботи, способи використання, недоліки телемедичних систем, проблеми впровадження, а також використання телемедичних мереж у різних країнах світу в тому числі в Україні. Також описано переваги застосування особливо в умовах світової пандемії. У другому розділі проаналізовано стандарти даних які використовуються в телемедичних системах і їх передача на великі відстані. У третьому розділі описано створену програму та її можливості застосування у телемедицині. Четвертий розділ присвячений питанням охорони праці.

У роботі представлені висновки по існуючим телемедичним технологіям. Проведений аналіз сучасних способів застосування телемедицини для допомоги пацієнтам. Надані рекомендації для покращення та розвитку телемедицини в Україні.

ANNOTATION

Analysis and development of open source software for processing and transmission of telemedicine data via WiFi // Qualification work for the degree "Master" // Krot Vadym Ihorovych // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and software engineering, Department of Computer Science, group SNm-61 // Ternopil 2021 // p.140, fig. - 24, bibliogr. - 52, appendix. - 2.

Key words: TELEMEDICINE, TELEMEDICAL SYSTEMS, TELEMETRY, ONLINE CONSULTATIONS.

The subject of this work is telemedicine. Principles of implementation, application and problems. The aim of the work is to study existing telemedicine solutions, modern data standards and create a prototype of our own telemedicine program.

The first section of the thesis reveals the concept of telemedicine, the history of the industry, legislation in this field, theoretical foundations, methods of use, shortcomings of telemedicine systems, problems of implementation and use of telemedicine in different countries, including Ukraine. The advantages of application especially in the conditions of a world pandemic are also described. The second section analyzes the data standards used in telemedicine systems and their transmission over long distances. The third section describes the created program and its possibilities of application in telemedicine. The fourth section is devoted to labor protection issues.

The paper presents conclusions on existing telemedicine technologies. The analysis of modern ways of application of telemedicine for the help to patients is carried out. Recommendations for the improvement and development of telemedicine in Ukraine are given.

ЗМІСТ

Вступ.....	7
1 Огляд існуючих даних.....	8
1.1 Поняття телемедицини.....	8
1.2 Історія появи та розвитку телемедицини.....	14
1.3 Аналіз складових частин телемедицини.....	15
1.4 Загрози для телемедицини та її розвитку в Україні.....	18
1.5 Законодавча база телемедицини в Україні.....	25
1.6 Існуючі телемедичні рішення в Україні.....	27
1.7 Телемедицина за кордоном.....	33
2Робота з телемедичними даними.....	37
2.1Типи телемедичних даних.....	37
2.2 Способи передачі даних.....	43
3 Розробка програми на мові програмування C++.....	47
3.1 Вибір мови програмування.....	47
3.2 Розробка програми.....	50
3.3 Експлуатація програми.....	54
4 Охорона праці та безпека в надзвичайних ситуаціях.....	60
4.1 Ергономіка робочого місця розробника програм.....	60
4.2 Основні вимоги щодо організації евакуації людей при пожежі на виробництві.....	64
Висновок.....	67
Перелік використаних джерел.....	68
Додатки	

ВСТУП

У великих містах з збільшенням кількості населення і кількості транспорту постає досить важлива проблема – дістатися до лікаря. Очікування у величезних чергах, нескінченні затори на дорогах в одну та іншу сторону, переповнений громадський транспорт – усе це змушує людей відкладати похід до лікаря, а тим часом хвороба може прогресувати, давати ускладнення і через деякий час хвороба яку можна було легко подолати антибіотиками розвивається і уже може знадобитися лікарське втручання, а у найгіршому випадку людину можна взагалі не врятувати. Іноді навіть швидка допомога не в змозі вибратись із затору і пацієнт досить довго очікує лікаря – при тому що можливо рахунок йде на хвилини.

Не найкраща ситуація і у жителів сіл. Невеликий медпункт і фельдшер – ось і вся охорона здоров'я у межах невеликих сіл. Тому іноді доводиться їхати у обласний центр і намагатись потрапити на прийом до спеціаліста. А такий шлях може зайняти декілька годин. А іноді з села потрібно буквально йти пішки, через жахливий стан доріг за межами міст допомога просто не може дістатись до пункту призначення.

У найгіршому положенні знаходяться люди із хронічними захворюваннями стан яких потрібно постійно контролювати. Таким людям необхідно самим слідкувати за своїм станом тому що не є можливим постійно знаходитись під наглядом лікаря (за виключенням перебування у стаціонарі).

Однак з розвитком технологій, у світі винайшли спосіб надавати набагато швидшу та більш кваліфіковану допомогу, а також допомогати жителям сіл і стежити за станом хронічно хворих. Одночасним вирішенням великої кількості проблем стала телемедицина. Будь яка людина що має комп'ютер та вихід в інтернет може зв'язатись з лікарем, а у селах фельдшер може обстежити хворого і одразу надіслати усі дані спеціалісту в профільній лікарні. Також лікарі можуть без тривалих та вартісних поїздок спілкуватись з закордонними колегами. Але навіть без закордонних спеціалістів можна поспілкуватись навіть з лікарями з інших областей або профільних клінік.

1. ОГЛЯД ІСНУЮЧИХ ДАНИХ

1.1 Поняття телемедицини

Телемедицина – це напрям медицини який надає медичну допомогу на відстані, на повну користуючись телекомунікаціями на сучасними комп'ютерними технологіями.

Телемедицина – це і ті працівники медицини які усі разом зібрались обговорити рідкісний та досить незвичайний випадок в такому форматі як онлайн консультація, роботизоване хірургічне втручання яким керує експерт з віддаленим контролем, і фізіотерапія яка тепер проводиться з віддаленим моніторингом та додатками для того щоб використовувати онлайн трансляції, а також безперервне надсилання даних щоб постійно знати в якому стані організм пацієнта.

Засоби телемедицини експерти поділяють на два різнопланові категорії залежно від способами звязку а також передачі інформації між лікарем та його пацієнтом. Асинхронна телемедицина не настільки популярна та передбачає спланований обмін попередньо уже записаними на відео в різний час в більшості між двома а інколи більше особами. Як от наприклад, клієнт, або мед працівник, який звернувся, відправляє опис усіх знайдених симптомів, і результати огляду якщо є необхідність, електронною поштою фаховому експерту, у відповідь спеціаліст має змогу відправити свій висновок щодо правильності діагнозу і оптимальності запланованого лікування. Практично це є обміном відео повідомленнями лише з тою різницею що проходить це не в онлайні, а записується в зручний час.

В своїй основі телемедицини покладені усі сучасні цифрові комунікаційні технології, з ними традиційний візит пацієнта до лікарні швидко перетворюється у аналог виклика лікаря в дім, але прийом уже був на екранах комп'ютера і\або планшета. Ідея відвідування лікаря віртуально розробляється уже впродовж декількох останніх років, але в Україні ринок подібних послуг почав зароджуватися і розвиватись недавно, а величезний сплеск так би мовити,

популярності телемедичних послуг стався під час карантину вірусу COVID-19, і зараз тільки збільшує обсяги допомоги.

Дистанційне консультування — сучасний засіб для будь якого фахівця первинної ланки, так і для уже кваліфікованих спеціалістів конкретних напрямків медицини. Введення послуг данної сфери у повсякденну клінічну діяльність дозволить розширити можливості які допомагають лікувати пацієнтів всюди, де є можливим підключення до мережі Інтернету. Одна з самих найбільших в Україні платформ що має можливість проведення та одразу ж реєстрації результатів проведених онлайн-прийомів і консультацій. Telemed24 не потребувала великих витрат для своєї експлуатації, досить легко інтегрується з загальною медичною інформаційною системою і не потрібно спеціальне спеціальне форматування, а також забезпечує хороший захист даних.

Є 7 загальних переваг телемедицини для персоналу лікарень і пацієнтів.

1. Безпека самого медичного персоналу від можливості зараження інфекційними хворобами.

Мінімізувати усі контакти з потенційно зараженими небезпечними хворобами пацієнтами можливо не лише за допомогою громіздких захисних костюмів, або протигазів, а також засобами проведення дистанційних онлайн консультувань в вигляді формату «лікар-пацієнт». У випадках з виникненням найперших симптомів, або ускладнень наявних захворювання пацієнт не потрібно буде відразу вирушати до лікарні для того щоб потрапити на прийом до сімейного лікаря, і доплати передавати віруси в дорозі іншим людям, можливо в транспорті а також в лікувальному закладі в черзі наприклад. Це грає велику роль в період пандемії вірусу COVID-19, сьогодні надто важливо обмежити всі контакти лікарів із хворими, а також обмежити відвідування лікарень для пацієнтів зокрема з ознаками ГРВІ, грипу, COVID-19, з різноманітними хронічними захворюваннями а також людьми з віком 60+ для запобігання можливого ризику зараження.

Зібрані наукові дані свідчать про те, що аж 80% інфікованих ковідом досить легко можуть перенести хворобу без серйозних ускладнень і тільки 5% мають необхідність госпіталізації пацієнта та втручання лікарів. Тому ці люди повинні якомога більше часу перебувати вдома і максимально використовувати

можливості дистанційної комунікації з лікарями а також спеціалістами вірусологами.

2. Розширює доступ людей до якісних та кваліфікованих медичних послуг і охоплює більшу кількість пацієнтів.

Онлайн-консультування покликане розширювати доступ усіх пацієнтів до медичної допомоги, якщо за певних причин в них нема бажання або за стану здоров'я залишати своє помешкання тільки заради відвідання лікаря. Крім того, є такий спосіб медичних послуг для звязку можна буде застосувати при допомозі пацієнтам що знаходяться у сільській місцевості, а також зазвичай у досить віддалених від обласних центрів населених пунктах і іноді в таких місцях доступ до кваліфікованої медичної допомоги банально відсутній.

3. Покращує всі можливі робочі процеси і значно підвищує загальну ефективність сучасної клінічної та реабілітаційної діяльності.

Впровадження медичних послуг з дистанційного онлайн консультування у великих закладах призначених для охорони здоров'я має на меті допомогу в більш швидкому зборі та використанні даних клієнта для того щоб більш швидкого та на порядок якісніше надання медичних послуг різної складності.

4. Новий рівень можливостей взаємодії з своїми пацієнтами за допомогою такого телемедичного інструменту як дистанційний моніторинг стану здоров'я.

Моніторинг стану здоров'я пацієнтів —телемедичний засіб що допомагає процесу надання якісної допомоги з використанням технологічних досягнень в галузі. Даний спосіб переніс більшу частину медичних послуг прямо до помешкання будь якого пацієнта, на заміну персональної присутності в кабінеті у лікаря. Технологія використовує деякі спеціальні пристрої, які займаються збором інформації про поточний стан здоров'я пацієнта, дані пересилаються в електронну мед карту та на персональний комп'ютер лікаря.

5. Збільшує дохід від практики.

Онлайн прийоми на українській телемедичній платформі під назвою Telemed24 – це хороша допомога для усієї лікарської спільноти. Тому що на медичній первинній ланці послуга буде сплачуватись НСЗУ по даним щомісячного звіту, в даній системі прийом хворого по інтернету буде позначатись

так само як звичайний фізичний, лише використовуються засоби по застосуванню відеозв'язку.

6. Знижує витрати пацієнта

Клієнти лікарів в наш час витрачають дуже великі грошові суми а також часу на відвідання та отримання консультації та допомоги лікаря на фізичному прийомі. Страшно уявити, як швидко можуть накопичуватися витрати тих громадян кому до лікаря потрібно їхати, догляд за малими дітьми або літніми\хворими родичами похилого віку і оформлення вихідних\відпусток\лікарняних. Додаткові і часто зайві зусилля витрачають пацієнти з територіально віддалених або важкодоступних населених пунктів. Для більшої ясності також розглядається досвід іноземних дослідників. Орегонський університет здоров'я і науки вирахував той факт що щороку заощадив більшості своїх пацієнтів 6,4 мільйонів доларів які йшли на дорожні витрати за рахунок запровадження в роботу телемедицини технологій. Під час опитувань майже половина пацієнтів запевнили що якби не можливість віртуального прийому то вони пропустили б діагностику та лікування

7. Підвищує лояльність пацієнтів.

Задоволеність послугами пацієнтів є одним із важливих основних показників підвищення ефективності при впровадженні телемедицини технологій в різних закладах призначених для охорони здоров'я. Лікарі, які уже мали час попрактикувати онлайн консультування, особливо ті що працювали на українській платформі Telemed24 розповідають, що більшість пацієнтів, для яких було запропоновано дистанційний формат звернення до мед закладу, після сеансу були задоволеними та планували за потреби звернутись за допомогою в такому форматі ще раз.

Де застосовується телемедицина?

Телемедицині технології мають широкий спектр використання і тому вони вже впроваджуються в педіатрії, психотерапія, дерматологія, неврологія або реаніматології (за таких ситуацій, коли людину з певних причини не можна або не має часу транспортувати до іншого клінічного закладу, відео дзвінок через інтернет компетентному лікарю певного фаху в багатьох випадках може

врятувати пацієнтові життя) але такі можливості можна застосувати майже в будь-яких медичних сферах.

Телемедицина надає важливу можливість пацієнтам почути думку лікаря не покидаючи власне житло. Це можна зробити, відправивши свою історію хвороби а також результати аналізів за необхідності, які зберігається в лікарняній інформаційній системі, до фахівця на зв'язку. Особливо це стає актуальним для онкохворих, які часто були вимушені добиратись за альтернативною думкою до іншого фахівця на прийом в інше місто або іноді якщо є можливість в іншу країну.

Сучасні технології та різні частини телемедичних систем також можуть та повинні допомагати рятувати населення в ураганах, повенях, землетрусах та інших надзвичайних ситуаціях техногенного та природного характеру. Коли потужний ураган «Харві» який наніс величезні збитки Техасу у 2017 році і там медики швидко встановили зв'язок різноманітні відеоконференції різних фахівців та допомагали лікарям що працювали безпосередньо на місці разом з фахівцями що перебували в інших штатах для тих людей, які зазнали шкоди від стихійного лиха. Це дало можливість в певній мірі розвантажити зайнятих пацієнтами з важкими травмами техаських лікарів.

Ще одним хорошим способом для використання телемедичних можливостей на повну стало розгортання дистанційних медичних пунктів. Для прикладу, американський ритейлер Walmart встановив у більшості магазинів своєї великої мережі пункти для того щоб вийти на дистанційну консультацію з лікарем, обладнані сучасними комп'ютерами. Пацієнт після того як введе наявні у нього симптоми, “відправляється” у таку собі віртуальну кімнату для очікування прийому і через кілька хвилин консультується через відео зв'язок з галузевим фахівцем.

Переклад з англomовного сайту <https://vsee.com> “Інструмент, який робить охорону здоров'я більш доступним, економічно вигідним та збільшує залучення пацієнтів - телемедицина. Починаючи з дебюту в кінці 1950-х, успіхи в телемедицині сприяли людям похилого віку. Крім того, пацієнти, які проживають

у сільській місцевості, які раніше мали труднощі з доступом до лікаря, тепер можуть проконсультуватись набагато простіше.”

Сфери де застосовується телемедицини

Дистанційна діагностика.

- Постановка правильного первинного діагнозу пацієнта дистанційно з допомогою оператором фахівця немедичної спеціалізації, або персоналу низького рівня кваліфікації.

- Підтвердження або зміна первинного діагнозу при консультації з більш досвідченим лікарем фахівцем, або профільним фахівцем в данній галузі.

Дистанційне консультування.

- Підтвердження або зміна діагнозу за участі профільного медичного центру, визначення процесу лікування та за потреби реабілітації, підтвердження показань до госпіталізації або ж дозвіл перебувати вдома.

- Консультації проводяться спеціалістами з клініки пацієнта, також його рідною мовою для того щоб покращити розуміння ситуації в обох сторін.

- Проведення іноді потрібного передопераційного обстеження в кваліфікованому медичному центрі під наглядом та керівництвом кваліфікованого фахівця певного медичного центру.

Віддалений моніторинг.

- Дистанційний моніторинг стану клієнта під час реабілітації та одразу після складних операцій.

- Первинний прийом, постановка діагнозу, визначення складності стану пацієнта без фізичного виїзду на місце.

Медицина катастроф

- Надання швидкої, кваліфікованої медичної консультації в місцях НС, сортування деяких пацієнтів, оцінка тяжкості ураження людей в жорстких часових рамках з непередбаченого для діагностики місця.

Дистанційне оперування

- Дистанційне проведення складних хірургічних операцій від хірургів високої кваліфікації.

Дистанційне навчання персоналу (лікарів)

- Дистанційні уроки та курси для підвищення кваліфікації медичного персоналу.

1.2 Історія появи та розвитку телемедицини

Офіційно такий новий термін як «телемедицина» використали вперше ще у далекому 1974 році. Але навіть на той час телемедицина – не виявилась новим явищем, як могло б здатися з перших поглядів. Така ідея як надання дистанційної медичної допомоги через відстані виникла доволі давно - ще століття тому. Саме у той час, винайшли і розпочали поширювати телефонні мережі, люди робили спроби передати наприклад, звучання тонів серця, сподіваючись що фахівець має змогу оцінити стан здоров'я саме таким чином.

У 1959 році один із канадських лікарів, вперше зміг передати відеозображення з місцевої лікарні на свою домашню станцію. Згодом стараннями цього лікаря створено одну з перших телемедичних систем, що зв'язала дві передові лікарні в Монреалі щоб почати здійснювати телерадіологічну діагностику.

У тому ж самому році у перший раз провели дистанційну демонстрацію пацієнтів з неврологічними захворюваннями спеціально для навчання студентам Психіатричного інституту штату Небраска. Але саме того року відбулась передача рентгенограми легенів муж країнами із Сполучених Штатів Америки до Канади з зв'язком через коаксіальний кабель було найяскравішим досягненням.

У 60-70-і роки активно створювалась дуже велика кількість нових систем а також і приладів телемедицини для покращення різних областей медицини. Але в більшості випадків це були всього лише експериментальні системи і пристрої, які отримували застосування лише в окремих випадках де у них була необхідність.

Для розвитку телемедицини ключовим моментом стала розробка мобільних телемедичних станцій, вони дали можливість надавати допомогу на великій відстані для віддалених та важкодоступних регіонів завдяки активному використанню сучасних супутникових технологій. Цей складний проєкт був

розроблений та організований NASA і йому надали назву STARPAHS (Space Technology Applied to Rural Papago Advanced Health Care, програма використання космічних технологій для забезпечення медичного обслуговування в сільських районах Папаго). Якісну медичну допомогу завдяки даному проєкту надали близько 4000 осіб, які без засобів телемедицини могли її не отримати взагалі.

Проте не дивлячись навіть на усі вищезазначені успіхи телемедичної сфери послуг, в другій половині 20 століття, справжній ріст галузі телемедицини трапився ближче до межі минулого століття і уже на початку нового тисячоліття. Цьому сприяв стрімкий та всебічний глобальний розвиток інформаційних технологій (як наслідок і телемедичних засобів) і запровадження ряду деяких міжнародних програм і документів, що почали регламентувати їх використання в галузі медицини.

1.3 Аналіз складових частин телемедицини

Телемедицина, як і будь який напрям звичайної медицини, складається з декількох частин. І як і у звичайній медицині, не завжди є можливість чітко визначити відмінність, де саме один з видів медичних процедур перейшов в зовсім інший. Далі буде представлено види телемедичних процедур.

Телемедичне консультування.

Найпоширенішою у наш час телемедичною процедурою є онлайн консультування, яке збільшує якість медичної допомоги за рахунок можливості вчасно її надати. Факт що для сучасного лікаря є надзвичайно важливим спілкування з кваліфікованими закордонними колегами через Інтернет, що також включає телеконсультування або конференції, які здійснюються без протоколювання, але з чітким дотриманням вимог безпеки.

Телемедичне консультування здійснюється насамперед на основі відео конференцій, з допомогою електронної пошти, та за використанням звичайних дзвінків. Тому ці телемедичні процедури настільки популярний у наш час, а у Україні особливо, це є хорошим знаком який сприяє подальшому розвитку телемедичної галузі в нашій країні.

Віддалені консультації і також діагностика пацієнтів являються найбільш поширеним сервісом. За допомогою спеціальних систем розроблених для відеоконференцзв'язку між лікарем, самим пацієнтом та можливо асистентом лікаря налагоджується повноцінний контакт з відео та аудіо, під час якого можна не тільки чути а ще й бачити один одного, також є можливість обмінюватися графічними а також текстовими даними онлайн.

Довгий час такий вид телемедицини позиціонувався засобом отримання медичних послуг в тих місцях, де немає поліклінік або конкретного спеціаліста.

Для прийняття найбільш раціонального, правильного та своєчасного клінічного рішення, лікар просто може зв'язатись з іншим спеціалістом та обговорити даний клінічний випадок в такому форматі як консультація "лікар-лікар". Підставами за яких необхідне проведення онлайн консультацій з лікарем можуть бути:

- Необхідність швидкого встановлення або ж підтвердження правильного діагнозу;
- визначення та відносно швидке підтвердження тактики лікування;
- діагностування при виникненні важких та досить рідкісних захворювань або захворювань з незвичайним атипичним перебігом;
- визначення методів для того щоб попередити ускладнення;
- розгляду популярних скарг;
- скорочення діагностичних та терапевтичних витрат що витрачаються на пацієнта при цьому не знижуючи їх якість та ефективність за допомогою засобів телемедицини;
- отримання консультації в випадках географічної віддаленості окремих фахівців, необхідних для обстеження;

Біотелеметрія (телемоніторинг)

Телеметрія — технічні засоби і методи для вимірювання на відстані різних параметрів організму. Передавання телемедичних даних з місця де проведений огляд на комп'ютер лікаря.

Телеметрія у контексті телемедицини виникла для потреб космічної медицини; а різновид — телемоніторинг — застосовується в відділеннях інтенсивної терапії а також для транспортування важких пацієнтів. Варто зазначити що телеметрія в основному застосовується в військовій, і навіть аерокосмічній медицині як і в медицині катастроф. Найпоширенішою на даний момент формою клінічного застосування телеметричних систем є теле-ЕКГ.

Домашня (індивідуальна) телемедицина

Навіть порівняно з іншими телемедичними системами, домашня або індивідуальна телемедицина ще зовсім молода галузь. Телесестринство, телеконференції, телеконсультації, а також складові частини біотелеметрії. Домашня телемедицина включає елементи цих напрямків.

Насамперед це процедура для людей, які через суспільне положення або проблеми із здоров'ям не мають бажання, або не хотіли б їздити а також знаходитись у лікарні. Політики, знаменитості, тяжко хворі з хронічними хворобами. Трохи інший випадок люди похилого віку, якимби не хотілось провести решту життя в лікарні, краще вдома у сімейному колі.

Телескринінг

Телескринінг забезпечує широкий спектр заходів, для покращення здоров'я дітей а також підлітків. Раннє виявлення онкологічної а також фтизіатричної патології.

Ця процедура має на меті заходи для покращення рівня загальної медичної допомоги. Скринінг направлено на виявлення і діагностування складних захворювань у перших стадіях.

Дана процедура працює за допомогою апаратного комплексу. Окрім обладнання, також телескринінг потребує ПЗ. Перша - БД, де зберігається необхідна медична інформація. Друга - програми, які проводять вибірку рядом ознак.

Дистанційне навчання

Дистанційне навчання — сукупність сучасних технологій, що подають інформацію за використання ІКТ від викладачів, великих постатей у певних медичних галузях, до студентів та інших слухачів. Основними принципами такого

виду дистанційного навчання є взаємодія, а також надання студентам можливості самостійного освоєння досліджуваного матеріалу.

У телемедицині дистанційне навчання це хороший спосіб підготовки персоналу до більш складних процедур які можуть бути у подальшому у подальшому.

1.4 Загрози для телемедицини та її розвитку в Україні

Телемедицина з часу започаткування пройшла досить довгий шлях, і допоки технології так стрімко розвиваються телемедицина також повинна використовувати в своїх цілях усе нові та нові засоби. Але сперше варто озирнутись на минуле телемедицини.

Всупереч достатньо поширеній думці, телемедицина не є новою практикою. Насправді перші спроби телемедицини датуються аж 19 століттям! Це розпочалося тоді, коли декілька лікарень, , стали інтеграційною системою для надання в віддалених місцях постійної допомоги.

Телемедицина в 19 столітті.

Створення телемедицини розпочалося з того часу коли сталось створення телекомунікаційної інфраструктури, де був телефон, радіо та телеграф. Це одне із перших випадків застосування телемедичної технології.

До 1879 р. У доповіді Ланце було про те, як використання телефону може суттєво зменшити кількість непотрібних відвідувань лікарні.

Телемедицина в 20 столітті.

У 1922 р. Гюго Гернсбак зміг представити теледактил у одному науковому журналі. Гернсбак передбачив, що з допомогою цього сенсорного пристрою зворотного зв'язку можна буде дозволити лікарям бачити пацієнта з допомогою ч телевізійного екрану і навіть торкатися з допомогою спеціального робота.

Телемедицина сьогодні.

Сьогодні більшість людей мають доступ до основних телемедичних пристроїв, таких як мобільні телефони, планшети та комп'ютери. Також нові розумні годинники в більшості можуть зафіксувати пульс, серцевий ритм, а

деякі навіть тиск. З покращеним доступом люди з сільської місцевості та зайнятих міських районів можуть легко зв'язатись з тими хто постачає послуги. Медичні прилади призначені для домашнього використання зробили можливим доглядачам контролювати практично все, як от життєво важливі органи і навіть рівень глюкози. Лікарі можуть зібрати абсолютно всю необхідну їм медичну інформацію і після цього поставити діагноз.

До 2020 року телемедицина, стане необхідною складовою частиною сучасних медичних послуг.

Перспективи телемедицини в Україні.

Більшість пацієнтів зможуть отримати доступ до набагато розширеного списку медичних послуг, час який очікуєш перед онлайн-прийомом у спеціаліста буде скорочено. Проведені опитування показали зокрема те, що середній час, який людина витратила на відвідування лікаря, становив приблизно дві години. У той же час онлайн-прийом у вигляді консультації пацієнта триває у більшості випадків не більше 20 хвилин.

Передбачається, що онлайн телемедична і офлайн, тобто паперова база у лікарнях об'єднуються, і таким чином інформація зібрана про стан здоров'я та проблеми людей буде надалі зберігатися у цифровому вигляді на ПК. Україна уже проходить цей складний шлях оцифрування даних, і eHealth уже надає лікарям можливість за лічені секунди отримувати необхідну медичну інформацію, і надалі коректно поставити діагноз з появою повного розуміння стану здоров'я пацієнта.

З поширенням різноманітних технологій як от смартфони і планшети, а також вебкамери для ПК телемедицина зайняла своє місце і сприймається звичною та продуктивною практикою серед професійного медичного товариства України. Вона надала доволі багато додаткових можливостей. Наприклад немає потреби їхати, а потім і провести купу часу в черзі, щоб дізнатись про необхідність повторно пройти огляд через місяць, пацієнт надсилає лікарю результати усіх проведених аналізів через месенджер.

Основною умовою при проведенні телемедичного сеансу – це захистити персональні дані пацієнта. Якщо поглянути на це з точки зору закону то у ЗУ «Основи законодавства України про охорону здоров'я» так звана лікарська

таємниця є конфіденційною інформацією і вона суворо охороняється Законом і тому не може бути розголошена. Повноцінна телемедична консультація передбачає обов'язкову наявність у медичних закладах спеціалізованого технічного обладнання. У даний час головні медичні установи в Україні поступово отримують необхідне обладнання для того щоб була можливість проводити телемедичні консультації. З допомогою спеціального мобільного діагностичного комплексу, інформація автоматично відсилається в хмару а також електронну картку кожного пацієнта, до Електронної інформаційної системи eHealth.

Наступним етапом стане впровадження в регіоні єдиної телемедичної платформи з дистанційним доступом. Цей веб-інтерфейс, не є складним у процесі експлуатації, це спеціальна сторінка, яку можна відкрити з всіх браузерів. Повідомлення про надходження запиту на телеконсультацію приходять лікарю, який зареєстрований в системі, на електронну пошту.

Активний розвиток для телемедичних систем в нашій державі розпочався досить пізно у порівнянні з іншими країнами, на початку 2000-х. Розпочали в Одесі, потім Дніпропетровську та Харкові. Пізніше, різноманітні телемедичні мережі розпочали проектувати та будувати свої клініки з підтримкою послуг через інтернет. Але тоді це все відбувалося за ініціативи мед закладів які розуміли користь телемедицини.

Вперше в Україні заговорили про телемедичні системи на рівні законів держави у 2017 році. Саме тоді й розпочали проектувати базу законів для використання телемедицини.

Важливим кроком для розвитку телемедицини по Україні є професійна мережа для телемедичного зв'язку Medinet. Всеукраїнська телемедична платформа стала першою що функціонувала в Одеській області у 2019 році. Понад 10 тис. телемедичних онлайн консультацій за менше року роботи провели лікарі, що працювали в даній мережі.

На Кіровоградщині уже у більш ніж 18 великих медичних закладах розпочали активне використання телемедичних систем, зокрема медичну інформаційну систему РІМС «Медстар» а також IDIS – спеціальний

діагностичний комплекс, який дає можливість спеціалісту отримати дані про здоров'я людини і додати їх до електронної мед карти. Сімейні лікарі мають змогу інформацію відправити консультанту.

Усім пацієнтам було надано простий спосіб доступу до кваліфікованої та своєчасної медичної консультації. Без особистої присутності у обласному або міському закладі, вони змогли отримати корисні поради від спеціалістів своїх.

У сучасному світі різноманіття видів захворювань та їх штамів, робить суттєво складнішим процес постановки точного та правильного діагнозу для усіх своїх пацієнтів. З використанням інноваційним технологіям цей процес істотно спрощується, оскільки консультація спеціаліста у віддаленому режимі є можливою. При проведенні консультацій лікарі отримують нові знання, вивчають різні симптоми та ситуації та вдосконалюють навички, це йде на користь розвитку медицини України. В Україні пацієнтам особливо важливо для важкодоступних та віддалених районів особливо важлива можливість дистанційного нагляду лікаря, де спостерігається дефіцит висококваліфікованих лікарів та немає нового обладнання.

Консультації висококваліфікованих лікарів досить важкодоступні. Це відомо абсолютно кожному пацієнту, який намагався потрапляти на консультації до хорошого спеціаліста – записуватись мінімум за місяць. Але тепер завдяки можливості консультування в онлайн режимі це стало значно простішим та доступнішим. Медики через відео конференції мають змогу обмінюватись досвідом між містами, а також навчати перспективних молодих спеціалістів, без залежності від територіального місцезнаходження. Телемедицина надає також можливість в прямому ефірі показувати хід складних операцій.

Плюси відеоконференцій в медичній системі –це збільшення швидкості діагностики, а також більша економічна ефективність. Пацієнти не будуть оформляти вихідний та витратити власний час та гроші на дорогу з іншого кінця міста або навіть з регіонів якщо необхідно дістатись до великих обласних медичних центрів.

Цілі і завдання.

Проект має на меті підвищення якості та правильності діагностики а також виявлення на початкових стадіях важких захворювань та доступність мед. допомоги високої якості.

Проект «Телемедицина» має на меті обладнання та включення у систему різних закладів медицини міст діяльності ДТЕК для сполучення усіх даних в одній мережі і забезпечити швидкість та доступності, якості та повної безпеки медичної допомоги для усього населення за потреби та незалежно від територіального дислокування пацієнта.

Заходи по реалізації:

- Для зв'язку процес побудови оптоволоконної лінії до лікувальних закладів;
- Будівництво внутрішньої мережі лікарень.
- Закупівля та встановлення з тестуванням необхідного телемедичного устаткування;
- Навчання лікарів для роботи з системою телемедицини: організація участі лікарів конференціях, семінарах, а також майстер-класах;
- Організація надефективної експлуатації телемедичних систем в лікувально-профілактичних установах;

Результати та перспективи:

У 2012-2013 роках національна телемедична мережа завдяки різноплановій підтримці ДТЕК була значно розширена. Мережу лікарень розширили на 9 медичних установ включаючи дві великі лікарні у Львові.

Для реалізації проекту телемедицини в лікарнях включених у мережу виконували поставлені задачі:

- побудована внутрішня телемедична мережа в медичних закладах, налагоджене обладнання та програмне забезпечення, і проведено підключення до мережі запланованої кількості робочих станцій, серверів та іншого необхідного мережевого обладнання. Закупівля необхідного обладнання потрібного лікарням, з можливістю оцифрування та передачі даних прямо з апарату;

- було проведено використання системи у тестовому режимі і забезпечення налагодження проблем якщо було виявлено недолік в разі потреби;

- проведено навчання персоналу, лікарів та консультантів, з користування системою телемедицини для використання технічної сторони, медичний персонал приймав участь в семінарах з експлуатації, конференціях, тренінгах;

Згодом у 2014 році було підключено 3 лікарні які були готові і мали преднатись до телемедичної мережі:

- КЗ «Тернівська центральна міська лікарня»;
- КЗ «Першотравенська центральна міська лікарня»;
- КЗ «Зеленодольський центр первинної медико-санітарної допомоги»;

Організовані навчання призначені для підвищення кваліфікаційного рівня медичного персоналу для лікарень ДТЕК дали змогу лікарям користуватись перевагами проекту, і також були проведені:

- телемедичні консультації лікарів у закордонних клініках – 3;
- наукові конференції різних рівнів – 6;
- тренінги, семінари для лікарів асистентів та студентів– 40;

А тепер перейдемо до проблем.

Інтернет-покриття і технічне обладнання

Найбільше шкодить розвитку онлайн-консультацій нестабільне підключенням до Інтернету навіть у деяких областях, не кажучи вже про з'єднання за межами великих міст. Взагалі для телемедицини не завжди принципове використання відеозв'язку — іноді може вистачити простого телефонного дзвінка. Але і навіть стабільне 3G-покриття є далеко по Україні вдається знайти далеко не завжди.

2019 року було проведено анкетування серед 1108 лікарів, яким довелося працювати у містах з не великою чисельністю населення - до 100 тис. осіб у 5 областях, серед яких такі великі області як Харківська та Дніпропетровська. Дане опитування показало що впровадити телемедицину в Україні можливо. Насправді, більшість із опитаних лікарів мають ПК вдома і також на роботі, а також мають можливість користуватись інтернетом на роботі.

Але перешкоди є:

- Сбої роботи інтернету можуть призвести до нестабільності зв'язку та втрати пакетів;
- низька якість апаратів дистанційного діагностування що ускладнює роботу;
- несумісність програмного забезпечення і як наслідок можливі збої обладнання або компонентів систем;

Також проблемою можуть стати самі люди. Медиків потрібно навчити правил користування сучасною технікою та програмним забезпеченням. Особливо важко буде з старшою категорією лікарів. Як показує практика навчити їх швидкому та правильному поводженню з комп'ютерами буває проблематично. А пацієнтам необхідно буде показати та пояснити переваги телемедицини. Як економічні, так і з точки зору швидкості надання допомоги..

Правова база.

Правові питання заважають телемедицині не тільки в Україні. Це і недостатність законів та стандартів за якими будуть надавати телемедичні послуги. І недоліки законів у сфері захисту особистої інформації та приватного життя пацієнтів. Поки що для практики віддаленої діагностики відсутня законодавча підтримка.

В Україні є лише наказ МОЗ від 26.03.2010 р. №261 «Про впровадження телемедицини в закладах охорони здоров'я» та від 19.10.2015 р. А також більш новий документ №681 «Про затвердження нормативних документів щодо застосування телемедицини у сфері охорони здоров'я».

Єдина для України платформа для онлайн-консультування попередньо задекларованих пацієнтів, що може передавати дані до Електронної телемедичної системи охорони здоров'я — Telemed24.

Також апаратна частина має бути стійка до перепадів наруги та вимкнення електроенергії, хоч у більшості лікарень і є аварійні генератори різка зміна наруги може пошкодити апарати і призвести до втрати даних.

1.5 Законодавча база телемедицини в Україні

Позитивним кроком до телемедицини став старт медичної реформи, яка розпочала швидкими темпами інформатизовувати медицину.

Другий – початок проекту в ході якого буде проведено тестування телемедичних технологій у сільських амбулаторій областей у яких уже є телемедична мережа. який Будуть тестувати консультації що до серцево судинних хвороб, бронхіальна астма, дерматологічні захворювання і цукровий діабет.

В Україні починають активно впроваджуватись телемедичні технології, а це в свою чергу робить необхідним розробку нових законопроектів для захисту як споживачів послуг (людей яким потрібна медична допомога), так і тих хто ці послуги буде надавати (лікарні у яких використовують засоби телемедицини).

Створенням цих законів займається Міністерство охорони здоров'я України. Закони повинні захищати лікарів які чітко та правильно визначають діагнози і цим самим допомагають швидше одужати або зупинити розвиток хвороби, захищати пацієнтів у випадку некомпетентності лікаря, а також захист документації, яка курсує по телемедичній мережі.

Як от наприклад наказ “Про затвердження нормативних документів щодо застосування телемедицини у сфері охорони здоров'я” 2015 року. З повним змістом документу можна ознайомитись на сайті <https://zakon.rada.gov.ua/laws/show/z1400-15#Text>.

У даному наказі є практично все що стосується телемедицини:

1. Організація допомоги із застосуванням телемедицини
2. Мета і завдання телемедицини
3. Форми облікової документації
4. З чого складається телемедична мережа
5. Використання інструментів та засобів телемедицини
6. Ресурсне забезпечення телемедицини
7. Правила проведення телемедичного консультування

Загалом у всіх цих правилах описано досить детально усі інструкції для проведення будь яких телемедичних процедур, і необхідне для їх виконання обладнання. Також вказано необхідний рівень кваліфікації лікарів та операторів що забезпечують з'єднання лікарів з пацієнтами, або з членами бригади швидкої допомоги.

Що таке телемедицина та як вона організована на території Вінниччини?

В області було представлено нову систему з можливістю доступу до допомоги лікаря на відстані.

«Сьогодні я ініціював презентацію телемедичної системи, яка вже впроваджується в Одеській і Харківській областях ,– коментує голова ОДА Владислав Скальський. – Ми зібрали в Департаменті охорони здоров'я облдержадміністрації усіх координаторів і відповідальних працівників, які забезпечують функціонування первинної ланки для первинного знайомства з платформою. Завдання – побачити як це працює і отримати вичерпні відповіді на хвилюючі запитання від розробників системи».

Голова ОДА запевнив, що телемедична консультація на Вінниччині запрацює вже зовсім скоро: «Наразі ми маємо вже працюючу систему, а телемедична мережа – це реальна можливість справді підняти на новий рівень якість надання медичних послуг на території області навіть у найвіддаленіших селах – у ФАПх, ЦПМСД. Тож очікуємо технічні умови, потім необхідно буде випробувати демо-версію програми, яка буде налаштована під нашу реальність і тільки після цього ми вже разом із Департаментом охорони здоров'я, виходячи із можливостей фінансування, будемо остаточно визначатися щодо запровадження системи телемедицини на території всієї області».

У впровадженні нової медичної реформи в Україні, Вінниччина стала учасником першого в своєму роді проекту «Телемедицина». І основна мета цього проекту проста схема отримання допомоги від найкращих обласних спеціалістів в усіх куточках країни.

1.6 Існуючі телемедичні рішення в Україні

Є такі основні види взаємодії в телемедичних системах: «лікар-лікар» а також «лікар-пацієнт». Перший уже давно і активно використовується на території України, в той час як перед другим видом постали досить серйозні виклики.

Ринок цих послуг на території нашої країни знаходиться на проведенні етапу становлення, але значне поживлення припало на карантин що досить логічно. Лікарі отримали досвід спілкування з тими хто заражений COVID-19 які власними силами лікувалися вдома. Деякі сучасні телемедичні сервіси пропонують безкоштовні консультації по інтернету на зразок наша компанія “Medstar Solutions LLC” яка у використанні надає Telemed24 для телеконсультацій на безоплатній основі

Сегменти за типом доставки телемедичних послуг:

- Веб-браузерні або мобільні додатки (що бувають у вигляді передбачати телефонний, відеозв’язок та/або текстові чати);
- Колл-центри (телефонні консультації).

Найпопулярніші послуги в Україні — телемедичні платформи для консультацій, оскільки телемедичне консультування проста й зрозміла усім функція і є компонентом лікувально-діагностичної повсякчасної роботи.

- отримання висновку спеціаліста в електронному вигляді (54%);
- телеконсультування онлайн, тобто в режимі в режимі реального часу у присутності пацієнта проякого буде йтись у розмові (28%);
- отримання експертних порад у асинхронному режимі (18%).

Найважливішим фактором для зростання галузі у 2020 році стала як не дивно пандемія і те як запобігти зараження. Телемедичні технології відкрили можливість людям, які відчувають або ще тільки підозрюють захворювання, не їхати до лікаря і отримувати допомогу віддалено. І лікарі, і пацієнти є зацікавленими сторонами у дистанційних засобах зв'язку.

Стримує розвиток ринку і відсутність ефективних економічних моделей діяльності. По суті, сьогодні немає обґрунтованих підходів для оцінки

економічної доцільності телемедицини “лікар-пацієнт”, заснованих на практиці телеконсультацій в українських реаліях. Але зрозуміло, що підключення за допомогою телемедицини декількох віддалених об’єктів до загальної мережі може виявитися більш рентабельним способом надання медичної допомоги в порівнянні з будівництвом нових об’єктів і наймом медичного персоналу.

Також, експерти з права відмічають недостатність правової бази й стандартів, що дозволяють надавати телемедичні послуги.

Одними з ментальних перешкод впровадження телемедицини є суто людські і культурні фактори. Деякі пацієнти і медичні працівники чинять опір застосуванню сучасних цифрових моделей лікування, які відрізняються від традиційних підходів або місцевої практики.

Незважаючи на це, розвиток та становлення телемедичних послуг — довготривалий шлях, який, без сумніву, ще належить пройти Україні.

Однозначно, ринок телемедицини України перебуває ще на етапі зародження. Проте на сьогодні вже існують окремі рішення, що успішно пройшли перші кроки впровадження у повсякденну у практику лікарів та випробувані пацієнтами.

Основні гравці ринку телемедицини.

Пандемія COVID-19 змусила поглянути на телемедицину під новим кутом і в значній мірі прискорити її впровадження по всьому світу. Зокрема і в Україні. Так, буквально за останній рік українські компанії-постачальники медичного програмного забезпечення значно активізувалися та змогли випустити свої телемедичні продукти на ринок.

Серед основних гравців українського ринку виділяємо:

- Телемедична платформа Telemed24 РІМС "Медстар". Розробник — компанія Medstar Solutions LLC;
- Телемедична мережа Medinet. Розробник — компанія IT4Medicine;
- Консультації лікаря по телефону Doc.ua. Розробник — ТОВ "ДОКЛАБ";
- Платформа для онлайн зв'язку з лікарем Medikit. Розробник — ТОВ "МЕДІКІТ";

- Телемедичний сервіс для пацієнтів Doctor Online. Розробники — IT4Medicine та Kyivstar.

Telemed24 — телемедична платформа, призначена для проведення консультацій та реєстрації результатів дистанційних прийомів. Telemed24 є модулем, інтегрованим до регіональної інформаційної медичної системи “Медстар”. Усі дані прийому фіксуються в електронній карті пацієнта і передаються до державної системи eHealth. Інформація пацієнтів захищена згідно з вимогами КСЗІ, а телемедичні прийоми є офіційними і враховуються в робочий час лікарів.

Пацієнти ж можуть отримати телемедичну консультацію Telemed24 через безкоштовний мобільний додаток MedCard24. У ньому необхідно знайти заклад і свого лікаря, обрати день/час прийому та зачекати на відеодзвінок доктора. Також є і веб-версія Medcard24, де можна не просто потрапити на дистанційний прийом, а ще й придбати медикаменти згідно з призначеннями лікаря.

“Огляд пацієнта відбувається майже так само, як і на звичайному прийомі. Проте є можливість не йти до лікарні особисто. Пацієнт обирає лікаря, записується на конкретну дату та час й отримує відеодзвінок від фахівця. Далі все, як на звичайному прийомі — пацієнт пояснює причину звернення, лікар фіксує скарги та надає рекомендації,” — розповів журналістам "ШОТАМ" Михайло Торченко, експерт з побудови телемедичних сервісів та мереж “Medstar Solutions LLC”.

З модулем телемедицини Telemed24 лікар може:

- провести телемедичне консультування у форматі відеочату або телефонного зв'язку;
- призначити лікувальні заходи і виписати електронний рецепт на медичні препарати;
- направити пацієнта на діагностику чи аналізи;
- налаштувати календар спостереження за показниками здоров'я пацієнта;
- зафіксувати в електронній медичній картці та передати епізод в eHealth;

- дистанційно оформити декларацію з пацієнтом, якому необхідна медична допомога.

Постачальник спорядження — компанія “Medstar Solutions LLC”, потужний інтегратор медичних технологій, обладнання та програмного забезпечення в Україні.

Ціна: телеконсультації через платформу безкоштовні для пацієнтів і лікарів. Лікарі ж, зокрема, зацікавлені проводити онлайн-прийоми через Telemed24, адже вони мають змогу отримати компенсацію за кожну таку консультацію, як за звичайний прийом, від Національної Служби Здоров'я України (НСЗУ) в межах Програми Медичних Гарантій.

Телемедична мережа Medinet

Medinet — це професійна телемедична мережа, платформа для взаємодії всіх ланок медичної допомоги на рівні міста чи області. Запуск платформи Medinet відбувався в 2019 році в Одеській області, що стало базою відкриття першого в Україні обласного центру телемедицини на базі Одеської обласної клінічної лікарні.

Сервіс працює у двох форматах: «лікар-лікар» та «лікар-пацієнт». Але головний фокус системи — взаємодія між лікарями, котрі підключені до телемедичної мережі, що вирізняє Medinet від інших телемедичних рішень на ринку України. Система забезпечує зв'язок між лікарями первинної медичної допомоги зі спеціалістами вузького профілю: «фельдшер ⇔ сімейний лікар ⇔ регіональний медичний центр». Робота телемедичної мережі Medinet базується на технічних рішеннях групи компаній IT4Medicine.

Ціна: сервіс безкоштовний для закладів охорони здоров'я та пацієнтів.

Doctor Online від Київстар.

Doctor Online — сервіс цілодобових консультацій лікарів. У сервісі працює понад 50 лікарів. Завдяки співпраці з мережею лабораторних центрів “Сінево” та системою замовлення і доставки ліків “Liki24”, пацієнт має змогу після отриманої консультації оплатити ліки та аналізи в додатку зі своєї банківської картки.

Консультація можлива у чаті, через відео або телефоном. Сервіс створений спільно групою компаній IT4Medicine та оператором Kyivstar.

Ціна: вартість консультації на 10 хв. — 50-70 грн.

Medikit — медична комунікаційна платформа, що працює у форматі онлайн-консультацій «лікар-пацієнт» через чат, відео або аудіозв'язок. За допомогою платформи користувачі можуть ставити питання щодо здоров'я та отримувати консультації від лікарів. На платформі працюють фахівці різних спеціалізацій. Зареєструвавшись, лікар може консультувати пацієнтів та отримувати безготівкову оплату за надані послуги.

Ціна: вартість миттєвої онлайн-консультації для пацієнта — 150 грн., підписка на рік — 590 грн., прийом у лікаря чи виклик лікаря на дім за попереднім записом — за ціною лікаря-виконавця.

Які на типи телеконсультацій є попит?

Як зазначає у своєму дослідженні ринку М. Волковська, через платформу Doctor Online близько 80% запитів — це допомога педіатрів та лікарів загального профілю. Ще один популярний напрямок — дерматологія. Так, найбільш затребувані онлайн-консультації, за даними платформи Medikit і клініки Добробут, — від терапевтів та дерматологів. Великим попитом також користуються повторні консультації вузьких фахівців, наприклад, якщо пацієнт був на консультації, здав аналізи, і, щоб повторно не йти в клініку, зв'язується з лікарем онлайн.

На сьогодні через платформу Telemed24 пацієнти найчастіше звертаються до лікарів сімейної медицини, рідше — до інших фахівців. Найчастішими причинами для проведення телемедичних консультацій з дорослими є:

- гострі респіраторні захворювання (як вірусні, так і інфекційні), а також бронхіт / бронхіоліт, грип (понад 38%);
- шкірні висипання (алергії, акне, герпес і т.д.) (21%);
- захворювання сечовивідних шляхів (11%);
- хвороби систем кровообігу (18%) тощо.

Серед дитячих консультацій найбільше звернень щодо висипань на шкірі, ГРЗ, лихоманки і порушення шлунково-кишкового тракту. І звичайно багато

звернень щодо симптомів інфекції COVID-19, спричиненої коронавірусом SARS-Cov-2.

Перспективи ринку телемедицини в Україні

В Україні телемедицина як загальнодоступна медична практика тільки починає розвиватися. Хоча для нашої країни з великими відстанями і транспортною нерозвиненістю телемедичні послуги мають особливо важливе соціальне і економічне значення.

Станом на серпень 2020 року, в Україні щодня фіксуються антирекордні кількості заражених COVID-19. На думку експертів, придбаний за минулі місяці досвід використання телемедицини "лікар-пацієнт" став поштовхом для розвитку цієї медичної галузі. І медичні організації, і пацієнти, в екстремальних умовах змушені були перейти до такої форми спілкування онлайн. І багатьом вона припала до душі.

Ринок буде поступово долати скептицизм і недовіру як пацієнтів, так і лікарів. З досвіду інших країн, на початку свого шляху телемедицина завжди орієнтувалась на досить вузький сегмент цільової аудиторії "ранніх послідовників нових технологій", але вже через 5-10 років ринок може дозріти до більш масштабного впровадження систем по всій території країни.

Через 2 роки фахівці очікують зростання ринку телемедицини України на 10-20%.

Для розвитку телемедицини в Україні варто зосередитися на 4 напрямках:

1. доступності стабільного інтернет з'єднання в будь-якій точці країни;
2. вдосконалення правової бази;
3. навчання медичного персоналу та населення роботі з комп'ютером;
4. залучення до консультацій висококласних лікарів та підвищення довіри до телеконсультацій.

Чи стане телемедицина широкою популярною серед населення в Україні? У поліклініки часто звертаються негативно налаштовані люди, що з недовірою ставляться до нових технологій. Експерти вважають, що довіру пацієнтів до телемедицини можна завоювати, показуючи на реальних прикладах, що в онлайн-консультаціях є цінність, і реально дистанційно отримати компетентні відповіді

на питання про стан здоров'я. Чим частіше люди будуть отримувати якісну допомогу від висококласних лікарів, тим швидше телемедицина стане гідною альтернативою пошуку симптомів в інтернеті.

Деякі проблеми ринку можна вирішити шляхом збагачення досвіду використання ІТ-технологій. Отримання державної та інституційної підтримки може бути вирішальним фактором для забезпечення успіху телемедицини в Україні.

1.7 Телемедицина за кордоном

Якщо у нас телемедичні технології й досі вважаються новим напрямком в наданні допомоги, то для багатьох провідних країн світу – це уже давно звичний метод медичної допомоги, що уже досить багато років експлуатації позитивно представив себе серед лікарів та пацієнтів.

Першими країнами, які оцінили зручність та інші переваги телемедицини, були такі країни як Норвегія та Фінляндія. Саме у цих країнах великі засніжені відстані між лікарями і пацієнтами є завадою, а тому мають ключове значення і сприяли розвитку цієї галузі. Тому впровадження телемедичних сервісів не змусило себе чекати і стало обов'язковою вимогою щоб була можливість надання якісних і вчасних медичних послуг мешканцям важкодоступних та віддалених місцевостей.

Потім новий напрямок медичних послуг підхопили в США та декількох європейських країнах. В США телемедичні можливості спочатку використовували тільки військові, а уже потім стали дозволяти застосовувати і у цивільному житті. А Китаї для цих потреб навіть розробили унікальний діагностичний пристрій, який швидко інтегрується з планшетом пацієнта. Даний пристрій фіксує найважливіші фізіологічні параметри людського організму та автоматично надсилає їх особистому лікарю пацієнта. За результатами отриманими в ході діагностики лікар і робить висновки і надає віддалену консультацію.

Важливий акцент в розвитку цієї галузі йде на законодавство держави і розвиток відповідної до законів інфраструктури. Це в основному важливо для різних страхових компаній, які покривають усі ці витрати, і що найголовніше дешевше для пацієнта. Тут телемедичні засоби та послуги вже інтегровані а також закладені в різноманітні страхові пакети.

Зараз у США скупчено практично половина всього світового ринку телемедицини (46%). Далі йдуть такі країни як Канада, Китай, країни Скандинавії. Однією з багатьох причин, що є потужним драйвером розвитку телемедицини для країн Скандинавії, є звісно що географічний чинник. Як от Норвегія –що має чималу засніжену складну північну територію. І саме тут телемедицина і прийшла на допомогу. Канада має приблизно такі самі географічні характеристики, дуже сприятливі для розвитку телемедичних технологій. Те саме можна сказати і про Китай, що має величезну територію і дуже високою кількістю та щільністю населення.

У Великобританії тим часом спостерігається найбільший попит на телемедицину серед країн Європи. Тут важливою причиною для такого розвитку як і в Японії є фактор старіння населення, багато довгожителів, і в цьому випадку телемедичні рішення і консультації допомагають лікарям і їх пацієнтам поліпшити систему медичної допомоги.

У США галузь телемедицини практикується з часу відкриття першої спец. клініки в Массачусетсі, ще з кінця 60-х років. В Америці є можливість часткового відшкодування витрат на телемедичні послуги із страховки. Існують суворі норми для захисту особистих лікарських даних пацієнта. Особливою популярністю користується сегмент послуг віддаленої психотерапії, і онлайн моніторинг стану здоров'я людей, проживаючих у сільській місцевості і людей похилого віку. Регулювання телемедичних послуг щороку доповнюють новими поправками й доповненнями законів, при цьому ускладнений цей процес тим що у кожному штаті діють свої особливі правила та закони.

Норвегія, країна, в якій вже досить давно займались питанням, як надати медичну допомогу людям, що живуть у віддаленій та важкодоступній місцевості. У 1993 році рішення було знайдене. У країні почався процес

впровадження телемедицинських технологій . Тут система охорони здоров'я рухалась у двох напрямках — консультації лікар-пацієнт а також конференційний напрямок для покращення освіти лікар-лікар. У першому випадку, лікарі надавали допомогу пацієнтам безпосередньо, в другому — лікарі консультували колег у глибинці. Стартували вони з локалізації на півночі країни і шести осіб персоналу. Зараз це загально державний великий health проєкт. І для громадян вже звичні ситуації — проконсультуватися по відеозв'язку, викликати його додому або прийти на консультацію до лікаря.

У європейських країнах телемедицина — поняття вже добре знайоме і давно звичне. Окрім Норвегії успішні проєкти є в Італії, Естонії, Фінляндії, Франції й Данії. У багатьох країнах телемедицина працює також і в комерційному секторі. Яскравий приклад — наш сусід Польща. Польські медики встановили, що 8 з 10 запитів пацієнтів по допомогу вони можуть задовольнити онлайн. Адже не всім потрібні важкі операції, або оперативне втручання у перебіг хвороби. У більшості пацієнтів діагнози досить просто встановлюються і вилікувати людину можна навіть простими антибіотиками. Якщо ж через декілька сеансів хвороба буде ускладнюватись тоді уже людина буде потребувати госпіталізації, але на щастя більшість проблем вирішуються за кілька дзвінків.

Система охорони здоров'я Австралії – одна з найефективніших у світі згідно з дослідженнями різних видань, зокрема Bloomberg. Телемедицині цієї країни надає активну підтримку держава поряд з іншими нововведеннями. Австралійське агентство цифрової медицини на даний момент активно займається розвитком цієї галузі. Більшість людей живе на узбережжі а більшість території це пустеля, в таких умовах транспортування хворого іноді стає проблемою. Велика кількість часу, погані дороги на більшості територій, спека, усі ці чинники стали потужним рушієм для використання та вдосконалення телемедицинських напрямків на території цієї країни. Компанія CSIRO представила дані згідно яких використання у лікуванні людей телемедицинських систем дозволило системі охорони здоров'я країни заощадити близько 2.1 мільярдів доларів лише за один рік використання. На дистанції економія буде ще більш вражаючою навіть із врахуванням вартості уже

створених мереж, подальшому їх вдосконаленні а також якщо система або деякі складові частини будуть потребувати налагодження та ремонту.

У Японії телемедичні процедури, а зокрема дистанційний моніторинг стану здоров'я а також розвиток та перебіг хронічних хвороб були доступними через високий технологічний розвиток країні, і необхідним тому що кількість людей похилого віку у цій країні невинно зростає через досить хороший загальний рівень здоров'я, а також країна досить часто страждає від важких стихійних лих наприклад землетруси, цунамі, тайфуни та інші. Як для прикладу події 11 березня 2011 року. Японію струсонув один із найбільш руйнівних землетрусів в історії країни та світу магнітудою 9 баллів. І через кілька годин після таких потужних поштовхів на Японію обрушилось цунамі. Коли на країну одне за одним обрушились дві природні катастрофи потрібна вся можлива допомога найбільш постраждалим районам. В такому випадку вціліла телемедична мережа стала дуже потужною підтримкою. При наданні медичної допомоги людям похилого віку вдома все більшого значення набувають можливості телемедицини і дистанційного моніторингу та діагностики.

вислови або слова з переліку тих що запропонує програма в спеціально розроблених шаблонах для оглядів.

Правильність і детальна повнота введення інформації пацієнта контролюється автоматично в час формування медичного документа.

Введення особистої інформації пацієнта в електронну карту відбувається з допомогою шаблонів тому її збереження в сталому та структурованому вигляді забезпечить широкий спектр можливостей для аналізу даних і суттєво полегшить обмін данною інформацією по інтернету з іншими медичними установами за необхідності. Зручна система для пошуку цих даних дозволяє швидко та без особливих зусиль персоналу знайти необхідну інформацію серед великих об'ємів медичної документації.

По перше не буде ризику загубити картку, або наприклад забути її перед прийомом та повертатись додому тому що у такому випадку доведеться чекати наступного часу на прийом. По друге лікар зможе переглянути її в будь який момент для кращого розуміння стану пацієнта.

Іноді пацієнт не має змоги вийти з дому, якщо хвороба підкосила зовсім неочікувано. Тому фельдшер у селі, або бригада швидкої у місті повинні мати змогу швидко зв'язатись з експертом. Наприклад в Одеській області 150 працівників служби екстреної медичної допомоги пройшли спец навчання та відпрацювали практичні навички по передачі ЕКГ за допомогою нового телеметричного пристрою до спеціаліста у консультативний телемедичний центр названий Єдина оперативна диспетчерська служба «103».



Рисунок 2.2 - Обладнання екіпажу швидкої в Одесі

Також фельдшери отримали у своє розпорядження нове обладнання.



Рисунок 2.3 - Кардіограф

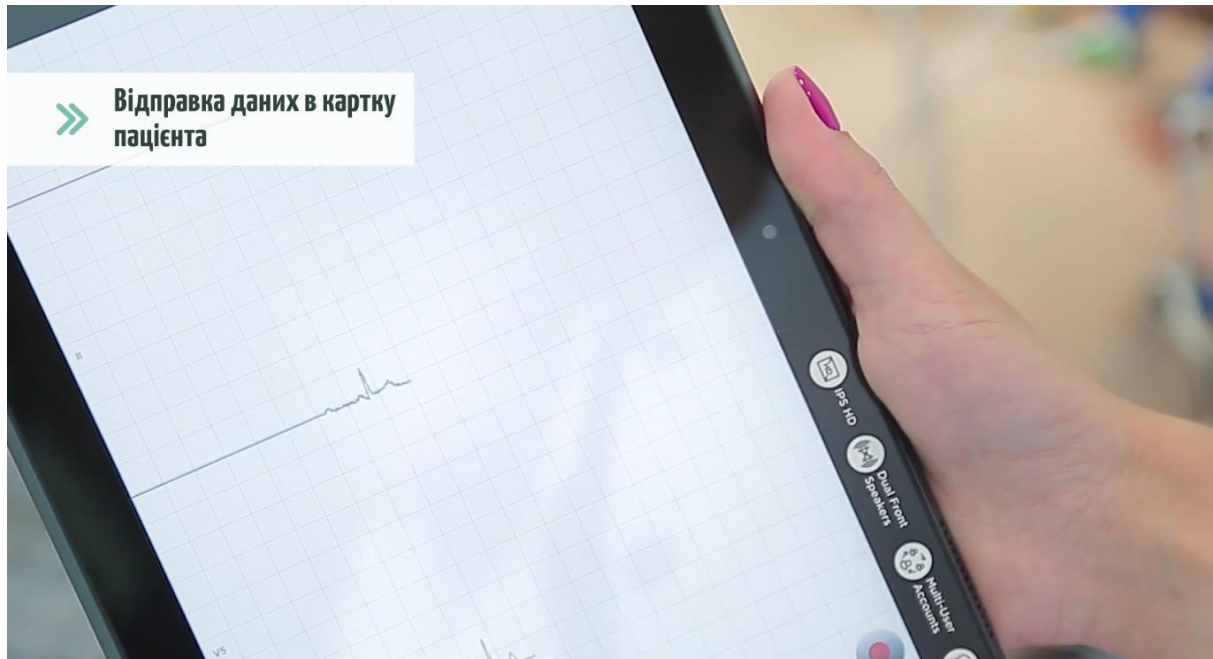


Рисунок 2.4 - Планшет для зв'язку з спеціалістом

Після зняття показів дані та запит на відеоконсультацію відправляються в диспетчерський пункт який є у лікарнях де впроваджено телемедицину, а диспетчер направляє результати обстеження разом із запитом до необхідного лікаря.

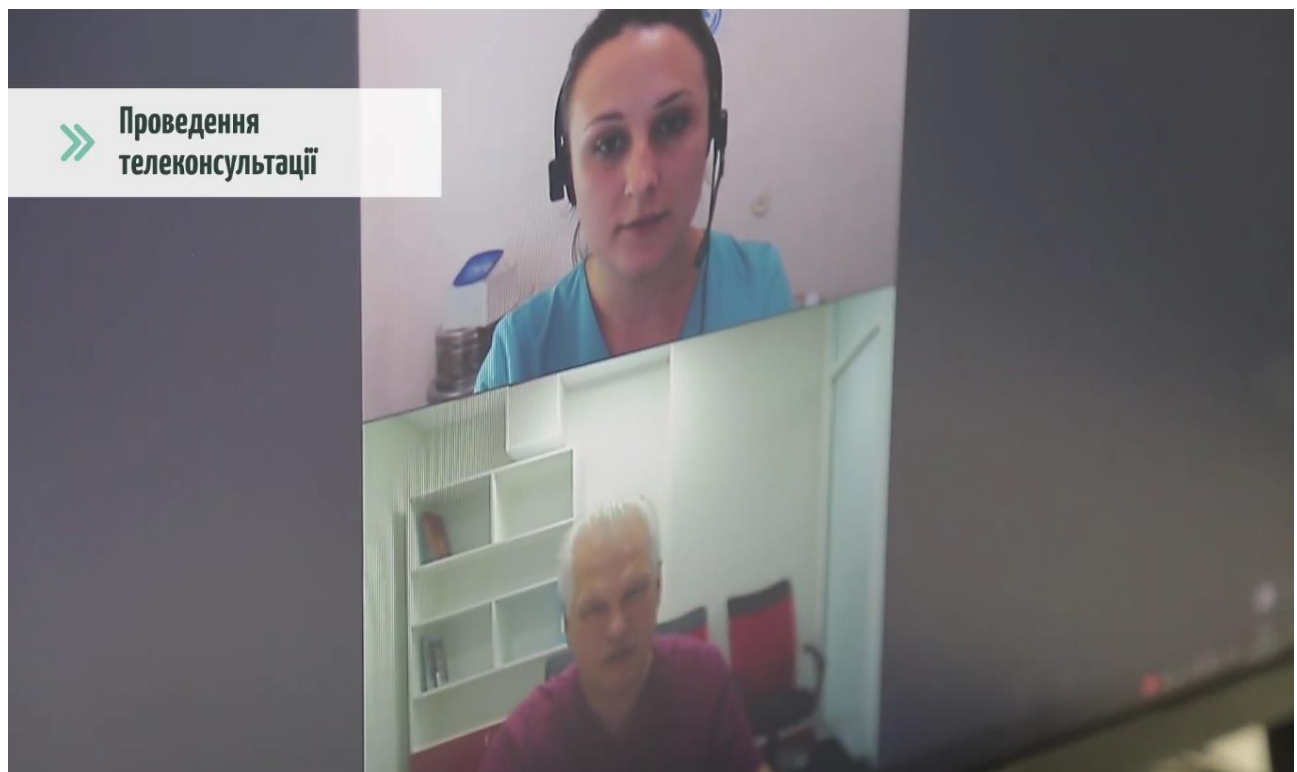


Рисунок 2.5 - Онлайн консультація з лікарем



Рисунок 2.6 - Онлайн консультація зі сторони лікаря

Після консультації лікар по готовим шаблонам складає висновок і встановлює діагноз та схему лікування і відправляє на планшет. Буквально за годину зроблено об'єм роботи який без телемедицини зайняв би кілька днів. Це навіть не враховуючи необхідність їздити до лікаря особисто.

Надсилати можна покази кардіограм, результати аналізів крові, рентгенівські знімки, результати ультразвукової діагностики, МРТ, електронну медичну карту, рецепти на антибіотики, та багато іншого, практично всю інформацію необхідну для розробки схеми лікування.

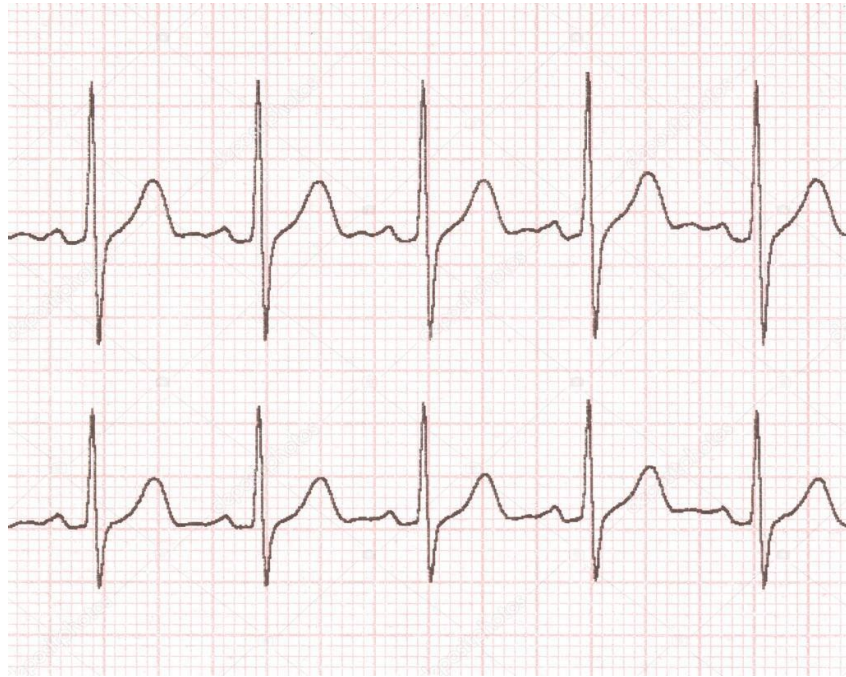


Рисунок 2.7 - Кардіограма



Рисунок 2.8 - Рентгенівський знімок



Рисунок 2.9 - Ультразвукова діагностика

Завдяки телемедичним технологіям лікарі з різних областей можуть контактувати, ділитись досвідом, а також досліджувати нові рідкісні хвороби з якими досі не стикались.

2.2 Способи передачі даних

Телемедична система складається із сукупності базових робочих станцій різної комплектації, з'єднаних каналами передачі даних: стандартними й цифровими телефонними лініями, волоконною оптикою і навіть супутниками зв'язку.

Застосування того або іншого із способів передачі обмежується параметром ширини частотної смуги і тісно пов'язаної з нею характеристикою швидкістю передачі інформації (тобто кількістю інформації яка передається за одиницю часу, позначається біт/сек). Системи з досить вузькою смугою частот, наприклад такі як звичайні телефонні лінії, зовсім недорогі, але при цьому їхня пропускна здатність зовсім недостатня для передачі повноцінних та чітких відео зображень. Однак системи побудовані на телефонних лініях цілком придатні для передачі

таких даних як фото зображення, мови, тексту і інших даних що не потребують великої швидкості передачі. Єдиного методу, як і потрібної ширини для смуги частот, що підійшла б для вирішення всіх завдань що стоять перед телемедициною, не існує. Технічні характеристики для кожної системи визначаються виходячи з потреб конкретних користувачів.

Пропускна здатність доступна для мереж із широкою смугою частот уже є достатня для передачі повноцінного відео. Однак у сільській місцевості або в інших віддалених від індустріальних центрів місцях такі мережі недоступні. При використанні різних процедур стиснення інформації стиснуте інтерактивне відео можна було б передавати й по телефонних системах з більш вузькою смугою частот, але одержувані відео при такому рухаються нерівномірно, що ускладнює спроби розрізнити подробиці а також тонкі рухи. Такі системи коштують досить дорого, витрати на передачу інформації залежать від ширини даної смуги.

Щоб не було плутанини та безлічі збоїв складових системи для даних у загальній мережі повинен існувати єдиний узгоджений стандарт. Стандарт або протокол передачі даних це певні програми створені для взаємодії зовсім різних функціональних елементів комп'ютерних мереж, створюючи правила обміну інформацією між різними комп'ютерами й периферійними пристроями які об'єднані у мережу. Відсутність єдиного затвердженого стандарту медичної інформації тобто для її накопичення, зберігання, а також передача це серйозна перешкода для глобалізації а також подальшого розвитку телемедичних технологій. На даний момент є близько сотні різноманітних стандартів, які потрібні для роботи з різноманітною медичною інформацією. В більшості випадків дані назви стандартів – це аббревіатури різних груп або організацій чи установ, що брали активну участь у їхній розробці.

У світі досить багато всіляких організацій, що займаються різними питаннями телемедицини. Існує величезна кількість стандартів що потрібні для передачі багатьох видів медичної інформації: HL7, ASTM, ASC, X12, DICOM тощо. Так, в даний час в Україні використовують такий стандарт як DICOM, що складається із 13 розділів, кожен з яких представлений приблизно яу окремий стандарт ISO. В роботі зараз стандарт ДСТУ ISO 17432:2009. Структура

повідомлень для стандарту DICOM дозволяє виокремлювати з отриманих повідомлень самі зображення, а це надає важливу можливість розпочинати роботу з приймання а також візуалізації зображень, які передали з діагностичних пристроїв а також пристроїв збору інформації, таких як сканери рентгенівських знімків. При цьому використовується програма для швидкоїобробки отриманих повідомлень сторонніми методами візуалізації, пропускаючи іншу інформацію.

Таким чином, стандарт DICOM дозволяє правильно організувати не тільки пересилання даних всередині мережі, але також автоматичну обробку. Він значно зменшує час потрібний на підготовку і проведення досліджень, а також управління зображеннями та інформацією. Для досягнення такої високої ефективності він підтримує абсолютно всі стадії діагностики і так знижуючи собівартість через скорочення часу обслуговування і внаслідок цього відмови від плівок та супутніх витрат на зберігання, а ще зменшення втрат зображень і результатів досліджень. У цілому використання такого стандарту як DICOM видається досить важливою задачею, вирішення якої зможе принести велику користь для медпрацівників яким необхідні частіше медичні зображення.

Спочатку цей стандарт розробляли в 1983 році два закладами –це Національною асоціацією виробників електронного устаткування а також Американським радіологічним коледжем, заради забезпечення сумісності між різним встановленим устаткуванням, усі виробники якого мали входити в асоціацію. В 1985 році вирішили створити об'єднаний комітет, в задачі якого мала входити розробка єдиного стандарту, що забезпечить стабільну передачу цифрових медичинських зображень. Це сприяло стрімкому розвитку систем архівації а також передачі зображень, забезпеченню їх нормальної взаємодії з автоматизованими лікарняними архівними системами та створенню спільних баз даних, які містили б усю діагностичну інформацію, яку отримували з допомогою багатьох географічно віддалених пристроїв. Це був найперший стандарт який передавав спершу зображення.

Стандарт HealthLevel 7 був створений для електронного обміну різними документами в віддалених закладах охорони здоров'я, особливо якщо в них

пацієнту надавалась інтенсивна медична допомога. Цей стандарт використовується в таких країнах як США, Канаді, Європі, Австралії, Ізраїлі, Японії та інших. Учасників, тобто користувачів та виробників, об'єднала одна спільна ціль така як спростити взаємодію всіх комп'ютерних додатків розроблених різними, часто конкуруючими виробниками. Згодом цьому комітету дали назву HL7 «Health Level 7» і поставили перед ним завдання стандартизувати різні формати а також протоколи обміну наборами даних.

Стандарт обміну різними медичними даними в електронному вигляді стандарту HL7 «Health Level 7» (Сьомий рівень) який легко сполучається з усіма іншими протоколами і стандартами, а це уже дозволяє використовувати його як я стандарт в приладах від багатьох виробників різноманітної медичної техніки. Організація HL7 засновали в 1987 році для того щоб розвивати стандарти покращення електронного обміну всією клінічною, фінансовою а також адміністративною інформацією між різноманітними комп'ютерними медичними системами, до яких часто відносять: госпітальні інформаційні медичні системи, клінічні та лабораторні системи та багато інших. У такий спосіб, цей стандарт використовують в більшості для обміну різними медичними записами.

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМИ НА МОВІ ПРОГРАМУВАННЯ C++

3.1 Вибір мови програмування

C++ (Сі-плюс-плюс) — мова програмування з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом в AT&T Bell Laboratories (Мюррей-Хілл у штаті Нью-Джерсі) 1979 року та початково мала таку назву як «Сі з класами». Згодом дана мова була перейменована розробником на C++ у 1983 р. Базується на більшості правил та можливостей мові С. Вперше описана стандартом ISO/IEC 14882:1998, найбільш актуальним же є стандарт ISO/IEC 14882:2014.

Ціль створення C++ було доповнення С можливостями зручними для масштабною розробки ПО, з збереження гнучкості, швидкості і портабельності С. Разом з тим розробники хотіли зберегти сумісність з С. Перший час мова розвивалась без формальних обмежень, а лише по мірі встановлення перед мовою задач.

В 1990 ця мова стала однією з найпопулярніших. Нововведення C++ порівняно з С:

- Підтримка програмування через шаблони функцій і класів.
- Додаткові типи даних.
- Простори імен.
- Перезавантаження імен функцій.
- Перезавантаження операторів

Типи даних.

В C++ типи практично повністю повторюють типи даних в С.

- Символьні, такі як `char`, `wchar_t`.
- Цілочисельнізнакові, як `signed char`, `short int`, `int`, `long int`.
- Цілочисельні без знакові, як наприклад `unsigned char`, `unsigned short int`, `unsigned int`, `unsigned long int`.
- З плаваючою точкою, `float`, `double`, `long double`.

- Логічний, `bool`, який має значення `true` або `false`.

Також мова програмування C++ ввела таке поняття як посилання.

Класи.

Основною відмінністю в C++ стали класи. Це основна концепція об'єктно орієнтованого програмування. Розширення структурного програмування, в якому основними є поняття класів та об'єктів.

Класи в C++ це абстракція що описує властивості ще не існуючих об'єктів. Клас описує дані та методи, які об'єкти цього класу будуть використовувати. Кожен клас має опис логічно завершеної одиниці програми. Об'єкти створені на основі одного класу називаються екземплярами класу. Таким чином класи це деякі описи, схеми по яким створюються об'єкти.

У даній мові програмування клас оголошується командою `class`, специфікатор доступу за умовчанням `private`. Класи мають свої елементи, такі як змінні (навіть інші класи), функції, конструктори та деструктори. Члени оголошуються за допомогою специфікаторів доступу публічно – `public` і приватно – `private`. Також між класами можливе наслідування за допомогою специфікатора `protected`. Наслідування – це процес коли визначення нового класу базується на визначенні вже існуючого, такий клас отримає властивості та поведінку базового класу, і може доповнити власними. Застосування механізму успадкування дозволяє покращити повторне використання коду через використання вже визначених властивостей та методів базових класів.

Для спрощення процесу ініціалізації об'єкту є спеціальна функція названа конструктором. Ці функції не повертають значення. Запускаються на виконання конструктори автоматично для кожного екземпляру класу під час його опису.

Також у C++ можна побудувати таку функцію як деструктор. Вона виконується під час знищення екземпляру класу, коли програма завершена, при виході з функції, а ще коли звільняється пам'ять, виділена для екземпляра класу.

Інтерфейс класу можна відділити від реалізації. Інтерфейс визначає методи та властивості класу, а реалізація це спосіб забезпечення його роботи. Процес відділення виконують для того щоб приховати спосіб забезпечення

працездатності класу.

Інкапсуляція даних означає що для членів класу можна встановлювати рівень доступності з других частин програмного коду. Інкапсуляція покращує збереження даних в класі, з допомогою інших методів перевірки даних на допустимі значення. Виникає поняття в класі, таке як приховання даних, доступ до таких даних не прямий, а через спеціальні методи доступу, або властивості класу. При цьому дані знаходяться в прихованій секції класу, а методи в загальнодоступному розділі.

Оголошений поза функціями клас називається глобальним і його об'єкти можуть бути використані у будь якому місці в програми. Якщо клас оголошений в тілі функцій то він є локальним класом, і використовується лише у межах даної функції. Клас може бути пустим. Створюють такі класи для того щоб можна було протестувати готову частину програми, в який ще не розроблені всі класи. Замість цих майбутніх класів створюють пустий, і проект можна компілювати.

Також у програмах використовують фреймворки. Фреймворки відрізняються від бібліотек тим, що бібліотека в програмному продукті може використовуватись як підпрограмний набір близької функціональності, але не впливати на архітектуру програми і не накладає обмежень. В той час як фреймворк має певні правила побудови архітектури, задаючи певний каркас – правила поведінки за замовчуванням, які потрібно буде розширювати та змінювати, згідно вказаним вимогам. У відмінності від бібліотеки, яка поєднує в собі набір близької функціональності, фреймворк може мати велику кількість різних за тематикою бібліотек.

Іншою фундаментальною відмінністю може бути інверсія управління, коли код користувача викликає функції бібліотеки і після виклику отримує управління.

Одна з головних переваг при використанні фреймворків це стандартна структура програми. Набувати популярності фреймворки почали з появою графічних інтерфейсів користувача, які мають тенденцію до реалізації стандартної структури програм. Набагато простішим став процес автоматичного створення графічних інтерфейсів, так як структура внутрішньої реалізації коду визначалась наперед.

Фреймворк визначається як множина класів, а також визначень способів їх взаємодії. Процес створення фреймворка полягає у виборі множини завдань. Перший комерційний фреймворк був MacApp. Написаний Apple для Macintosh з допомогою розширеної версії мови ObjectPascal, а з часом переписаний на C++.

3.2 Розробка програми

Метою аналізу технічного завдання є максимально повний опис поставленої задачі, відповідно до технічного завдання.

Під час проходження етапу аналізу спочатку було визначено такі питання: – Хто буде діючими суб'єктами?

- Хто вводитиме інформацію?
- Хто запрошуватиме?
- Чи буде хто-небудь ще взаємодіяти з програмою?
- Чи буде сама програма взаємодіяти з іншими програмами?

В даному випадку з програмою Telehealth має працювати декілька ролей користувачів – Адміністратор, медпрацівник.

Наступне, що потрібно було з'ясувати, це групу дій, які ініціюватимуть діючі суб'єкти. Виходячи із технічного завдання, діючі суб'єкти будуть виконувати такі дії:

Медпрацівник:

- почати роботу з програмою;
- зареєструвати нового користувача з правами «медпрацівник»;
- переглянути список медпрацівників у системі;
- надіслати повідомлення потрібному користувачу месенджера;

Адміністратор:

- почати роботу з програмою;
- переглянути список медпрацівників у системі;
- переглянути список студентів вибраної групи;
- керувати медпрацівниками;

Розробка методів

В даному підрозділі описується наступний етап розробки програмного забезпечення – еволюцію, що представляє собою етап послідовної реалізації. Етап починається із створення основного інтерфейсу проекту та бібліотеки класів. Після цього створений прототип необхідно буде протестувати та налагодити

. В програмі використовуються контейнери Qt, що допомагають організувати роботу із записами таблиць програми.

У всіх агрегатах було прийнято рішення зберігати покажчики замість самих об'єктів, щоб уникнути зайвого копіювання даних у пам'яті. Зберігання самих об'єктів необхідно застосовувати в тих випадках, коли об'єктів мало і вони невеликі.

Як уже було зазначено вище, багато атрибутів (методи) для кожного з класів є похідними з тих іменників, які самі не стали класами. Інші атрибути можуть бути виведені з асоціацій в діаграмі класів. Асоціації можуть визначати ті атрибути, які є покажчиками або посиланнями на інші класи. Це пояснюється неможливістю асоціювати щось з чимось, що знаходиться невідомо де.

Проектування і опис інтерфейсу користувача

Для програмного клієнта-месенджера розроблено графічні вікна за допомогою QtDesigner.

QtDesigner - кросплатформенне вільне середовище для розробки графічних інтерфейсів (GUI) програм, які використовують бібліотеку Qt . Входить до складу Qtframework.

QtDesigner дозволяє створювати графічні інтерфейси користувача за допомогою ряду інструментів.

QtDesigner може бути запущений як окремий додаток , так і у вбудованому в IDE QtCreator вигляді.

Опис файлової структури програми

Оскільки програмний месенджер складається із серверної та клієнтської частини, тому вихідні файли поділені на файли сервера та клієнта, проте є програмні файли класів, які є спільними. Розглянемо перелік вихідних файлів.

Вихідні файли сервера:

- «main.cpp» -лістинг даного файлу знаходиться в додатках. В цьому файлі міститься головна функція сервера, за допомогою якої запускається ;

- «connectionhandler.h» - лістинг даного файлу знаходиться в додатках.

Даний файл містить оголошення класу ConnectionHandler.

- «connectionhandler.cpp» - лістинг даного файлу знаходиться в додатках.

Даний файл містить реалізацію методів класу ConnectionHandler.

- «serverdatabasehelper.h» - лістинг даного файлу знаходиться в додатках. Даний файл містить оголошення методів класу ServerDatabaseHelper;i

- «serverdatabasehelper.cpp» - лістинг даного файлу знаходиться в додатках. Даний файл містить реалізацію методів класу ServerDatabaseHelper.

Далі розглянемо вихідні програмні файли програмного клієнта:

- «authmenu.h» – даний файл містить оголошення класу AuthMenu;

- «authmenu.cpp» – даний файл містить реалізацію методів класу AuthMenu;

- «chatwidget.h» – даний файл містить оголошення класу ChatWidget ;

- «chatwidget.cpp» – даний файл містить реалізацію методів класу ChatWidget;

- «database.h» – даний файл містить оголошення класу DataBase;

- «database.cpp» – даний файл містить реалізацію методів класу DataBase;

- «mytextedit.h» – даний файл містить оголошення класу MyTextEdit;

- «mytextedit.cpp» – даний файл містить реалізацію методів класу MyTextEdit;

- «netservice.h» – даний файл містить оголошення класу NetService;

- «netservice.cpp» – даний файл містить реалізацію методів класу NetService;

- «popup.h» – даний файл містить оголошення класу PopUp;

- «popup.cpp» – даний файл містить реалізацію методів класу PopUp;

- «popupmanager.h» – даний файл містить оголошення класу PopUpManager;

- «settings.h» – даний файл містить оголошення класу Settings;

– «settings.cpp» – даний файл містить реалізацію методів класу Settings.

Спільні програмні файли знаходяться у каталозі «Data», це є такі файли: authanswer.cpp, authanswer.h, basedata.cpp, basedata.h, databasehelper.cpp, databasehelper.h, databaseresponse.cpp, databaseresponse.h, filedata.cpp, filedata.h, message.cpp, message.h, user.cpp, user.h, userdata.cpp, userdata.h, userlist.cpp, userlist.h.

3.3 Експлуатація програми

Так як телемедицина призначена для спрощення комунікації між пацієнтом та лікарем, а в Україні рівень комп'ютерної грамотності досить низький, то програма має бути проста та зрозуміла навіть для людей які рідко користуються ПК. Екран входу та роботи мають мати максимально простий та функціональний інтерфейс. Також програмою будуть користуватись люди похилого віку яким досить складно розбиратись у сучасних технологіях.

При запуску програми в першу чергу з'являється екран входу в систему де потрібно ввести логін та пароль, стандартна процедура аутентифікації.

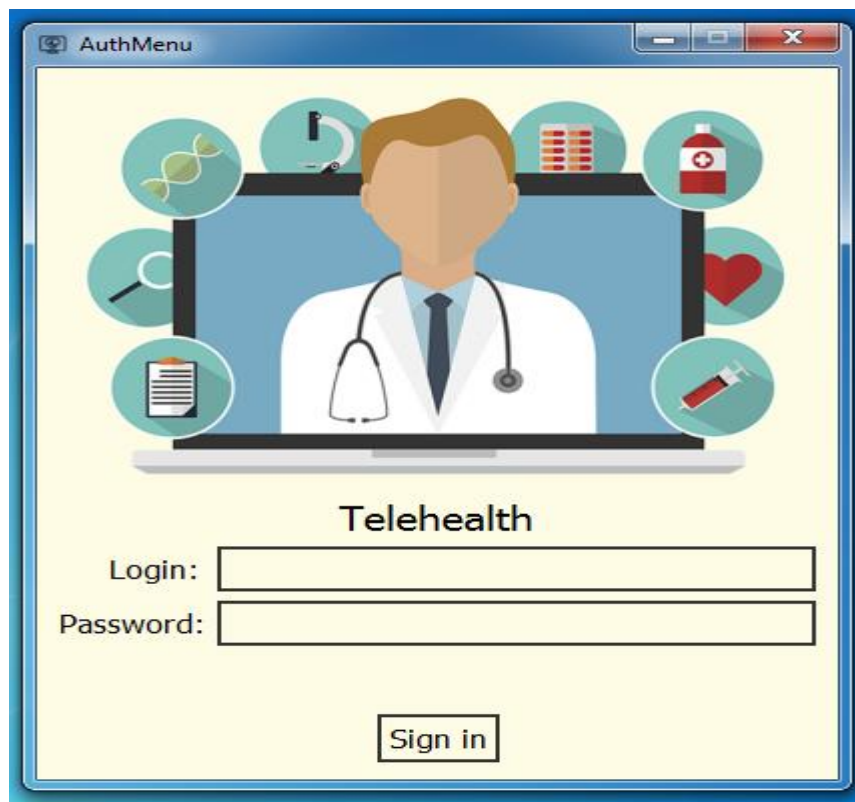


Рисунок 3.2 - Вікно входу

Вікно серверу під час входу. Сервер є основою всіх програм де люди обмінюються інформацією між собою. Він забезпечує стабільний обмін повідомленнями та пересилання файлів.

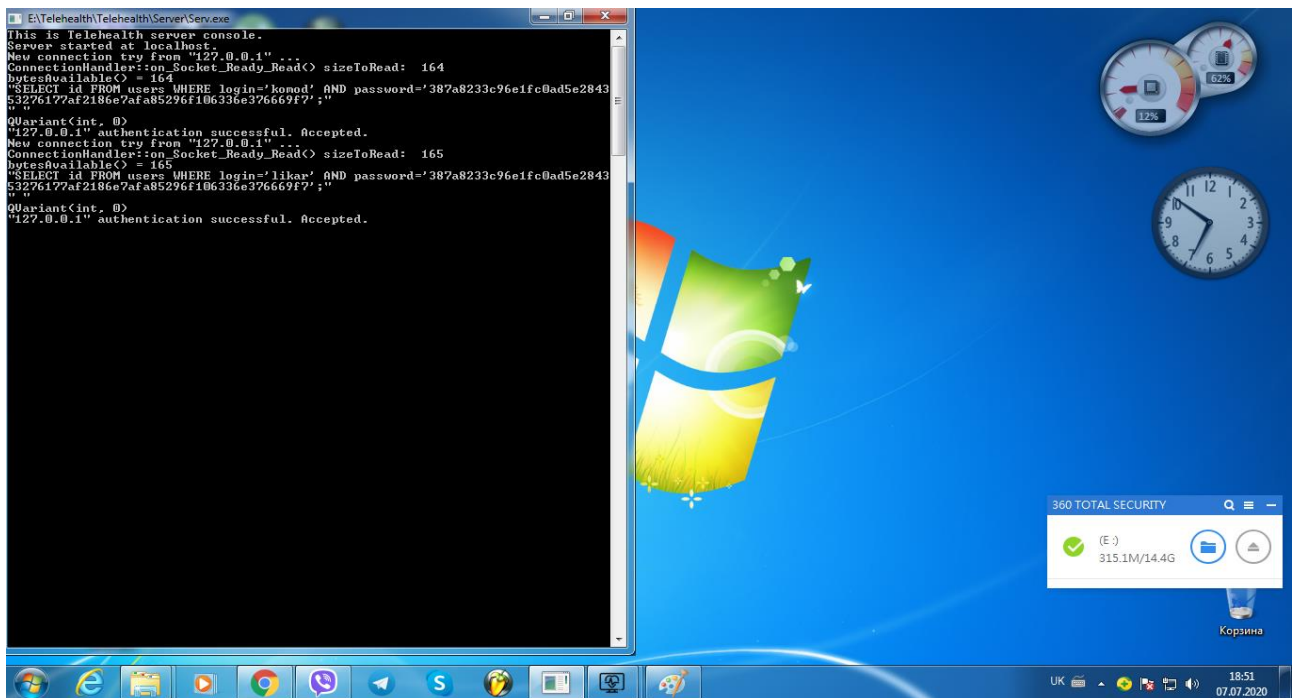


Рисунок 3.3 - Вікно сервера

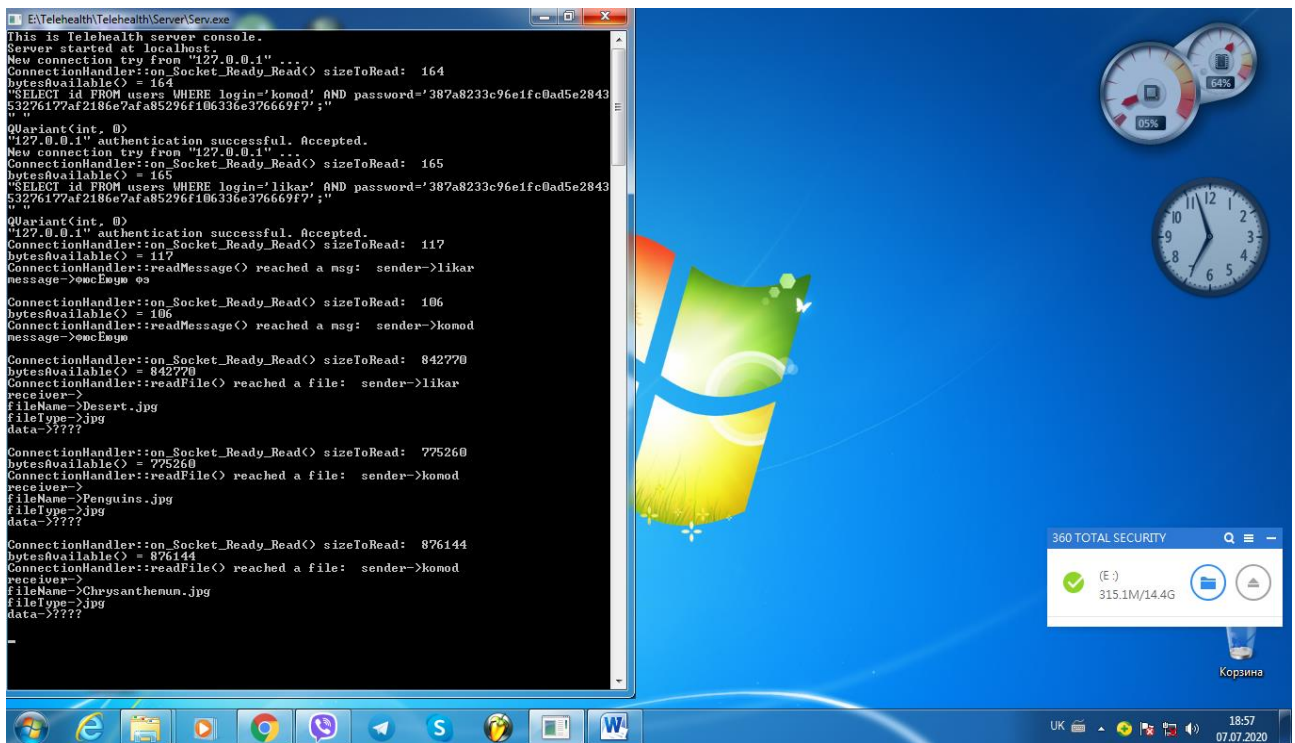


Рисунок 3.4 - Вікно сервера

Після під'єднання до сервера відкривається спрощений чат клієнт. З лівого боку список під'єднаних користувачів. Справа ж вікно для перегляду та відправки повідомлень, обговорення проблем та рекомендацій. Основа телемедицини це діагностування та консультування. Для цього добре підходить текстовий формат спілкування з можливістю пересилання зображень.

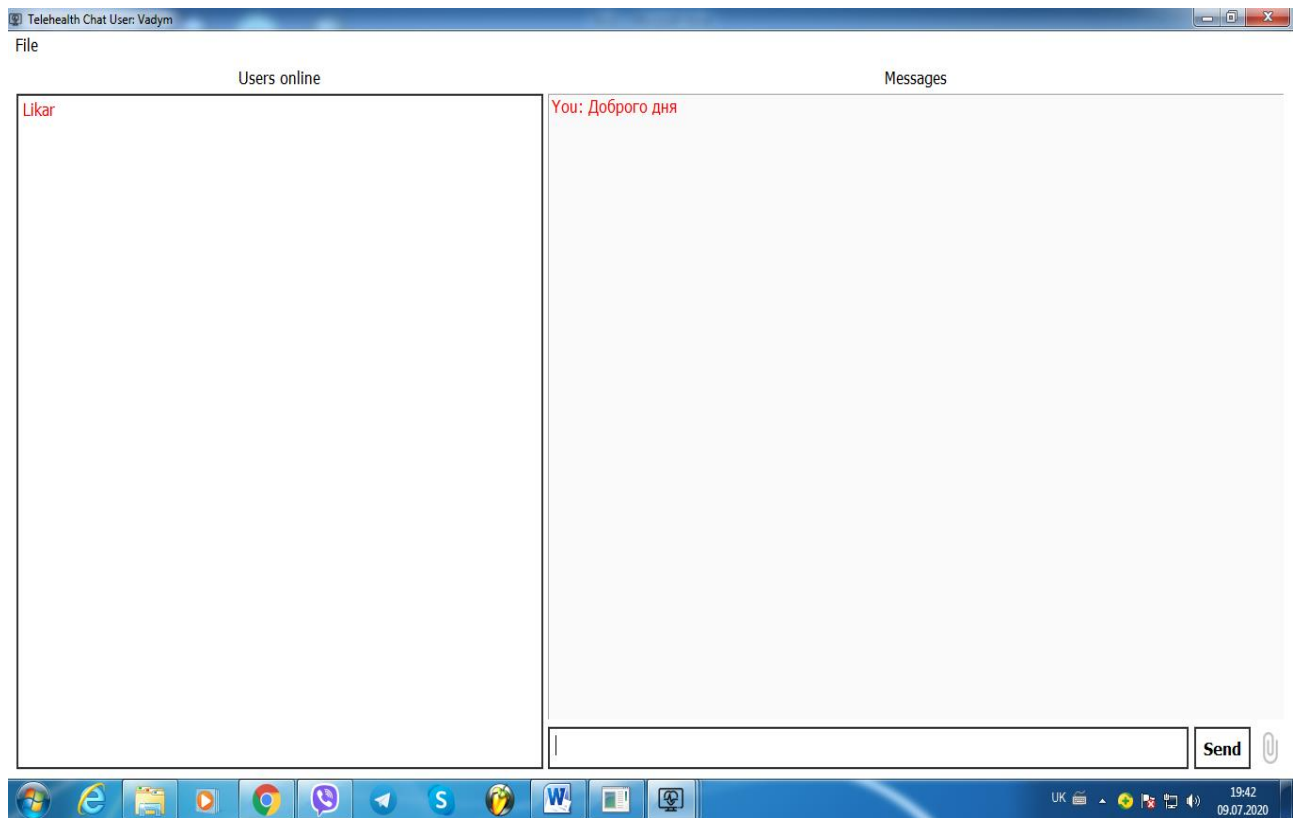


Рисунок 3.5 - Вікно чату

При отриманні повідомлення з'являється спливаюче сповіщення. В цьому є необхідність якщо під час спілкування лікарю або пацієнту потрібно відволіктись на зображення, скажемо ту ж електронну медичну карту, або результати аналізів пацієнта надіслані для вирішення проблеми, або відкрити інше вікно на комп'ютері. Тоді сповіщення вчасно нагадає про можливо важливе повідомлення.

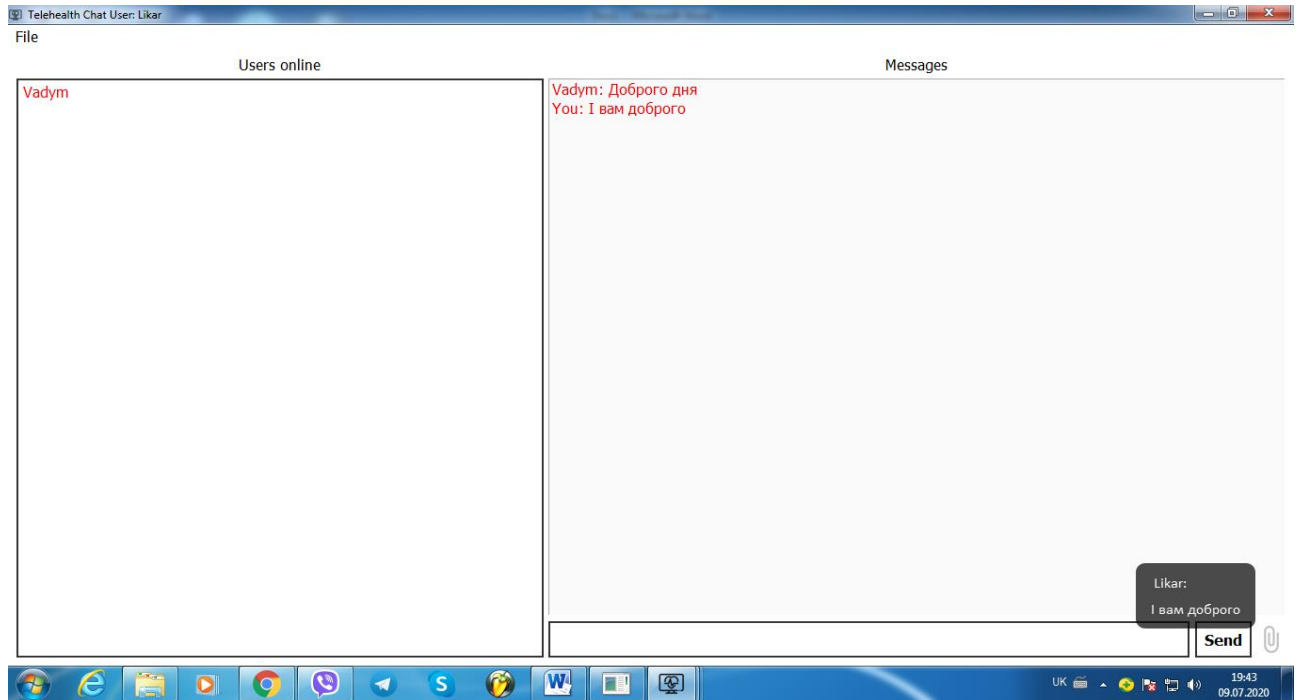


Рисунок 3.6 - Вікно чату

Надсилати можна не лише текстові повідомлення а й зображення. Можна надіслати для прикладу рентген або результати аналізів. Це одна з головних складових телемедицини. Без достатньої кількості діагностичних даних встановити правильний діагноз стає досить складно, а іноді не просто складно а практично неможливо. В таких умовах визначати діагноз стає небезпечно. Швидкий, але не правильний діагноз призводить до неправильної стратегії лікування, а це може погіршити ситуацію. Або при тривалому не визначенні діагнозу пацієнту може погіршати.

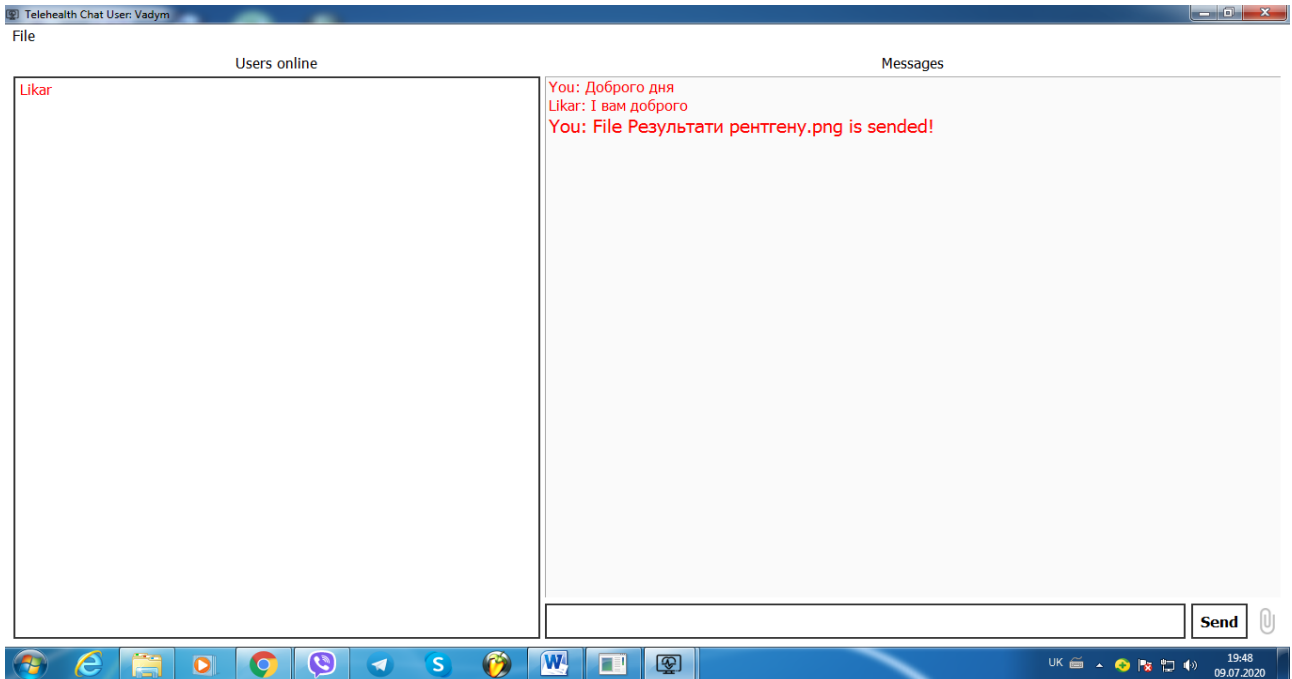


Рисунок 3.7 - Надсилання зображення

При отриманні зображення програма пропонує вказати до якої папки зберігати файл. Тому вибрати сховище краще заздалегідь щоб результати аналізів пацієнта були систематизовані та при необхідності їх знайти не виникало ніяких проблем.

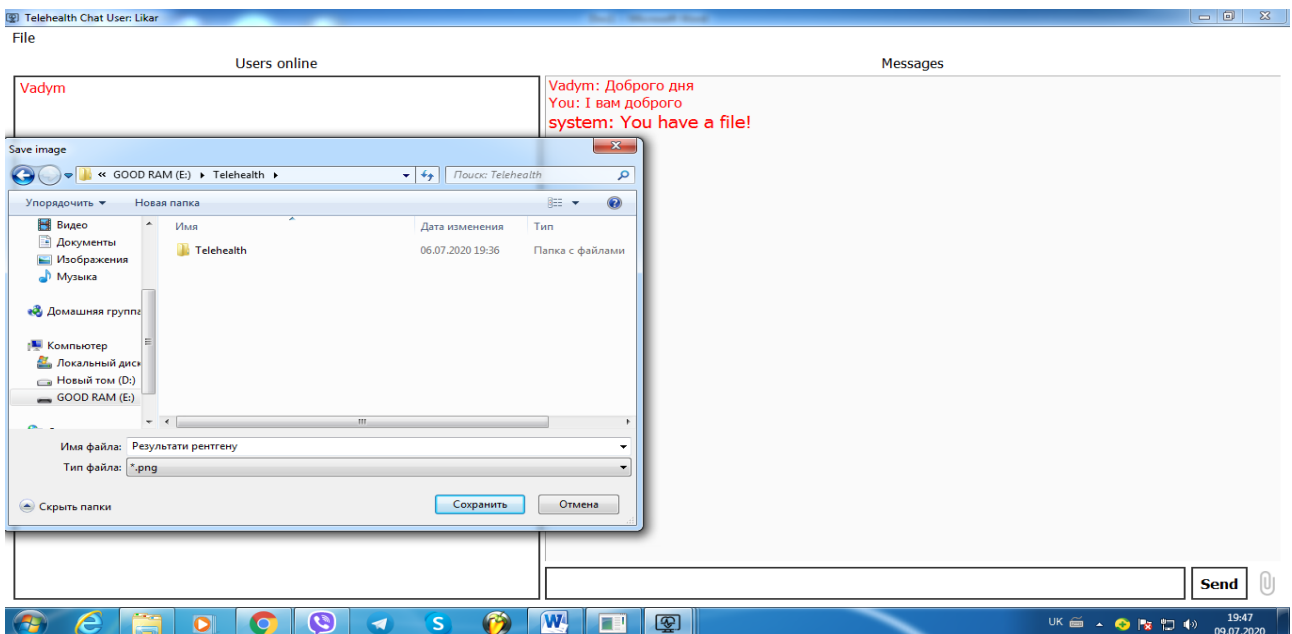


Рисунок 3.8 - Збереження зображення

Також завдяки налаштуванням навіть при згортанні програми вона залишається у пам'яті, а також може надсилати спливаючі повідомлення.

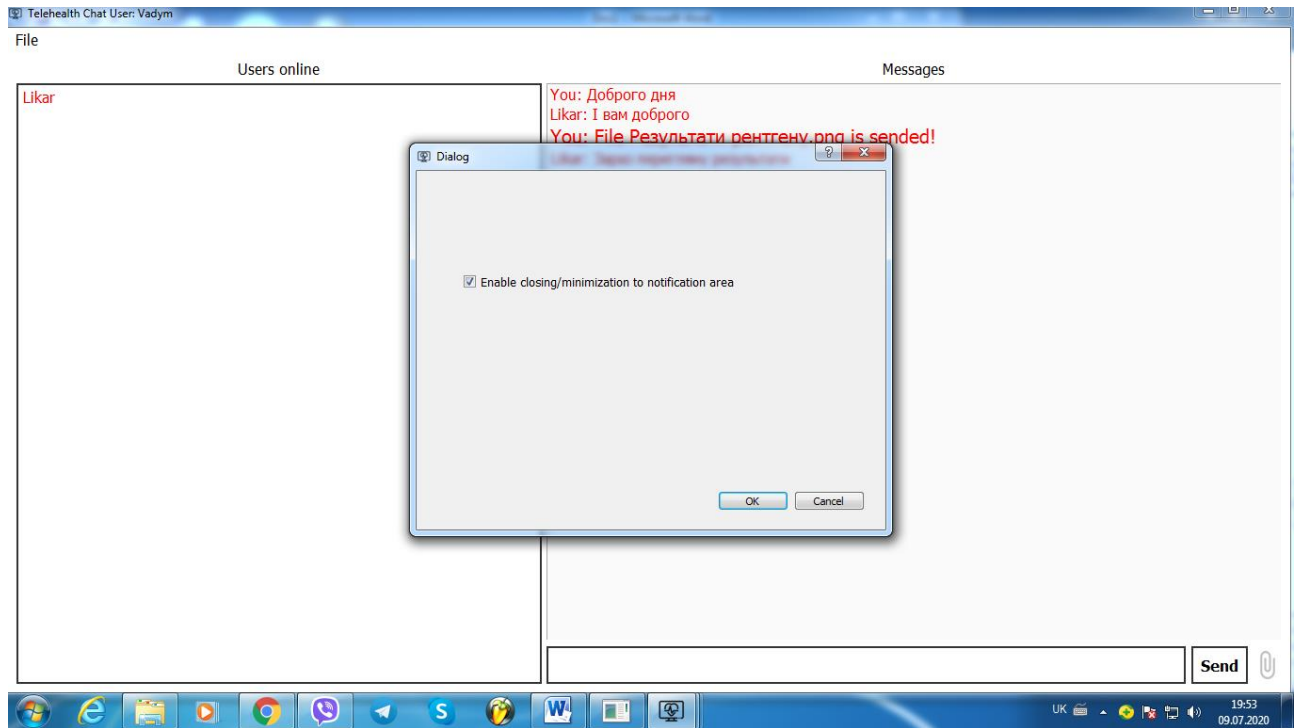


Рисунок 3.9 – Налаштування

Даного функціоналу достатньо для вирішення елементарних проблем. Для більш детальних обстежень, постановлень складніших діагнозів, або дослідження рідкісних клінічних випадків та не типового розвитку хвороб потрібно вдосконалити програму з додаванням відеозв'язку.

РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Ергономіка робочого місця розробника програм

Ергономіка (від грецьк. ἔργον –«робота» і νόμος –«закон» –«закон роботи») –у традиційному розумінні –наука про пристосування посадових обов'язків, робочих місць, обладнання та комп'ютерних програм задля створення найбільш безпечних та ефективних умов праці для людини, виходячи з фізичних і психічних особливостей людського організму.

Більш широке визначення ергономіки, яке було прийняте в 2010 році Міжнародною асоціацією ергономіки (IEA) (Міжнародною ергономічною асоціацією), звучить так: «Ергономіка–це наукова дисципліна, що вивчає взаємодію людини та інших елементів системи, а також сфера діяльності щодо застосування теорії, принципів, даних і методів цієї науки для забезпечення благополуччя людини та оптимізації загальної продуктивності системи».

З цього визначення випливають такі головні завдання ергономіки:

1. Проведення досліджень, спрямованих на пристосування елементів системи "людина –трудовий процес" до природних фізичних і психічних можливостей працівника.
2. Прагнення до забезпечення таким шляхом умов для максимальної ефективності праці.
3. Прагнення запобігти всім можливим загрозам для здоров'я працівника.
4. Прагнення до оптимальної витрати біологічних ресурсів у процесі праці.

Будь-який працівник для успішного виконання покладених на нього функцій потрібні необхідні умови праці. Вони повинні забезпечувати максимальні зручності робітника під час праці при мінімальних затратах сил і часу. Ці умови праці важливі для будь-якої професії.

Умови праці – це сукупність факторів виробничої середовища, які впливають на здоров'я і працездатність людини в процесі її діяльності.

При роботі з комп'ютерами дуже важливо мати правильно обладнане місце, тому як тривала робота може шкодити як зору так і поставі, які можуть перейти у серйозні проблеми зі спиною. Також погіршується працездатність, а відповідно кількість виконаної роботи. Сучасні ІТ компанії облаштовують свої офіси для максимально комфортної роботи, а також спеціально створюють зони для відпочинку. Таким чином для працівників є всіумови для комфортноїпраці. Перед початком роботи необхідно перевірити усе устаткування. При відсутності, або не справному стані роботу розпочинати не можна. Правильне і не правильне положення при роботі з ПК зображено на рисунку 4.1.



Рисунок 4.1 - Неправильне та правильне положення про роботі з ПК

Окрім робочого місця великий вплив має освітлення та рівень шуму в приміщенні. Нормування значень освітленості є у ДБНВ.2.5-28-2006, а нормування шумів здійснюється згідно з ДСН 3.3.6.037.

Довго сидіти це шкідливо для здоров'я тому сучасні ІТспеціалісти практикують роботу стоячи (рис.4.2).



Рисунок 4.2 - Стояче робоче місце працівника ІТ

Основа робочого місця — стіл із регулюванням висоти. Жодних дротів зі столу до підлоги, крім дроту живлення. Це дає можливість піднімати стіл не боячись, що щось буде стягнуто за дріт на підлогу, наприклад, смартфон, підключений в розетку. Для цього під стільницею розташовано органайзер дротів із розетками (спеціальний подовжувач).

Стіл має дуже широкий діапазон висот, нагадує про необхідність встати чи сісти через певні проміжки, бо підключається до комп'ютера чи смартфону. Також має три комірки пам'яті для стандартних висот. Використовую звичайне крісло без регулювання, яке підходить під мій зріст, адже висоту столу можна відрегулювати під посадку у кріслі (рис.4.3).

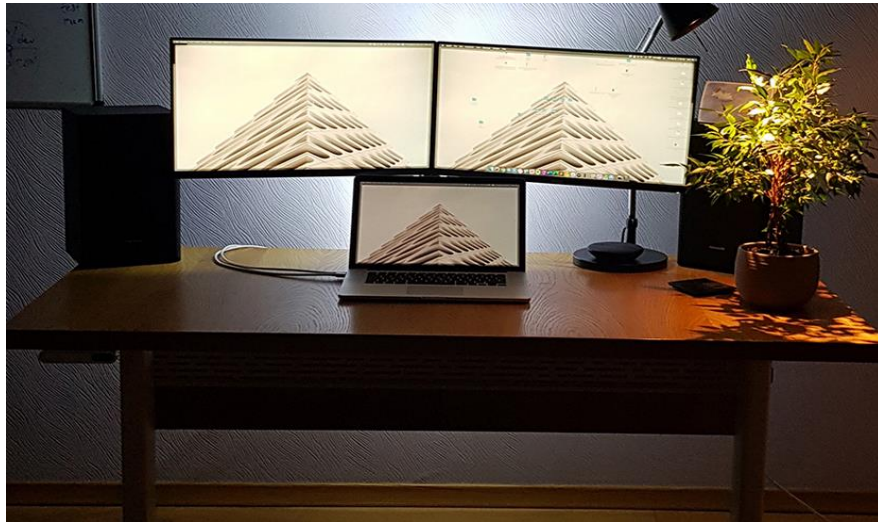


Рисунок 4.3 - Стіл із регулюванням висоти

На столі — два монітори та колонки, що підключаються через Bluetooth, настільна лампа та м'яка підсвітка для роботи в темну пору доби (також вмикаються з комп'ютера чи смартфона). Із аксесуарів — бездротова зарядка, дискети-підставки для чашки та фікус (кактуси більше не в моді, бо немає шкідливих випромінювань від монітору). Ну і ноут.

При роботі стоячи збільшується концентрація, а відтак — і продуктивність, яку всі прагнуть. Також це корисно для всього тіла: фізіологічно та емоційно,

Нормативні документи, які стосуються ергономіки робочого місця:

- ДСТУ 8604:2015 Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. Наказ від 21.12.2015 № 204 Про прийняття нормативних документів України, гармонізованих з міжнародними та європейськими нормативними документами, національних стандартів України та змін до національних стандартів України.

- ДСТУ 7950:2015 Дизайн і ергономіка. Робоче місце під час виконання робіт стоячи. Загальні ергономічні вимоги. Наказ від 22.06.2015 №61 Про прийняття нормативних документів України, гармонізованих з міжнародними та європейськими нормативними документами, національних стандартів України, скасування нормативних документів України та міждержавних стандартів України.

– ДСанПіН 3.3.6.096-2002 Державні санітарні норми і правила при роботі з джерелами електромагнітних полів. Наказ від 18.12.2002 року № 476 Про затвердження Державних санітарних норм та правил при роботі з джерелами електромагнітних полів.

– НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин. Затверджені наказом Комітету по нагляду за охороною праці України Міністерства праці та соціальної політики від 26.03.2010 №65.

4.2 Основні вимоги щодо організації евакуації людей при пожежі на виробництві

Пожежа — це неконтрольоване горіння поза спеціальним середовищем осередком, що розповсюджується в часі і просторі та створює загрозу життю і здоров'ю людей, навколишньому середовищу, призводить до матеріальних збитків.

Пожежна безпека — це комплекс організаційних заходів та технічних засобів, спрямованих на попередження та гасіння пожежі.

Система попередження пожежі — це комплекс організаційних і технічних заходів, спрямованих на не уможливлення умов, необхідних для виникнення пожежі.

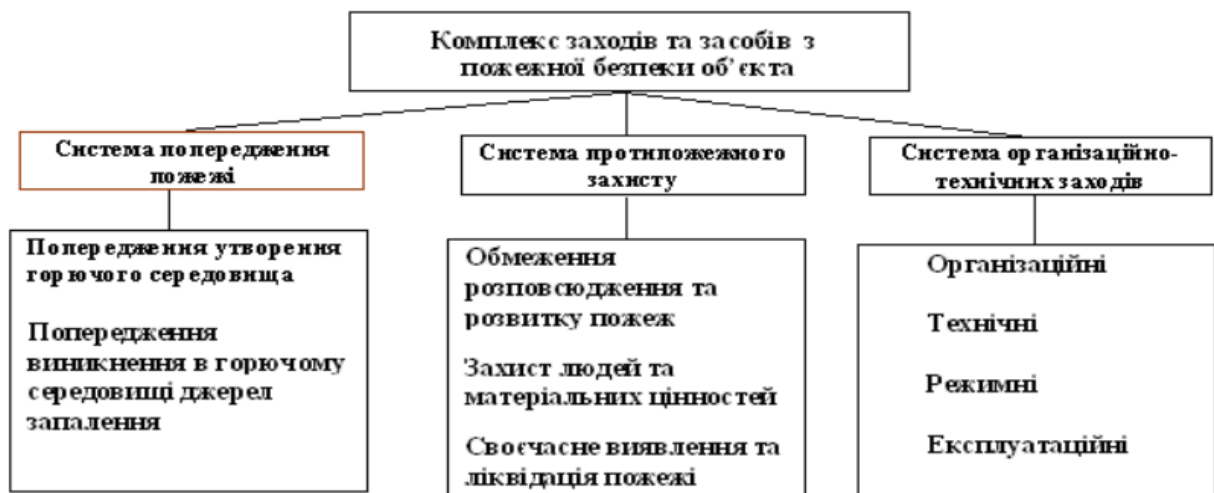


Рисунок 4.4 - Заходи пожежної безпеки

Не дивлячись на всі сучасні засоби протипожежної безпеки, на високотехнологічних підприємствах досить часто все ж трапляються ситуації, що призводять до пожежі. Якщо уникнути небезпечної ситуації не вдалось необхідно чітко розуміти що потрібно робити щоб запобігти шкоді здоров'ю та життю людей.

“Керівники підприємств, а також уповноважені ними особи повинні суворо слідкувати за дотриманням протипожежного режиму, підготовкою планів евакуації та їх практичним відпрацюванням, утриманням евакуаційних шляхів і виходів у належному стані, щоб унеможливити нещасні випадки та затримку під час евакуації людей у разі виникнення пожеж та інших надзвичайних ситуацій.”Світлана Трошина,провідний фахівець з протипожежної безпеки ДПТ України.

Основною метою управляючих евакуації є безпечне виведення людей з будівлі де розгорілась пожежа і запобігання паніці. Допомагають направлять людей світлові покажчики виходу. На підприємствах керівники самі призначають відповідального за пожежну безпеку, а у навчальних закладах евакуацією керує навчальний склад установи.Така людина має мати вищу або середню технічну освіту і працює на підприємстві не менше трьох років. Також відповідальний повинен знати існуючу нормативну базу, технічні характеристики і область застосування первинних засобів пожежогасіння, основні причини, які можуть привести до виникнення вогнищ загоряння або задимлення на об'єкті та іншу інформацію і матеріали, необхідні для виконання своїх функціональних обов'язків.

Кожна споруда підприємства повинна чітко відповідати вимогам протипожежної безпеки. Мають бути розроблені та донесені до працівників плани евакуації з будівлі (рис.4.5), встановлена система оповіщення про небезпеку. Плани евакуації та первинні засоби пожежогасіння (рис. 4.6) мають розташовуватись на видимих місцях.

ПЛАН ЕВАКУАЦІЇ арт.124

розмір 30x40 см

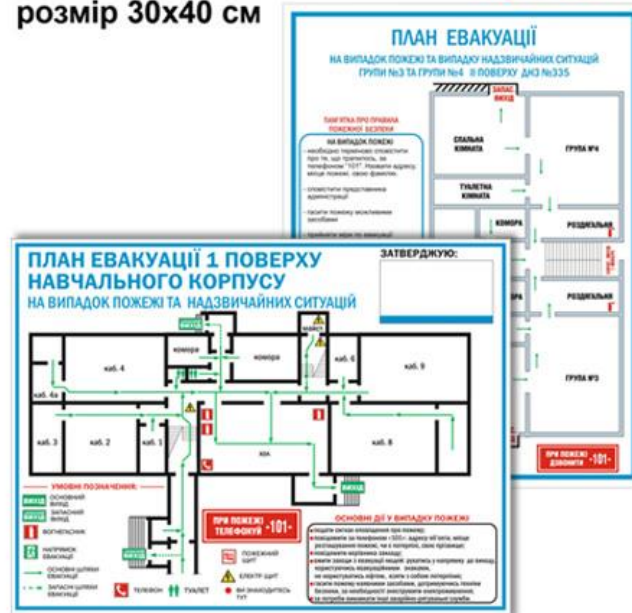


Рисунок 4.5 - План евакуації



Рисунок 4.6 - Укомплектований пожежний щит

Базова комплектація:

- Пінний і\або порошковий вогнегасник
- Пожежний лом
- Пожежний багор
- 2 сокири
- Відро

- Лопата
- Контейнер з піском

Шляхи евакуації ні в якому разі не можуть бути захащені, а виходи відкриватись без перешкод. Не можна улаштовувати на евакуаційних шляхах сходи, турнікети та все що ускладнює прохід людей. Не можна замикати, заварювати, забивати зовнішні евакуаційні двері. А також як евакуаційні виходи можуть використовуватись двері які ведуть одразу назовні, на сходовий майданчик або у вестибюль. Необхідно якісне зовнішнє освітлення. Між будинками та спорудами мають бути відповідні протипожежні розриви, які також мають бути не захащені. Всі працівники повинні пройти інструктаж, та в разі небезпеки діяти згідно заздалегідь розробленої інструкції. Також евакуацією керує керівництво та уповноважені ними особи.

Правила пожежної безпеки в Україні затверджені відповідним наказом Міністерства внутрішніх справ України 30.12.2014 № 1417.
<https://zakon.rada.gov.ua/laws/show/z0252-15#Text>.

Висновки

Правила евакуації та чітко сплановані алгоритми створені та затверджені для запобігання паніки та збереження здоров'я та життя людей на підприємствах. Спеціально призначені відповідальні за пожежну безпеку люди мають слідкувати за станом шляхів евакуації та проводити з людьми інструктажі з пожежної безпеки. За виникнення надзвичайної ситуації необхідна швидка та злагоджена евакуація. При необхідності люди повинні знати та вміти використовувати інструменти пожежного щита для звільнення виходу з приміщення. Сукупність виконання зазначених правил забезпечить безпеку персоналу при виникненні пожежі.

ВИСНОВКИ

Телемедицина є досить перспективним напрямком послуг охорони здоров'я. Є як плюси, у вигляді швидкості надання допомоги, безпека медика від можливо зараження якщо у пацієнта якась вірусна інфекція, як от сучасна пандемія COVID 19, швидка передачі аналіз, зберігання даних на великих цифрових сховищах, обмін даних між лікарем та пацієнтом. А також мінуси у вигляді недостатньої точності в діагностуванні, необхідності високого ступення захисту для захисту великих масивів персональних даних, хороше апаратне, комунікаційне та програмне забезпечення, а також покращення кваліфікації лікарів у користуванні ПК для спілкування з пацієнтами.

На мою думку телемедицина має досить високий потенціал. Але вона вимагає покращення комп'ютерної грамотності лікарів. А головне це потребує виділення величезних коштів на розвиток технологічної частини телемедицини. Коли ці вимоги будуть виконані тоді перед нашою системою охорони здоров'я відкриється безліч можливостей як в швидкості діагностування так і в плані своєї безпеки коли необхідно оглянути та перевірити аналізи пацієнта від якого можливо підхопити вірус. Різні компанії вже впроваджують окремі функції телемедицини в свої сервісах. Але без підтримки державою страхова та телемедицина не зможуть утворити єдину та добре працюючу системою, що негативно позначиться на досвілі лікарів та якості цих телемедичних послуг.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1.Що таке телемедицина.[Електронний ресурс] Режим доступу:<https://telemed.od.ua/ru/what-is-telemedicine>
- 2.Телемедицина в епоху COVID-19.[Електронний ресурс] Режим доступу:<http://elartu.tntu.edu.ua/handle/lib/34420>
- 3.Телемедицина по українськи. [Електронний ресурс] Режим доступу:<https://medstar.ua/ru/telemedicina-po-ukrainski-opyt-i-primery-vnedrenija/>
4. Як застосовувати телемедицину лікарям первинної ланки. [Електронний ресурс] Режим доступу:<https://moz.gov.ua/article/for-medical-staff/jak-zastosovuvati-telemedicinu-likarjam-pervinnoi-lanki-metodichni-rekomendacii>
- 5.Телемедицина в Україні. [Електронний ресурс] Режим доступу:<https://www.medsprava.com.ua/article/ru/1199-telemeditsina-v-ukraine-rus>
- 6.Що таке телемедицина та як зросла її роль в умовах пандемії. [Електронний ресурс] Режим доступу:<https://susplne.media/68648-telemedicina-tse-ta-rol-v-umovah-pandemii/>
7. Що таке телемедицина та як вона працює на Вінниччині. [Електронний ресурс] Режим доступу:<http://www.vin.gov.ua/departament-okhorony-zdorovia/22219-shcho-take-telemedytsyna-ta-iak-vona-pratsiuie-na-vinnychchyni>
- 8.Сучасні технології в телемедицині.[Електронний ресурс] Режим доступу:<http://elartu.tntu.edu.ua/handle/lib/23876>
- 9.Що таке телемедицина та які її можливості. [Електронний ресурс] Режим доступу:<https://www.apteka.ua/article/533622>
- 10.Що таке телемедицина, яку пообіцяв президент. Пояснює МОЗ. [Електронний ресурс] Режим доступу:<https://life.pravda.com.ua/health/2017/09/13/226425/>
- 11.Телемедицина. Світовий досвід та впровадження в Україні. [Електронний ресурс] Режим доступу:<https://emci.ua/statti/telemedytsyna-v-sviti-ta-ukrayini/>
- 12.Право на телемедицину. [Електронний ресурс] Режим доступу:<https://eba.com.ua/pravo-na-telemedytsynu/>

13.Телемедицина в світі та можливості її застосування. [Електронний ресурс] Режим доступу:<https://medinet.com.ua/telemedytsyna-y-sviti>

14.Телемедицина як форма якісного надання медичних послуг. [Електронний ресурс] Режим доступу:<https://ojs.tdmu.edu.ua/index.php/visnyk-gigieny/article/view/10492>

15.Телемедицина в Україні: реалії та перспективи.[Електронний ресурс] Режим доступу:<http://nbuviar.gov.ua>

16.Телемедицина. [Електронний ресурс] Режим доступу:<https://www.kmu.gov.ua/news/telemedicina-ce-absolyutno-nova-yakist-medichnoyi-poslugi-na-seli-gennadij-zubko>

17.Законодавча база телемедицини. [Електронний ресурс] Режим доступу:<https://zakon.rada.gov.ua/laws/show/z1400-15#Text>

18.Сучасні технології в телемедицині.[Електронний ресурс] Режим доступу:<http://elartu.tntu.edu.ua/handle/lib/23876>

19.Телемедицина. [Електронний ресурс] Режим доступу:<https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BB%D0%B5%D0%BC%D0%B5%D0%B4%D0%B8%D1%86%D0%B8%D0%BD%D0%B0>

20.eHealth. [Електронний ресурс] Режим доступу:<https://uk.wikipedia.org/wiki/EHealth>

21.Медичний моніторинг. [Електронний ресурс] Режим доступу:[https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D1%96%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3_\(%D0%BC%D0%B5%D0%B4%D0%B8%D1%86%D0%B8%D0%BD%D0%B0\)](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D1%96%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3_(%D0%BC%D0%B5%D0%B4%D0%B8%D1%86%D0%B8%D0%BD%D0%B0))

22.Законодавча база телемедицини. [Електронний ресурс] Режим доступу:<https://zakon.rada.gov.ua/rada/show/v0269282-07#Text>

23.Перше застосування телемедицини в Україні. [Електронний ресурс] Режим доступу:http://nbuv.gov.ua/j-pdf/Ujtm_2012_10_1_5.pdf

24.Законодавча база телемедицини. [Електронний ресурс] Режим доступу:<https://zakon.rada.gov.ua/laws/show/2206-19#Text>

25.Телемедична система Telemed24. [Електронний ресурс] Режим доступу:<https://telemed24.ua/>

- 26.Право на телемедицину. [Електронний ресурс] Режим доступу:<https://platforma-msb.org/pravo-na-telemedytsynu/>
- 27.Телемедицина. [Електронний ресурс] Режим доступу:<https://euroderm.com.ua/ru/methods/telemedicina.html>
- 28.Телемедицина в Україні та світі. Можливості та перспективи розвитку. <https://euroderm.com.ua/ru/methods/telemedicina.html>
- 29.Телемедицина. [Електронний ресурс] Режим доступу:<https://euromed.ua/trends/telemedicina>
- 30.Телемедицина. [Електронний ресурс] Режим доступу:<https://www.mtb.ua/ru/telemedicine>
- 31.Центр телемедичних технологій. [Електронний ресурс] Режим доступу:<https://gpmu.org/treatment/telemedicine/>
32. Чим насправді займається телемедицина. [Електронний ресурс] Режим доступу:<https://ain.ua/special/telemedicine-faq/>
- 33.Телемедицина. [Електронний ресурс] Режим доступу:<https://www.europeanlung.org/assets/files/ru/publications/telemedicine-ru.pdf>
- 34.Телемедицина стає musthaveдля страхових компаній. [Електронний ресурс] Режим доступу:<https://mind.ua/ru/publications/20212751-telemedicina-stanovitsya-must-have-dlya-strahovyh-v-ukraine>
- 35.Телемедицина. [Електронний ресурс] Режим доступу:<https://uniqa.ua/blog/telemedytsyna-oglyad-eksperta-pwc-ukraina/>
- 36.Телемедицина в Україні. [Електронний ресурс] Режим доступу:<https://www.medsprava.com.ua/article/ru/1199-telemeditsina-v-ukraine>
- 37.Лікар в смартфоні. [Електронний ресурс] Режим доступу:<https://happy monday.ua/ru/5-prichin-pochemu-kompanijam-nuzhna-telemedicina>
- 38.Послуги телемедицини від VISA.[Електронний ресурс] Режим доступу:https://www.visa.com.ua/ru_UA/pay-with-visa/promotions/telemedicine.html
- 39.Сучасні телемедичні системи. [Електронний ресурс] Режим доступу:https://web.posibnyky.vntu.edu.ua/firen/3zlepko_osnovy_biomedychnogo_radioelektronного_aparatobuduvannya/13.html

40. Телекардіологія. [Електронний ресурс] Режим доступу:http://pdf.lib.vntu.edu.ua/books/Vladzh_2012_116.pdf
41. Медична інформатика. [Електронний ресурс] Режим доступу:<https://medinformatic.umsa.edu.ua/storage/resources/docs/NLDdqmB4C2fxUxt7M7OYq2pAXlpVzAWkhLTQaiHg.pdf>
42. Стандарти передачі медичної інформації в телемедичних системах. [Електронний ресурс] Режим доступу:www.hups.mil.gov.ua%2Fperiodic-app%2Farticle%2F10630%2Fsoi_2013_2_59.pdf&usg=AOvVaw2MpOBy72qLNglGEVR-Q226
43. Медико-фізіологічні дослідження. [Електронний ресурс] Режим доступу:<https://www.inneco.org>
44. Медична інформатика. [Електронний ресурс] Режим доступу:<https://med.cc.ua/wp-content/uploads/2014/>
45. Клінічна інформатика. [Електронний ресурс] Режим доступу:http://kit-journal.com.ua/uk/article_uk.html?article/013-010
46. С++. [Електронний ресурс] Режим доступу:<https://ravesli.com/uroki-cpp/>
47. Основи програмування С++. [Електронний ресурс] Режим доступу:<http://cppstudio.com/>
48. Програмування на С++. [Електронний ресурс] Режим доступу:<https://acode.com.ua/uroki-po-cpp/>
49. Основи програмування на С++, детальні інструкції, приклади простих кодів. [Електронний ресурс] Режим доступу: <https://purecodecpp.com>
50. Портал з кодами та поясненнями. [Електронний ресурс] Режим доступу:<https://code-live.ru/tag/cpp-manual>
51. Огляд телемедичних технологій. [Електронний ресурс] Режим доступу:<http://elartu.tntu.edu.ua/handle/lib/30417>
52. Дослідження проблем розвитку телемедицини в Україні. Електронний ресурс] Режим доступу: <http://elartu.tntu.edu.ua/handle/lib/30416>

ДОДАТКИ

УДК 004.415.5

В. Крот

(Тернопільський національний технічний університет імені Івана Пулюя)

ОГЛЯД ТЕЛЕМЕДИЧНИХ ТЕХНОЛОГІЙ

UDC 004.415.5

V. Krot

(Ternopil Ivan Puluj National Technical University, Ukraine)

REVIEW OF TELEMEDICINE TECHNOLOGIES

У сучасному світі телемедицина посіла важливе місце у системі охорони здоров'я. Європейські країни давно запровадили цю систему, наразі і в Україні починається запровадження телемедицини і законів потрібних для її функціонування.

Лікувально-діагностичні консультації - зв'язок організується за схемою "точка- точка", що надає можливість лікуваному лікарю обговорювати пацієнта з консультантом-спеціалістом. Першими в Україні тестовий телемедичний сеанс провели на Рівненщині. Медики, які працюють у новобудованій лікарській амбулаторії загальної практики-сімейної медицини с. Заборозь Рівненського району 22 лютого спілкувалися з фахівцями Рівненської обласної клінічної лікарні. Консультація пройшла успішно і наразі розпочато підготовку до системної роботи, яку зможуть розпочати за 1-1,5 місяці.

До цього виду телемедичних технологій відносять також телемоніторинг. Телеметрія функціональних показників налаштовується за рахунок зв'язку "багато точок - точка", коли дані (рентгенограми, ЕКГ, епікризи, тощо) багатьох пацієнтів лікувально-профілактичного закладу, району передаються в консультативний центр.

Інформаційна система "Медицинський Інтернет - консультації" - комплексна інтерактивна система визначення стану здоров'я людини. Система містить тести, за якими можна визначити показники соматичного, психофізіологічного та психічного компоненти здоров'я. Мета цієї інформаційної системи - долікарське самопознання та надання консультацій користувачеві.

Також для зберігання документів використовуються електронні бази даних і регулярно оновлюються (у закладах які мають необхідне обладнання).

У планах також є оснащення машин швидкої допомоги планшетами щоб лікар приїхавши на виклик міг прямо на місці обговорити стан пацієнта зі спеціалістом якщо є сумніви на рахунок його стану та діагнозу.

Завдяки ініціативі та фінансовій підтримці енергетичної компанії ДТЕК, яка є підписантом Меморандуму, медичні установи міст її діяльності обладнані та підключені до єдиної національної телемедичної мережі, щоб забезпечити доступну та безпечну медичну допомогу в будь-який момент, незалежно від того, де знаходиться пацієнт. Так, у грудні 2013 р. Львівська обласна дитяча клінічна лікарня «ОХМАДИТ» стала п'ятим медичним закладом Львівської області, котрий завдяки компанії ДТЕК має можливість за допомогою відеозв'язку в режимі онлайн консультуватися із провідними фахівцями зарубіжних клінік. Необхідним аудіо та відеообладнанням оснащені три операційні ОХМАДИТу. Додатково мережа доповнюється медичними закладами також завдяки фінансуванню партнерами по проєкту. Зокрема, завдяки фінансовій підтримці Благодійного фонду Ріната Ахметова «Розвиток України» до проєкту приєднався і Львівський обласний центр.

5 липня 2016 р. відбувся перший телевізійно-презентаційний сеанс «Телемедицина» в Запорізькій обласній лікарні та Енергодарській медико-санітарній частині № 1. Телевізійний трьохмагістрал - Запоріжжя, Енергодаром та Києвом - провів головний лікар Київського центру клінічної телемедицини В. Осташко.

Прогрес у запровадженні телемедичних технологій йде досить повільно, але опираючись на досвід Європейських країн зрозуміло що це необхідні кроки які допоможуть у майбутньому.

УДК 004.415.5

В. Крот

(Тернопільський національний технічний університет імені Івана Пулюя)

ДОСЛІДЖЕННЯ ПРОБЛЕМ РОЗВИТКУ ТЕЛЕМЕДИЦИНИ В УКРАЇНІ

UDC 004.415.5

V. Krot

(Ternopil Ivan Puluj National Technical University, Ukraine)

RESEARCH OF TELEMEDICINE DEVELOPMENT PROBLEMS IN UKRAINE

Головною метою реформи медичної галузі в Україні є поліпшення здоров'я населення завдяки забезпеченню рівного й справедливого доступу всіх громадян до медичних послуг належної якості. Міжнародний досвід показує, що одним із кращих та перспективних способів підвищення рівня медичного обслуговування, розширення можливостей щодо доступності та якості медичних послуг населенню, особливо для тих груп, що проживають на сільських територіях, є запровадження телемедицини.

Якщо в 1960-х роках телемедицина розумілася, як передача медичних даних пацієнта по телефону і радіо, то в 90-ті роки за допомогою інтернету і нових телекомунікаційних - інформаційних технологій. Але, на даний момент телемедицину в Україні практично не використовують, і цей напрям немає необхідних ресурсів для розвитку.

Одною з проблем є навіть не технології та економічна ситуація. Проблема полягає у неосвідчаності населення, а також частини лікарів. За даними проведених дослідів близько половини лікарів не можуть швидко увімкнути ноутбук, підключитись до інтернету та налаштувати передачу даних чи конференцію. Ще однією перевагою було б друкування рецептів, без нерозбірливого лікарського почерку. Також багато людей похилого віку, котрим часто потрібна невідкладна допомога через проблеми з серцем або загострення хронічних хвороб через вік. Людям похилого віку доволі важко призначити користуватись комп'ютерами або навіть мобільними телефонами.

Наступною проблемою стають телекомунікації. Якщо у великих містах це не є проблемою, то в маленьких містах із зв'язку можна кіба що знайти слабкий сигнал мобільного оператора, і навіть він дуже слабкий. В таких населених пунктах не вийде організувати стабільну або безпечну передачу даних.

Третьою проблемою стає економічна ситуація в Україні. На даний момент більшість коштів потрібні на формування армії та у енергетичній галузі, тому на даний момент проблемами телемедицини майже не займаються і не виділяють коштів. Однак Президентська програма реформ до 2020 р. та проєкт Коаліційної угоди проукраїнської більшості в парламенті називають реформу системи охорони здоров'я та, зокрема, розвиток телемедицини серед першочергових завдань. Хоча на інвестиції не такі вже й значні: Експертами підраховано, щоб накрити системою телемедицини всю Україну потрібна зовсім незначна, як для масштабів держави, сума – трохи більше 8 млн грн. А запровадження телемедицини дає змогу заощадити до 40 % витрат на охорону здоров'я.

При наданні телемедичних послуг має бути забезпечено збереження особистості, лікарської таємниці та інших таємниць, передбачених законодавством України, а також конфіденційності персональних даних. Тобто при передачі телемедичні дані потрібно шифрувати.

Ще у 2007 р. було створено Державний клінічний науково-практичний центр телемедицини МОЗ України – єдиний спеціалізований заклад охорони здоров'я, створений для надання

Сервер

main.cpp

```
#include <iostream>
#include <QCoreApplication>
#include "connectionhandler.h"

int main(int argc, char **argv)
{
    QCoreApplication app(argc, argv);
    ConnectionHandler netmanager;
    std::cout << "This is Telehealth server console.\n"
                "Server started at localhost.\n";
    netmanager.startServer();

    return app.exec();
}
```

connectionhandler.h

```
#ifndef CONNECTIONHANDLER_H
#define CONNECTIONHANDLER_H

#include <QTcpServer>
#include <QTcpSocket>

#include <QMap>
#include <QString>

#include "../Data/authanswer.h"
#include "../Data/userlist.h"
#include "../Data/userdata.h"
#include "../Data/message.h"
#include "../Data/filedata.h"
#include "../3rdparty/simplecrypt.h"
```

```
#include "serverdatabasehelper.h"
```

```
class ConnectionHandler : public QObject
```

```
{
```

```
    Q_OBJECT
```

```
public:
```

```
    explicit ConnectionHandler(QObject *parent = nullptr);
```

```
    void startServer();
```

```
    void stopServer();
```

```
    bool isListening() const;
```

```
    void write(QTcpSocket *socket, const BaseData &from);
```

```
    ~ConnectionHandler();
```

```
private slots:
```

```
    void on_New_Connection();
```

```
    void on_Client_Disconnection();
```

```
    void on_Socket_Error(QAbstractSocket::SocketError);
```

```
    void on_Socket_Ready_Read();
```

```
private:
```

```
    void readAuthRequest(QTcpSocket *socket, QDataStream &readStream);
```

```
    void readMessage(QTcpSocket *socket, QDataStream &readStream);
```

```
    void readFile(QTcpSocket *socket, QDataStream &readStream);
```

```
    void writeAuthAnswer(QTcpSocket *socket, bool answer, quint64 key =  
0);
```

```
    void writeUserList(QTcpSocket *socket, const QList<UserData>&lst);
```

```
    void sendOutMessage(QTcpSocket *sender, const Message &msg);
```

```
    void sendOutFile(QTcpSocket *sender, const FileData &filedata);
```

```
    void writeAboutNewConnection(const UserData &connectedUser);
```

```
    void writeAboutUserDisconnection(const UserData &disconnectedUser);
```

```
    QString getIPv4AddrString(QTcpSocket *socket);
```

```
void connectSocketSignals(QTcpSocket *socket);
```

```
private:
```

```
    QMap<UserData, QTcpSocket*> clients;  
    QTcpServer *serv;  
    ServerDatabaseHelper DBhelper;  
    SimpleCrypt *crypto;  
    QMap<QTcpSocket*, quint64> clientsKey;
```

```
};
```

```
#endif // CONNECTIONHANDLER_H
```

```
connectionhandler.cpp
```

```
#include "connectionhandler.h"
```

```
#include <QBuffer>
```

```
ConnectionHandler::ConnectionHandler(QObject *parent) :
```

```
    QObject(parent),  
    serv(new QTcpServer(this)),  
    crypto(new SimpleCrypt)
```

```
{
```

```
    DBhelper.InitializeDatabase();  
    connect(serv, SIGNAL(newConnection()),  
           this, SLOT(on_New_Connection()));
```

```
        crypto->setCompressionMode(SimpleCrypt::CompressionAlways); //завжди  
стискати дані
```

```
        crypto->setIntegrityProtectionMode(SimpleCrypt::ProtectionHash);
```

```
//належним чином захистити цілісність даних
```

```
}
```

```
ConnectionHandler::~~ConnectionHandler()
```

```
{
```

```
    delete serv;
```

```

}

void ConnectionHandler::startServer()
{
    serv->listen(QHostAddress::Any, 8001);
}

void ConnectionHandler::stopServer()
{
    serv->close();
    clients.clear();
}

void ConnectionHandler::on_New_Connection()
{
    auto client = serv->nextPendingConnection();
    connectSocketSignals(client);
    clientsKey.insert(client, Q_UINT64_C(0x0c2ad5a5acb9f007));
    qDebug() << "New connection try from" << getIPv4AddrString(client) <<
    "...";
}

void ConnectionHandler::connectSocketSignals(QTcpSocket *socket)
{
    connect(socket, SIGNAL(readyRead()),
            this, SLOT(on_Socket_Ready_Read()));

    connect(socket, SIGNAL(disconnected()),
            this, SLOT(on_Client_Disconnection()));

    connect(socket, SIGNAL(error(QAbstractSocket::SocketError)),
            this, SLOT(on_Socket_Error(QAbstractSocket::SocketError)));
}

void ConnectionHandler::on_Client_Disconnection()
{

```

```

    auto disconnectedSocket = static_cast<QTcpSocket *>(sender());
    auto username = clients.key(disconnectedSocket);
    clients.remove(username);
    username.setType(Disconnection);
    writeAboutUserDisconnection(username);
}

bool ConnectionHandler::isListening() const
{
    return serv->isListening();
}

void ConnectionHandler::on_Socket_Error(QAbstractSocket::SocketError)
{
    auto socket = static_cast<QTcpSocket *>(sender());
    qDebug() << getIPv4AddrString(socket) << socket->errorString();
}

QString ConnectionHandler::getIPv4AddrString(QTcpSocket *socket)
{
    QString IPv4IPv6 = socket->peerAddress().toString();
    QString IPv4 = IPv4IPv6.right(IPv4IPv6.size() - 7);
    return IPv4;
}

void ConnectionHandler::write(QTcpSocket *socket, const BaseData &data)
{
    QByteArray datagram;
    QDataStream writeStream(&datagram, QIODevice::WriteOnly);
    writeStream.setVersion(QDataStream::Qt_5_12);
    writeStream << quint32(0) << qint8(data.type());

    QBuffer cryptoBuffer;
    cryptoBuffer.open(QIODevice::WriteOnly);
    QDataStream cryptoStream(&cryptoBuffer);
    //записуємо дані у буфер

```



```

    cryptoStream.setVersion(QDataStream::Qt_5_12);
    cryptoStream << data;
    crypto->setKey(clientsKey.value(socket));
    QByteArray cryptoData = crypto-
>encryptToArray(cryptoBuffer.data());
    // якщо не було помилки при шифруванні
    if (crypto->lastError() == SimpleCrypt::ErrorNoError) {
        writeStream << cryptoData;
//        qDebug() << datagram.size();
//        qDebug() << quint32(quint32(datagram.size()) -
sizeof(quint32));
        writeStream.device()->seek(0);
        writeStream << quint32(quint32(datagram.size()) -
sizeof(quint32)); // записуємо розмір переданих даних
        socket->write(datagram);
        socket->waitForBytesWritten();
    }
    cryptoBuffer.close();
}

void ConnectionHandler::on_Socket_Ready_Read()
{
    auto socket = static_cast<QTcpSocket *>(sender());
    QDataStream readStream(socket);
    readStream.setVersion(QDataStream::Qt_5_12);
    quint32 sizeRead;
    qint8 type;
    readStream >> sizeRead;
    qDebug() << "ConnectionHandler::on_Socket_Ready_Read() sizeToRead: "
<< sizeRead;

    /* очікуємо поки всі пакети даних прийдуть */
    while(socket->bytesAvailable() < sizeRead){
        if (!socket->waitForReadyRead())
        {
            qDebug() << "waitForReadyRead() timed out";

```

```

        return;
    }
}
qDebug() << "bytesAvailable() =" << socket->bytesAvailable();

readStream >> type;

QByteArray cypherData;
readStream >> cypherData;

crypto->setKey(clientsKey.value(socket));
QByteArray plainData = crypto->decryptToByteArray(cypherData);
if (crypto->lastError() != SimpleCrypt::ErrorNoError) {
    qDebug() << "ConnectionHandler::on_Socket_Ready_Read() ->
crypto.lastError() = " << crypto->lastError();
    return;
}

QBuffer buffer(&plainData);
buffer.open(QIODevice::ReadOnly);
QDataStream stream(&buffer);
stream.setVersion(QDataStream::Qt_5_12);

switch (type) {
case DataType::AuthRequest:
    readAuthRequest(socket, stream);
    break;
case DataType::Msg:
    readMessage(socket, stream);
    break;
case DataType::File:
    readFile(socket, stream);
    break;
default:
    qDebug() << "Unknown type of DataType received: " << type;
    break;
}

```

```

    }
    buffer.close();
}

void ConnectionHandler::readAuthRequest(QTcpSocket *socket, QDataStream
&readStream)
{
    UserData read;
    readStream >> read;

    DatabaseResponse *response =
DBhelper.Select(ServerDatabaseHelper::TABLE_USERS_NAME,
QList<QString>{ServerDatabaseHelper::ARG_USERS_ID},
ServerDatabaseHelper::ARG_USERS_LOGIN+"="+read.getUsername()+"' AND
"+ServerDatabaseHelper::ARG_USERS_PASSWORD+"="+read.getPassword()+"'
");
    qDebug() << response->GetRecord().value(0);

    // TODO авторизацію, зчитування користувача і паролю з БД

    if (!response->IsError())
    {
        read.setType(DataType::NewConnection);

        quint64 cryptoKey = QRandomGenerator::global()->generate64();
        writeAuthAnswer(socket, true, cryptoKey);
        clientsKey.insert(socket, cryptoKey);
        writeUserList(socket, clients.keys());
        writeAboutNewConnection(read);

        clients.insert(read, socket);
        qDebug() << getIPv4AddrString(socket) << "authentication
successful. Accepted.";
    }
    else

```

```
{
    writeAuthAnswer(socket, false);
    qDebug() << getIPv4AddrString(socket) << "authentication
unsuccessful. Aborted.";
    qDebug() << response->GetError();
}
}
```

```
void ConnectionHandler::readMessage(QTcpSocket *socket, QDataStream
&readStream)
```

```
{
    Message msg;
    readStream >> msg;
    qDebug().noquote() << "ConnectionHandler::readMessage() reached a
msg: " << msg.toString();
    sendOutMessage(socket, msg);

    //TODO збереження повідомлень в БД
}
```

```
void ConnectionHandler::readFile(QTcpSocket *socket, QDataStream
&readStream)
```

```
{
    FileData fileData;
    readStream >> fileData;
    qDebug().noquote() << "ConnectionHandler::readFile() reached a file:
" << fileData.toString();
    sendOutFile(socket, fileData);
}
```

```
void ConnectionHandler::writeAuthAnswer(QTcpSocket *socket, bool answer,
quint64 key)
```

```
{
    AuthAnswer ans(answer, key);
    write(socket, ans);
}
```

```
}
```

```
void ConnectionHandler::writeUserList(QTcpSocket *socket, const  
QList<UserData>&lst)
```

```
{  
    UserList participants(lst);  
    write(socket, participants);  
}
```

```
void ConnectionHandler::writeAboutNewConnection(const UserData  
&connectedUser)
```

```
{  
    for (auto &user : clients.values()) write(user, connectedUser);  
}
```

```
void ConnectionHandler::writeAboutUserDisconnection(const UserData  
&disconnectedUser)
```

```
{  
    for (auto &user : clients.values()) write(user, disconnectedUser);  
}
```

```
void ConnectionHandler::sendOutMessage(QTcpSocket *sender, const Message  
&msg)
```

```
{  
    auto senderUsername = clients.key(sender).getUsername();  
    for (auto &user : clients.values())  
        /* не відпавляти повідомлення самому собі */  
        if (clients.key(user).getUsername() != senderUsername)  
            write(user, msg);  
}
```

```
void ConnectionHandler::sendOutFile(QTcpSocket *sender, const FileData  
&fileData)
```

```
{  
  
    if (fileData.getReceiver() == "") { //відправити файл усім
```

```

        //      auto senderUsername =
clients.key(sender).getUsername();
        auto senderUsername = fileData.getSender();
        for (auto &user : clients.values())
            /* не відправляти файл собі назад*/
            if (clients.key(user).getUsername() != senderUsername)
                write(user, fileData);
    } else {
        //      write(clients.key)
        for (auto &user : clients.keys())
            if (user.getUsername() == fileData.getReceiver())
                write(clients.value(user), fileData);
    }
}

```

serverdatabasehelper.h

```
#pragma once
```

```
#ifndef SERVERDATABASEHELPER_H
```

```
#define SERVERDATABASEHELPER_H
```

```
#include "../Data/databasehelper.h"
```

```
//#include "../Data/user.h"
```

```
#include <QCoreApplication>
```

```
class User;
```

```
class Doctor;
```

```
class ServerDatabaseHelper: public DatabaseHelper
```

```
{
```

```
// void RefreshWorkTimes(int userId, QList<WorkTime> workTimesNew);
```

```
// QList<WorkTime> GetWorkTimes(int doctorId);
```

```
// int GetUserId(User* user);
```

```
// int GetDoctorId(Doctor* doctor);
```

```

//    int GetDoctorId(int userId);
public:
    const static QString DB_NAME;

    const static QString TABLE_USERS_NAME;
    const static QString ARG_USERS_ID;
    const static QString ARG_USERS_LOGIN;
    const static QString ARG_USERS_PASSWORD;
    const static QString ARG_USERS_SURNAME;
    const static QString ARG_USERS_NAME;
    const static QString ARG_USERS_FATHER_NAME;
    const static QString ARG_USERS_PHONE_NUMBER;
    const static QString ARG_USERS_PRIVILEGIES;

    const static QString TABLE_DOCTORS_NAME;
    const static QString ARG_DOCTORS_ID;
    const static QString ARG_DOCTORS_USER_ID;
    const static QString ARG_DOCTORS_SPECIALITY_ID;

    const static QString TABLE_SPECIALITIES_NAME;
    const static QString ARG_SPECIALITIES_ID;
    const static QString ARG_SPECIALITIES_SPECIALITY;

    ServerDatabaseHelper();
    void InitializeDatabase() override;

/*
    void SaveUser(User* user);
    QList<User> GetUsers(QString condition);
    void DeleteUser(User* user);

    void SaveSpeciality(QString spec);
    QList<QString> GetSpecialities(QString condition);
    void DeleteSpeciality(QString spec);
    int GetSpecialityId(QString spec);

```

```

    void SaveDoctor(Doctor* doctor);
    QList<Doctor> GetDoctors(QString condition, User *user=nullptr);
    QList<Doctor> GetDoctors(User *user);
    void DeleteDoctor(Doctor* doctor);
*/
};

```

```
#endif // SERVERDATABASEHELPER_H
```

```
serverdatabasehelper.cpp
```

```
#include "serverdatabasehelper.h"
```

```

const QString ServerDatabaseHelper::DB_NAME =
"TelehealthDB";

const QString ServerDatabaseHelper::TABLE_USERS_NAME =
"users";
const QString ServerDatabaseHelper::ARG_USERS_ID =
"id";
const QString ServerDatabaseHelper::ARG_USERS_LOGIN =
"login";
const QString ServerDatabaseHelper::ARG_USERS_PASSWORD =
"password";
const QString ServerDatabaseHelper::ARG_USERS_SURNAME =
"surname";
const QString ServerDatabaseHelper::ARG_USERS_NAME =
"name";
const QString ServerDatabaseHelper::ARG_USERS_FATHER_NAME =
"father_name";
const QString ServerDatabaseHelper::ARG_USERS_PHONE_NUMBER =
"phone_number";
const QString ServerDatabaseHelper::ARG_USERS_PRIVILEGIES =
"privileges";

```



```

const QString ServerDatabaseHelper::TABLE_DOCTORS_NAME           =
"doctors";
const QString ServerDatabaseHelper::ARG_DOCTORS_ID              =
"id";
const QString ServerDatabaseHelper::ARG_DOCTORS_USER_ID        =
"user_id";
const QString ServerDatabaseHelper::ARG_DOCTORS_SPECIALITY_ID  =
"speciality_id";

```

```

const QString ServerDatabaseHelper::TABLE_SPECIALITIES_NAME    =
"specialities";
const QString ServerDatabaseHelper::ARG_SPECIALITIES_ID        =
"id";
const QString ServerDatabaseHelper::ARG_SPECIALITIES_SPECIALITY =
"speciality";

```

```

ServerDatabaseHelper::ServerDatabaseHelper():DatabaseHelper(DB_NAME)
{
}

```

```

void ServerDatabaseHelper::InitializeDatabase()
{

```

```

    DatabaseHelper::InitializeDatabase();

```

```

    CreateTable(TABLE_USERS_NAME, QList<QString>{ARG_USERS_ID + " INTEGER
PRIMARY KEY",

```

```

ARG_USERS_LOGIN + "

```

```

VARCHAR(20) NOT NULL UNIQUE",

```

```

ARG_USERS_PASSWORD + "

```

```

VARCHAR(20) NOT NULL",

```

```

ARG_USERS_SURNAME + "

```

```

VARCHAR(30)",

```

```

ARG_USERS_NAME + "

```

```

VARCHAR(30)",

```

```

ARG_USERS_FATHER_NAME +

```

```

" VARCHAR(30)",

```

```

ARG_USERS_PHONE_NUMBER +
" VARCHAR(13)",
ARG_USERS_PRIVILEGIES +
" INTEGER NOT NULL"});

CreateTable(TABLE_DOCTORS_NAME, QList<QString>{ARG_DOCTORS_ID + "
INTEGER PRIMARY KEY",
ARG_DOCTORS_USER_ID +
" INTEGER NOT NULL UNIQUE",
ARG_DOCTORS_SPECIALITY_ID + " INTEGER NOT NULL"});

CreateTable(TABLE_SPECIALITIES_NAME,
QList<QString>{ARG_SPECIALITIES_ID + " INTEGER PRIMARY KEY",
ARG_SPECIALITIES_SPECIALITY + " VARCHAR(30) NOT NULL UNIQUE"});
}
/*
void ServerDatabaseHelper::SaveUser(User* user)
{
Update(TABLE_USERS_NAME, QList<QString>{ARG_USERS_CARD_ID,
ARG_USERS_LOGIN, ARG_USERS_PASSWORD, ARG_USERS_SURNAME, ARG_USERS_NAME,
ARG_USERS_FATHER_NAME, ARG_USERS_DATE_OF_BIRTHD,
ARG_USERS_ADDRESS, ARG_USERS_PHONE_NUMBER, ARG_USERS_PRIVILEGIES},
QList<QString>{user->GetCardId(), user->GetLogin(), user-
>GetPassword(), user->GetSurname(), user->GetName(), user-
>GetFatherName(),
user->GetAddress(), user->GetPhoneNumber(),
QString::number(user->GetPrivileges())},
ARG_USERS_LOGIN+"="+user->GetLogin()+"");
if(databaseQuery.numRowsAffected() == 0)
{
Select(TABLE_USERS_NAME, QList<QString>{ARG_USERS_LOGIN},
ARG_USERS_LOGIN+"="+user->GetLogin()+"");
if(!databaseQuery.first())
{

```

```

        Insert(TABLE_USERS_NAME, QList<QString>{ARG_USERS_CARD_ID,
ARG_USERS_LOGIN, ARG_USERS_PASSWORD, ARG_USERS_SURNAME, ARG_USERS_NAME,
        ARG_USERS_FATHER_NAME, ARG_USERS_DATE_OF_BIRTHD,
ARG_USERS_ADDRESS, ARG_USERS_PHONE_NUMBER, ARG_USERS_PRIVILEGIES},
        QList<QList<QString>>{{user->GetCardId(), user-
>GetLogin(), user->GetPassword(), user->GetSurname(), user->GetName(),
user->GetFatherName(),
        user->GetAddress(), user->GetPhoneNumber(),
QString::number(user->GetPrivileges())}}});
    }
}
QList<User> ServerDatabaseHelper::GetUsers(QString condition)
{
    DatabaseResponse *response = Select(TABLE_USERS_NAME,
QList<QString>{ARG_USERS_CARD_ID, ARG_USERS_LOGIN, ARG_USERS_PASSWORD,
ARG_USERS_SURNAME, ARG_USERS_NAME,
        ARG_USERS_FATHER_NAME, ARG_USERS_DATE_OF_BIRTHD,
ARG_USERS_ADDRESS, ARG_USERS_PHONE_NUMBER, ARG_USERS_PRIVILEGIES},
        condition);

    QList<User> users;
    if(response->IsError()) return users;

    QSqlRecord record = response->GetRecord();
    QSqlQuery* databaseQuery = new QSqlQuery(this->databaseQuery);
    //qDebug()<<databaseQuery.numRowsAffected();
    while(databaseQuery->next())
    {
        User user(databaseQuery-
>value(record.indexOf(ARG_USERS_LOGIN)).toString(), databaseQuery-
>value(record.indexOf(ARG_USERS_PASSWORD)).toString(),
        databaseQuery-
>value(record.indexOf(ARG_USERS_SURNAME)).toString(), databaseQuery-
>value(record.indexOf(ARG_USERS_NAME)).toString(),

```

```

        databaseQuery-
>value(record.indexOf(ARG_USERS_ADDRESS)).toString(), databaseQuery-
>value(record.indexOf(ARG_USERS_PHONE_NUMBER)).toString(),
        databaseQuery-
>value(record.indexOf(ARG_USERS_PRIVILEGIES)).toInt());
        user.ConnectCard(databaseQuery-
>value(record.indexOf(ARG_USERS_CARD_ID)).toString());
        users.append(user);
    }
    return users;
}

void ServerDatabaseHelper::DeleteUser(User* user)
{
    for(int i = 0; i < user->privilegesCount; i++)
        user->RemovePrivilege(((User::Privileges)i));
    Delete(TABLE_USERS_NAME, ARG_USERS_LOGIN+"='"+user->GetLogin()+"'");
}

void ServerDatabaseHelper::SaveSpeciality(QString spec)
{
    if(GetSpecialities(ARG_SPECIALITIES_SPECIALITY+"='"+spec+"'").length()==0
    )
        Insert(TABLE_SPECIALITIES_NAME,
QList<QString>{ARG_SPECIALITIES_SPECIALITY},
                QList<QList<QString>>{{spec}});
    else
        CreateDatabaseResponse("Ця спеціальність уже існує!");
}

QList<QString> ServerDatabaseHelper::GetSpecialities(QString condition)
{
    DatabaseResponse *response = Select(TABLE_SPECIALITIES_NAME,
QList<QString>{ARG_SPECIALITIES_SPECIALITY}, condition);
    QList<QString> specs;
    if(response->IsError()) return specs;
    QSqlRecord record = response->GetRecord();
    while(databaseQuery.next())

```

```

specs.append(databaseQuery.value(record.indexOf(ARG_SPECIALITIES_SPECIALI
TY)).toString());
    return specs;
}
void ServerDatabaseHelper::DeleteSpeciality(QString spec)
{
    Delete(TABLE_SPECIALITIES_NAME,
ARG_SPECIALITIES_SPECIALITY+"='"+spec+"'");
}
void ServerDatabaseHelper::SaveDoctor(Doctor* doctor)
{
    //Search user ID and speciality ID
    int userId, specId;

//    userId = GetUserId(doctor->GetUser());
//    if(userId == -1) return;

//    specId = GetSpecialityId(doctor->GetSpeciality());
//    if(specId == -1) return;

//Create or update doctor
DatabaseResponse* response = Update(TABLE_DOCTORS_NAME,
QList<QString>{ARG_DOCTORS_SPECIALITY_ID},
        QList<QString>{QString::number(specId)},
        ARG_DOCTORS_USER_ID+"='"+QString::number(userId)+"'");
    if(databaseQuery.numRowsAffected() == 0)
    {
        response = Select(TABLE_DOCTORS_NAME,
QList<QString>{ARG_DOCTORS_SPECIALITY_ID},
ARG_DOCTORS_USER_ID+"='"+QString::number(userId)+"'");
        if(!databaseQuery.first())
        {
            response = Insert(TABLE_DOCTORS_NAME,
QList<QString>{ARG_DOCTORS_USER_ID, ARG_DOCTORS_SPECIALITY_ID},

```

```

        QList<QList<QString>>{{QString::number(userId),
QString::number(specId)}}});
    }
}

```

```

    if(response->IsError()) return;

```

```

//    RefreshWorkTimes(userId, doctor->GetWorkTimes());
}

```

```

QList<Doctor> ServerDatabaseHelper::GetDoctors(QString condition, User*
user)

```

```

{
    DatabaseResponse *response = Select(TABLE_DOCTORS_NAME,
QList<QString>{ARG_DOCTORS_ID, ARG_DOCTORS_USER_ID,
ARG_DOCTORS_SPECIALITY_ID}, condition);
    QSqlRecord record = response->GetRecord();
    QSqlQuery* databaseQuery = new QSqlQuery(this->databaseQuery);
    QList<Doctor> doctors;
    while(databaseQuery->next())
    {
        int userId = databaseQuery-
>value(record.indexOf(ARG_DOCTORS_USER_ID)).toInt();
        int specId = databaseQuery-
>value(record.indexOf(ARG_DOCTORS_SPECIALITY_ID)).toInt();
        int doctorId = databaseQuery-
>value(record.indexOf(ARG_DOCTORS_ID)).toInt();
        User* thisUser;
        if(user==nullptr)
        {
            QList<User> users =
GetUsers(ARG_USERS_ID+"='"+QString::number(userId)+"'");
            if(users.length() > 0)
                thisUser = new User(users[0]);
            else
                continue;

```

```

        }else thisUser = user;

        QString specName;
        QList<QString> specNames =
GetSpecialities(ARG_SPECIALITIES_ID+"='"+QString::number(specId)+"'");
        if(specNames.length() > 0)
            specName = specNames[0];
        else
            continue;

        QList<WorkTime> workTimes = GetWorkTimes(doctorId);

//        doctors.append(*Doctor::CreateDoctor(thisUser, specName,
workTimes));
    }
    return doctors;
}

QList<Doctor> ServerDatabaseHelper::GetDoctors(User *user)
{
    int userId = GetUserId(user);
    if(userId == -1)
    {
        CreateDatabaseResponse("Користувач з логіном '" + user-
>GetLogin() + "' не існує. Не можливо загрузити лікаря.");
        return QList<Doctor>{};
    }else
        return
GetDoctors(ARG_DOCTORS_USER_ID+"='"+QString::number(userId)+"'", user);
}

void ServerDatabaseHelper::DeleteDoctor(Doctor* doctor)
{
//    if(doctor->GetUser()==nullptr) return;
//    int userId = GetUserId(doctor->GetUser());
//    if(userId == -1) return;

```

```

// DatabaseResponse* response = Delete(TABLE_DOCTORS_NAME,
ARG_DOCTORS_USER_ID+"='"+QString::number(userId)+"'");

// if(response->IsError())
// {
//     CreateDatabaseResponse("Не можливо видалити лікаря");
//     return;
// }else
// {
//     RefreshWorkTimes(userId, QList<WorkTime>{});
// }
}

int ServerDatabaseHelper::GetUserId(User* user)
{
    int userId;
    DatabaseResponse *response = Select(TABLE_USERS_NAME,
QList<QString>{ARG_USERS_ID, ARG_USERS_LOGIN},
ARG_USERS_LOGIN+"='"+user-
>GetLogin()+"'");
    if(!databaseQuery.first())
    {
        CreateDatabaseResponse("Користувач з логіном '" + user-
>GetLogin() + "' не існує. Не можливо отримати його id.");
        return -1;
    }else
        userId = databaseQuery.value(response-
>GetRecord().indexOf(ARG_USERS_ID)).toInt();
    return userId;
}

int ServerDatabaseHelper::GetDoctorId(Doctor* doctor)
{
// return GetDoctorId(GetUserId(doctor->GetUser()));
}

```



```

int ServerDatabaseHelper::GetDoctorId(int userId)
{
    int doctorId;
    DatabaseResponse* response = Select(TABLE_DOCTORS_NAME,
    QList<QString>{ARG_DOCTORS_ID},
    ARG_DOCTORS_USER_ID+"='"+QString::number(userId)+"'");
    if(!databaseQuery.first())
    {
        CreateDatabaseResponse("Лікаря з id '" + QString::number(userId)
+ "' не існує.");
        return -1;
    }else doctorId = databaseQuery.value(response-
>GetRecord().indexOf(ARG_DOCTORS_ID)).toInt();
    return doctorId;
}

```

```

int ServerDatabaseHelper::GetSpecialityId(QString spec)
{
    int specId;
    DatabaseResponse* response = Select(TABLE_SPECIALITIES_NAME,
    QList<QString>{ARG_SPECIALITIES_ID},
    ARG_SPECIALITIES_SPECIALITY+"='"+spec+"'");
    if(!databaseQuery.first())
    {
        CreateDatabaseResponse("Професії '" + spec + "' не існує.");
        return -1;
    }else specId = databaseQuery.value(response-
>GetRecord().indexOf(ARG_SPECIALITIES_ID)).toInt();
    return specId;
}
*/

```

Клієнт

main.cpp

```
#include "netservice.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    //Initializes the resources specified by the qrs.qrc file
    Q_INIT_RESOURCE(qrc);

    QCoreApplication::setOrganizationName(ORGANIZATION_NAME);
    QCoreApplication::setOrganizationDomain(ORGANIZATION_DOMAIN);
    QCoreApplication::setApplicationName(APPLICATION_NAME);

    QApplication a(argc, argv);

    auto core = new NetService;

    return a.exec();
}
```

authmenu.h

```
#ifndef AUTHMENU_H
#define AUTHMENU_H

#include <QMainWindow>
#include <QString>
#include <QMovie>

namespace Ui {
class AuthMenu;
}

class AuthMenu : public QMainWindow
{
    Q_OBJECT
```

```

public:

    explicit AuthMenu(QWidget *parent = nullptr);
    ~AuthMenu();

    void setStatus(const QString &status);
signals:
    void signIn(const QString &username, const QString &password);

private slots:

    void on_SignButton_clicked();
    void on_Login_returnPressed();

private:
    Ui::AuthMenu *ui;
};

#endif // AUTHMENU_H
authmenu.cpp
#include "authmenu.h"
#include "ui_authmenu.h"

#include <QCryptographicHash>

AuthMenu::AuthMenu(QWidget *parent)
    : QMainWindow(parent),
    ui(new Ui::AuthMenu)
{
    ui->setupUi(this);
    ui->Login->setFont(QFont("Gadugi", 12));
    ui->Password->setFont(QFont("Gadugi", 12));
    ui->Password->setEchoMode(QLineEdit::Password);
}

```

```
AuthMenu::~AuthMenu()
{
    delete ui;
}

void
AuthMenu::on_SignButton_clicked()
{
    auto login = ui->Login->text();
    auto password = QString(QCryptographicHash::hash((ui->Password-
>text()).toUtf8()),QCryptographicHash::Keccak_256).toHex());

    if (login.isEmpty())
    {
        this->setStatus("Login is empty.");
        return;
    }

    if (login.contains(QRegExp("[^a-zA-Z0-9]")))
    {
        this->setStatus("Login contains forbidden characters.");
        return;
    }

    if (password.isEmpty())
    {
        this->setStatus("Password is empty.");
        return;
    }

    // Disable all buttons while making request to server
    ui->SignButton->setEnabled(false);
    ui->Login->setEnabled(false);
    ui->Password->setEnabled(false);
```

```
// Show preloader gif until server response is caught
auto preloader = new QMovie(":/Image/46.gif");
preloader->start();
ui->Status->setMovie(preloader);

// Send signal with login and pass data
// to network handler
emit signIn(login, password);
}
```

```
void
AuthMenu::setStatus(const QString &status)
{
    ui->Status->setText(status);
    ui->SignIn->setEnabled(true);
    ui->Login->setEnabled(true);
    ui->Password->setEnabled(true);
}
```

```
void
AuthMenu::on_Login_returnPressed()
{
    on_SignButton_clicked();
}
```

chatwidget.h

```
#ifndef CHATWIDGET_H
#define CHATWIDGET_H

#include <QMainWindow>
#include <QFileDialog>
#include <QImage>
#include <QDebug>
#include <QSystemTrayIcon>
#include <QStyle>
```

```

#include <QTimer>

#include "../Data/userlist.h"
#include "../Data/userdata.h"
#include "../Data/message.h"
#include "../Data/filedata.h"
#include "styledialog.h"
#include "settings.h"
#include "database.h"
#include "popupmanager.h"
#include "mytextedit.h"

namespace Ui {
class ChatWidget;
}

class ChatWidget : public QMainWindow
{
    Q_OBJECT
public:

    explicit ChatWidget(DataBase *db, QWidget *parent = nullptr);
    ~ChatWidget();

    void addParticipant(const UserData &username);
    void removeParticipant(const QString &username);
    void addParticipants(const UserList &lst);
    void addMessage(const Message &msg);

private slots:
    void on_SendButton_clicked();
    void on_loadImageButton_clicked();
    void on_loadFileButton_clicked();
    void on_StyleChanged(QFont &font, QColor &color);
    void on_SettingsChanged();

```

```
void on_actionExit_triggered();
void iconActivated(QSystemTrayIcon::ActivationReason reason);
void checkWifi();
```

signals:

```
void messageSent(Message &msg);
void fileSent(FileData &fileData);
void close();
```

private:

```
void createActions();
void createTrayIcon();

UserData style;
Ui::ChatWidget *ui;
QPixmap bufferedImage;
StyleDialog dlgStyle;
Settings dlgSettinds;
QSystemTrayIcon *trayIcon;
QMenu *trayIconMenu;
PopUpManager *popUpMngr;
QTimer *wifiTimer;
DataBase *db;

QAction *minimizeAction;
QAction *maximizeAction;
QAction *restoreAction;
QAction *quitAction;
```

```
};
```

```
#endif // CHATWIDGET_H
```

```
chatwidget.cpp
```

```
#include "chatwidget.h"
```

```
#include "ui_chatwidget.h"
```

```
#include <QTextEdit>
```

```
#include <QNetworkInterface>
```

```
ChatWidget::ChatWidget(DataBase *db, QWidget *parent)
```

```
    : QMainWindow(parent),  
      ui(new Ui::ChatWidget),  
      db(db),  
      popUpMgr(new PopUpManager)
```

```
{
```

```
    ui->setupUi(this);  
    ui->LoadedImages->setVisible(false);  
    QSettings settings(ORGANIZATION_NAME, APPLICATION_NAME);
```

```
QApplication::setQuitOnLastWindowClosed(!settings.value(SETTINGS_CLOSEtoT  
RAY,true).toBool());
```

```
    this->createActions();  
    this->createTrayIcon();
```

```
    wifiTimer = new QTimer(this);  
    wifiTimer->setInterval(10000);  
    connect(wifiTimer, &QTimer::timeout, this, &ChatWidget::checkWifi);  
    this->checkWifi();
```

```
    connect(trayIcon, &QSystemTrayIcon::messageClicked, this,  
&QWidget::showNormal);
```

```
    connect(trayIcon, &QSystemTrayIcon::activated, this,  
&ChatWidget::iconActivated);
```

```
    trayIcon->show();
```

```
    connect(ui->actionStyle, &QAction::triggered,&dlgStyle,  
&QDialog::open);
```

```
    connect(ui->actionSettings, &QAction::triggered, &dlgSettinds,  
&QDialog::open);
```

```
    connect(&dlgSettinds, &Settings::settingsChaged, this,  
&ChatWidget::onSettingsChanged);
```



```

        connect(&dlgStyle, &StyleDialog::styleChanged, this,
&ChatWidget::onStyleChanged);
        connect(ui->MessageLine, &MyTextEdit::returnPressed, this,
&ChatWidget::on_SendButton_clicked);
        ui->MessageLine->setFocus();
    }

ChatWidget::~ChatWidget()
{
    trayIcon->hide();
    //    delete trayIcon;
    delete popUpMngr;
    delete ui;
}

void ChatWidget::createActions()
{
    minimizeAction = new QAction(tr("Mi&nimize"), this);
    connect(minimizeAction, &QAction::triggered, this, &QWidget::hide);

    maximizeAction = new QAction(tr("Ma&ximize"), this);
    connect(maximizeAction, &QAction::triggered, this,
&QWidget::showMaximized);

    restoreAction = new QAction(tr("&Restore"), this);
    connect(restoreAction, &QAction::triggered, this,
&QWidget::showNormal);

    quitAction = new QAction(tr("&Quit"), this);
    connect(quitAction, &QAction::triggered, qApp,
&QCoreApplication::quit);
}

void ChatWidget::createTrayIcon()
{
    trayIcon = new QSystemTrayIcon(this);

```

```

    trayIcon->setIcon(QApplication::style()-
>standardIcon(QStyle::SP_ComputerIcon));
    trayIcon->setToolTip("Telehealth_v1.0" "\n"
                        "Messenger for health workers");

    trayIconMenu = new QMenu(this);
    trayIconMenu->addAction(minimizeAction);
    trayIconMenu->addAction(maximizeAction);
    trayIconMenu->addAction(restoreAction);
    trayIconMenu->addSeparator();
    trayIconMenu->addAction(quitAction);
    trayIcon->setContextMenu(trayIconMenu);
}

void ChatWidget::iconActivated(QSystemTrayIcon::ActivationReason reason)
{
    switch (reason){
    case QSystemTrayIcon::Trigger:
        if(!this->isVisible()){
            this->showNormal();
        } else {
            this->hide();
        }
        break;
    default:
        break;
    }
}

void ChatWidget::addParticipant(const UserData &username)
{
    auto participant = new QListWidgetItem;
    participant->setFont(username.getFont());
    participant->setTextColor(username.getColor());
    participant->setText(username.getUsername());
    // Disable selecting participant in list widget

```

```
    participant->setFlags(participant->flags() & ~Qt::ItemIsSelectable);
    ui->Users->addItem(participant);
}
```

```
void ChatWidget::addMessage(const Message &msg)
```

```
{
    db->saveMsg(msg);

    auto item = new QListWidgetItem;
    item->setTextColor(msg.getTextColor());
    item->setFont(msg.getTextFont());
    item->setText(msg.getSender() + ": " + msg.getMessage());

    // Disable selecting message in dialog
    item->setFlags(item->flags() & ~Qt::ItemIsSelectable);
    ui->Dialog_2->addItem(item);
    ui->Dialog_2->scrollToBottom();

    if (!this->isActiveWindow())
        popUpMngr->newPopUp(msg.getMessage(), msg.getSender());
}
```

```
void ChatWidget::addParticipants(const UserList &lst)
```

```
{
    auto users = lst.getUsers();
    for (auto &user : users) this->addParticipant(user);
}
```

```
void ChatWidget::removeParticipant(const QString &username)
```

```
{
    auto itemToRemove = ui->Users->findItems(username,
Qt::MatchFixedString).at(0);
    delete itemToRemove;
}
```

```
void ChatWidget::on_SendButton_clicked()
{
    auto text = ui->MessageLine->toPlainText();
    if (text.isEmpty())
    {
        ui->MessageLine->setFocus();
        return;
    }

    /* todo: user can select font and color of his message */
    /* msg.setTextColors(...); msg.setFontColor(...); */

    //TODO вибір отримувача із списку клієнтів
    int receiverID = 5;
    QString receiver;

    Message msg("You", 0, receiver, receiverID,
QDateTime::currentDateTimeUtc(),
                text, QDateTime::currentDateTime(), style.getFont(),
style.getColor());

    // If user has loaded any image, add it to Message data
    if (!bufferedImage.isNull()) msg.setImage(bufferedImage);

    // Add message to dialog widget
    addMessage(msg);
    ui->MessageLine->setFocus();

    // Clear message line and loaded image label
    ui->MessageLine->clear();
    ui->LoadedImages->clear();
    ui->LoadedImages->setVisible(false);

    // Send message to server
    emit messageSent(msg);
}
```

```

        // Clear buffered image
        bufferedImage = QPixmap();
    }

void ChatWidget::on_loadImageButton_clicked()
{
    auto path = QFileDialog::getOpenFileName(this, tr("Open image"),
                                             "", tr("*.png *.jpg
*.jpeg"));
    // Get image name from string
    auto imageName = path.split("/").last();

    // Load image to UI
    QImage img;
    img.load(path);
    bufferedImage.fromImage(img);

    // Make UI visible
    ui->LoadedImages->setVisible(true);
    ui->LoadedImages->setText("Selected image: " + imageName);
}

void ChatWidget::on_loadFileButton_clicked()
{
    auto path = QFileDialog::getOpenFileName(this, tr("Open image"),
                                             "", tr("*.png *.jpg
*.jpeg"));
    QFile file(path);
    if (!file.open(QFile::ReadOnly))
    {
        qDebug() << "Could not open the file for reading";
        return;
    }
    QByteArray byteArray = file.readAll();

```

```

// Get file name from string
QString fileName = path.split("/").last();
QString fileType = fileName.split(".").last();
if (fileType.isEmpty())
    fileType = ".*.*";

// // Load image to UI
// QImage img;
// img.load(path);
// bufferedImage.fromImage(img);

FileData fileData("", "", fileName, fileType, byteArray,
bufferedImage);
// Send message to server
emit fileSent(fileData);
this->addMessage(Message("You", "File " + fileName + " is sended!"));
}

void ChatWidget::onStyleChanged(QFont &font, QColor &color)
{
    style.setFont(font);
    style.setColor(color);
}

void ChatWidget::onSettingsChanged()
{
    QSettings settings(ORGANIZATION_NAME, APPLICATION_NAME);

QApplication::setQuitOnLastWindowClosed(!settings.value(SETTINGS_CLOSEtoT
RAY,true).toBool());
}

void ChatWidget::on_actionExit_triggered()
{
    emit this->close();
}

```

```

void ChatWidget::checkWifi()
{
    wifiTimer->start();
    QList<QNetworkInterface> ifaces = QNetworkInterface::allInterfaces();

    for (auto &&iface : ifaces) {
        if ( iface.type() == QNetworkInterface::Wifi
&& !iface.flags().testFlag(QNetworkInterface::IsLoopBack))
        {
            if (iface.flags().testFlag(QNetworkInterface::IsUp))
            {
                ui->lblWifi->setPixmap(QPixmap(":/Image/wifiOn.png"));
                QString str("Wi-Fi network adapter is on and
connected!\n");
                str.append("Network interface name:
").append(iface.name())
                    .append("\nIp address:
").append(iface.addressEntries().at(1).ip().toString())
                    .append("\nNetwork mask:
").append(iface.addressEntries().at(1).netmask().toString())
                    .append("\nMac address:
").append(iface.hardwareAddress());
                ui->lblWifi->setToolTip(str);

            }
            else
            {
                ui->lblWifi->setPixmap(QPixmap(":/Image/wifiOff.png"));
                ui->lblWifi->setToolTip(QString ("Wi-Fi network adapter
is disconnected\n"));
            }
            return;
        }
    }
}

```

```
database.h
#ifndef DATABASE_H
#define DATABASE_H

#include <QObject>
#include <QSql>
#include <QSqlQuery>
#include <QSqlError>
#include <QSqlDatabase>
#include <QFile>
#include <QDate>
#include <QDebug>

#include "../Data/message.h"

/* Директиви імен таблиць, полів і бази даних */
#define DATABASE_HOSTNAME      "ExampleDataBase"
#define DATABASE_NAME         "LocalDataBase.db"

#define TABLE                 "Clients"
#define TABLE_DATE           "Date"

#define TABLE_MSG             "messages"
#define TABLE_MSG_ID         "id"
#define TABLE_MSG_TIMESTAMP  "time_stamp"
#define TABLE_MSG_FROM       "from_user_id"
#define TABLE_MSG_TO         "to_user_id"
#define TABLE_MSG_TEXT       "message"
#define TABLE_MSG_CREATED    "created_at"

// #define TABLE_USERS        "users"
// #define TABLE_USERS_ID     "user_id"
// #define TABLE_USERS_NAME   "user_name"
// #define TABLE_USERS_DESC   "user_desc"
// #define TABLE_USERS_registration_date  last_appearance online
```



```

class DataBase : public QObject
{
    Q_OBJECT
public:
    explicit DataBase(QObject *parent = nullptr);
    ~DataBase();

    /* Методи для роботи з класом
     * Підключення к базі даних і вставка записей в таблицю
     * */
    void connectToDataBase(QString userName);
    bool insertIntoTable(const QVariantList &data);
    bool saveMsg(const Message &msg);
    QString getLastMsgTimeStamp();

signals:

private:
    /* Внутрішні методи для роботи з БД
     * */
    bool openDataBase();
    bool restoreDataBase();
    void closeDataBase();
    bool createTable();

private:
    // Об'єкт бази даних, з яким буде виконуватись робота
    QSqlDatabase db;
    QString dbName;
    QString dbLocation;
};

#endif // DATABASE_H
database.cpp
#include "database.h"

```

```

DataBase::DataBase(QObject *parent) : QObject(parent)
{
}
DataBase::~DataBase()
{
}

/* Метод для підключення до БД
 * */
void DataBase::connectToDataBase(QString userName)
{
    dbName = userName;
    /* Перед підключенням до БД виконуємо перевірку на її існування.
     * В залежності від результату виконуємо відкриття БД або її
відновлення
     * */
    if(!QFile(dbLocation + dbName + ".sqlite").exists()){
        this->restoreDataBase();
    } else {
        this->openDataBase();
    }
}

QString DataBase::getLastMsgTimeStamp()
{
    QString str;
    return str;
}

/* Метод відновлення бази даних
 * */
bool DataBase::restoreDataBase()
{
    if(this->openDataBase()){
        if(!this->createTable()){
            return false;
        }
    }
}

```

```

        else {
            qDebug() << "database restored";
            return true;
        }
    } else {
        qDebug() << "Failed to restore database";
        return false;
    }
}

/* Метод відкриття бази даних
 * */
bool DataBase::openDataBase()
{
    /* База даних відкривається по заданому шляху
    * та імені БД, якщо вона існує
    * */
    db = QSqlDatabase::addDatabase("QSQLITE");
    db.setHostName(dbName);
    db.setDatabaseName(dbLocation + dbName + ".sqlite");
    if(db.open()){
        return true;
    } else {
        return false;
    }
}

/* Метод закриття бази даних
 * */
void DataBase::closeDataBase()
{
    db.close();
}

/* Метод для створення таблиць в базі даних
 * */

```

```

bool DataBase::createTable()
{
    /* В даному випадку викорситовується формування сирого SQL-запиту
    * з наступним його виконанням
    * */
    QSqlQuery query;
    if(!query.exec( "CREATE TABLE " TABLE_MSG " ("
                    "id INTEGER PRIMARY KEY ON CONFLICT IGNORE
AUTOINCREMENT, "
                    TABLE_MSG_TIMESTAMP " TIME                NOT NULL, "
                    TABLE_MSG_FROM      " INTEGER              NOT NULL, "
                    TABLE_MSG_TO        " INTEGER              NOT NULL, "
                    TABLE_MSG_TEXT      " VARCHAR(500)         NOT NULL, "
                    TABLE_MSG_CREATED   " DATETIME              NOT NULL "
                    " )"
                    ))
    {
        qDebug() << "DataBase: error of create " << TABLE;
        qDebug() << query.lastError().text();
        qDebug() << query.lastQuery();
        return false;
    } else {
        return true;
    }
}

/* Метод для вставки запису в базу даних
* */
bool DataBase::insertIntoTable(const QVariantList &data)
{
    /* Запит SQL формуємо із QVariantList,
    * в який передаються данні для вставки в таблицю.
    * */
    QSqlQuery query;
    /* Спочатку SQL запит формуємо з ключамизапрос формируется с ключами,

```

```
* які потім зв'язуємо методом bindValue
* для підстановки даних із QVariantList
*
```

```
query.prepare("INSERT INTO " TABLE " ( " TABLE_DATE ", "
              //          TABLE_TIME ", "
              //          TABLE_RANDOM ", "
              //          TABLE_MESSAGE " ) "
              "VALUES (:Date, :Time, :Random, :Message )");
query.bindValue(":Date",      data[0].toDate());
query.bindValue(":Time",     data[1].toTime());
query.bindValue(":Random",   data[2].toInt());
query.bindValue(":Message",  data[3].toString());
// Після чого виконуємо запит методом exec()
if(!query.exec()){
    qDebug() << "error insert into " << TABLE;
    qDebug() << query.lastError().text();
    return false;
} else {
}
*/

return true;
```

```
}
```

```
bool DataBase::saveMsg(const Message &msg)
{
// QSqlQuery query;
// query.prepare("INSERT INTO " TABLE_MSG " ( " TABLE_DATE ", "
//          //          TABLE_TIME ", "
//          //          TABLE_RANDOM ", "
//          //          TABLE_MESSAGE " ) "
//          "VALUES (:Date, :Time, :Random, :Message )");
// query.bindValue(":Date",      data[0].toDate());
// query.bindValue(":Time",     data[1].toTime());
// query.bindValue(":Random",   data[2].toInt());
// query.bindValue(":Message",  data[3].toString());
```

```
//    if(!query.exec()){
//        qDebug() << "error insert into " << TABLE;
//        qDebug() << query.lastError().text();
//        return false;
//    } else {
//        return true;
//    }
}
```

mytextedit.h

```
#ifndef MYTEXTEDIT_H
#define MYTEXTEDIT_H

#include <QObject>
#include <QTextEdit>
#include <QKeyEvent>

class MyTextEdit : public QTextEdit
{
    Q_OBJECT
public:
    explicit MyTextEdit(QWidget *parent = nullptr);
    virtual ~MyTextEdit();

signals:
    void returnPressed();

protected:
    void keyPressEvent(QKeyEvent *event);
};

#endif // MYTEXTEDIT_H

mytextedit.cpp
#include "mytextedit.h"
```

```

MyTextEdit::MyTextEdit(QWidget *parent):
    QTextEdit(parent)
{
}
MyTextEdit::~MyTextEdit()
{
}
#include <QDebug>
void MyTextEdit::keyPressEvent(QKeyEvent *event)
{
    if (event->key() == Qt::Key_Return &&
        !(event->modifiers().testFlag(Qt::ControlModifier) || event-
>modifiers().testFlag(Qt::ShiftModifier)))
    {
        emit returnPressed();
    }
    else
    {
        QTextEdit::keyPressEvent(event);
    }
}
netservice.h
#ifndef NETSERVICE_H
#define NETSERVICE_H

#include "authmenu.h"
#include "chatwidget.h"
#include "settings.h"
#include "database.h"

#include "../Data/authanswer.h"
#include "../Data/userdata.h"
#include "../Data/userlist.h"
#include "../Data/filedata.h"
#include "../3rdparty/simplecrypt.h"

```

```

#include <QApplication>
#include <QTcpSocket>
#include <QHostAddress>
#include <QTimer>
#include <QCloseEvent>
#include <QSystemTrayIcon>
#include <QAction>
#include <QSettings>
#include <QMessageBox>
#include <QBuffer>

class NetService : public QObject
{
    Q_OBJECT
public:

    explicit NetService(QObject *parent = nullptr);
    virtual ~NetService();

    void write(QTcpSocket *socket, const BaseData &data);
private slots:

    // Slot handles clicking "Sing in" button in auth menu
    void onSignIn(const QString &username, const QString &password);

    void onSocketConnected();
    void onSocketDisconnected();
    void onConnectionTimeOut();
    void onSocketReadyRead();
    void onMessageSent(Message &msg);
    void onFileSent(FileData &fileData);
    void onClose();

private:

    void connectSocketSignals();

```



```
// Incoming data handling methods
void readUserList(QDataStream &readStream);
void readAuthAnswer(QDataStream &readStream);
void readNewConnection(QDataStream &readStream);
void readDisconnection(QDataStream &readStream);
void readMessage(QDataStream &readStream);
void readFile(QDataStream &readStream);
```

```
private:
```

```
AuthMenu *autMenu;
ChatWidget *chat;
QTcpSocket *socket;
QTimer *ctimer; // connection timer
SimpleCrypt *crypto;
UserData identifier;
DataBase *db;
```

```
};
```

```
#endif // NETSERVICE_H
```

```
netservice.cpp
```

```
#include "netservice.h"
```

```
NetService::NetService(QObject *parent)
```

```
: QObject(parent),
  autMenu(new AuthMenu()),
  socket(new QTcpSocket(this)),
  ctimer(new QTimer(this)),
  crypto(new SimpleCrypt(Q_UINT64_C(0x0c2ad5a5acb9f007)))
```

```
{
```

```
autMenu->show();
connectSocketSignals();
connect(autMenu, &AuthMenu::signIn, this, &NetService::onSignIn);
connect(ctimer, SIGNAL(timeout( )),
        this, SLOT(onConnectionTimeOut()));
```

```
        crypto->setCompressionMode(SimpleCrypt::CompressionAlways); //always
compress the data
        crypto->setIntegrityProtectionMode(SimpleCrypt::ProtectionHash);
//properly protect the integrity of the data
    }
```

```
NetService::~~NetService()
```

```
{
    delete autMenu;
    if (chat)
        delete chat;
    delete crypto;
}
```

```
void NetService::connectSocketSignals()
```

```
{
    connect(socket, &QTcpSocket::connected, this,
&NetService::onSocketConnected);
    connect(socket, &QTcpSocket::disconnected, this,
&NetService::onSocketDisconnected);
    connect(socket, &QTcpSocket::readyRead, this,
&NetService::onSocketReadyRead);
}
```

```
void NetService::onSignIn(const QString &username, const QString
&password)
```

```
{
    identifier.setUsername(username);
    identifier.setPassword(password);
    //TODO configuration IP
    // Connect to server and write auth data on success
    // see onSocketConnected() slot
    socket->connectToHost(QHostAddress::LocalHost, 8001);

    // Create 10 seconds timer for waiting server response
    ctimer->start(10000);
}
```

```
}
```

```
void NetService::onSocketConnected()
```

```
{
```

```
    ctimer->stop();
```

```
    identifier.setType(DataType::AuthRequest);
```

```
    write(socket, identifier);
```

```
}
```

```
void NetService::onSocketDisconnected()
```

```
{
```

```
    socket->abort();
```

```
}
```

```
void NetService::onConnectionTimeOut()
```

```
{
```

```
    socket->abort();
```

```
    autMenu->setStatus("Connection timeout.");
```

```
}
```

```
void NetService::readAuthAnswer(QDataStream &readStream)
```

```
{
```

```
    AuthAnswer ans;
```

```
    readStream >> ans;
```

```
    if (!ans.isSigned())
```

```
    {
```

```
        autMenu->setStatus("Authentication error.");
```

```
        socket->abort();
```

```
        return;
```

```
    }
```

```
    crypto->setKey(ans.getKey());
```

```
    db = new DataBase(this);
```

```
    db->connectToDataBase(identifier.getUsername());
```

```

    chat = new ChatWidget(db);

    connect(chat, &ChatWidget::messageSent, this,
&NetService::onMessageSent);
    connect(chat, &ChatWidget::fileSent, this, &NetService::onFileSent);
    connect(chat, &ChatWidget::close, this, &NetService::onClose);
    chat->setWindowTitle(chat->windowTitle() + " User: " +
identifier.getUsername());
    chat->show();
    autMenu->close();
}
void NetService::readUserList(QDataStream &readStream)
{
    UserList lst;
    readStream >> lst;
    chat->addParticipants(lst);
}

void NetService::readNewConnection(QDataStream &readStream)
{
    UserData user;
    readStream >> user;
    chat->addParticipant(user);
}

void NetService::readDisconnection(QDataStream &readStream)
{
    UserData user;
    readStream >> user;
    chat->removeParticipant(user.getUsername());
}

void NetService::readMessage(QDataStream &readStream)
{
    Message msg;
    readStream >> msg;
}

```

```
    db->saveMsg(msg);
    chat->addMessage(msg);
}
```

```
void NetService::readFile(QDataStream &readStream)
```

```
{
    FileData fileData;
    readStream >> fileData;
    Message msg("system","You have a file!");

    chat->addMessage(msg);
    auto path = QFileDialog::getSaveFileName(nullptr, tr("Save image"),
                                             fileData.getFileName(),
                                             "*. " +
fileData.getFileType());
    QFile loadedFile(path);
    if (loadedFile.open(QIODevice::WriteOnly))
    {
        loadedFile.write(fileData.getData());
    }
    loadedFile.close();
}
```

```
void NetService::write(QTcpSocket *socket, const BaseData &data)
```

```
{
    QByteArray datagram;
    QDataStream writeStream(&datagram, QIODevice::WriteOnly);
    writeStream.setVersion(QDataStream::Qt_5_12);
    writeStream << quint32(0) << qint8(data.type());

    QBuffer cryptoBuffer;
    cryptoBuffer.open(QIODevice::WriteOnly);
    QDataStream cryptoStream(&cryptoBuffer);
    //stream the data into our buffer
    cryptoStream.setVersion(QDataStream::Qt_5_12);
    cryptoStream << data;
}
```

```

    QByteArray cryptoData = crypto-
>encryptToByteArray(cryptoBuffer.data());
    // if no error occurred
    if (crypto->lastError() == SimpleCrypt::ErrorNoError) {
        writeStream << cryptoData;
        qDebug() << datagram.size();
        qDebug() << quint32(quint32(datagram.size()) - sizeof(quint32));
        writeStream.device()->seek(0);
        writeStream << quint32(quint32(datagram.size()) -
sizeof(quint32)); // writing data size
        socket->write(datagram);
        socket->waitForBytesWritten();
    }
    cryptoBuffer.close();
}

```

```

void NetService::onSocketReadyRead()
{
    auto socket = static_cast<QTcpSocket *>(sender());
    QDataStream readStream(socket);
    readStream.setVersion(QDataStream::Qt_5_12);
    quint32 sizeRead;
    qint8 type;

    readStream >> sizeRead;
    /* waiting till we get full data */
    while(socket->bytesAvailable() < sizeRead){
        if (!socket->waitForReadyRead())
        {
            qDebug() << "waitForReadyRead() timed out";
            return;
        }
    }
    readStream >> type;

    QByteArray cypherData;

```

```

readStream >> cypherData;

QByteArray plainData = crypto->decryptToByteArray(cypherData);
if (crypto->lastError() != SimpleCrypt::ErrorNoError) {
    qDebug() << "NetService::on_Socket_Ready_Read() ->
crypto.lastError() = " << crypto->lastError();
    return;
}

QBuffer buffer(&plainData);
buffer.open(QIODevice::ReadOnly);
QDataStream stream(&buffer);
stream.setVersion(QDataStream::Qt_5_12);

if (type == DataType::AuthResponse)
    readAuthAnswer(stream);
if (type == DataType::UserListResponse)
    readUserList(stream);
if (type == DataType::NewConnection)
    readNewConnection(stream);
if (type == DataType::Disconnection)
    readDisconnection(stream);
if (type == DataType::Msg)
    readMessage(stream);
if (type == DataType::File)
    readFile(stream);
}

void NetService::onMessageSent(Message &msg)
{
    msg.setSender(identifier.getUsername());
    write(socket, msg);
}

void NetService::onFileSent(FileData &fileData)

```

```
{
    fileData.setSender(identifier.getUsername());
    write(socket, fileData);
}
```

```
void NetService::onClose()
{
    socket->abort();
    QApplication::quit();
}
```

popup.h

```
#ifndef POPUP_H
```

```
#define POPUP_H
```

```
#include <QWidget>
```

```
#include <QLabel>
```

```
#include <QGridLayout>
```

```
#include <QPropertyAnimation>
```

```
#include <QTimer>
```

```
QT_WARNING_DISABLE_DEPRECATED
```

```
class PopUp : public QWidget
```

```
{
```

```
    Q_OBJECT
```

```
    // Властивість напівпрозорості
```

```
    Q_PROPERTY(double popupOpacity READ getPopupOpacity WRITE
setPopupOpacity)
```

```
    void setPopupOpacity(double opacity);
```

```
    double getPopupOpacity() const;
```

```
public:
```

```
    explicit PopUp(QWidget *parent = nullptr);
```

```
signals:
```



```

void hided();

protected:
    void paintEvent(QPaintEvent *event);    // Фон буде промальований
через метод перемалювання

public slots:
    void setPopupText(const QString &text, const QString &sender); //
Встановлюємо текст в повідомлення
    void show(int lastPopupHeight);        /* Власним меод
відображення віджету
                                           * потрібно для
попередньої настройки анімації
                                           * */

private slots:
    void hideAnimation();                  // Слот для запуску анімації
приховування віджета
    void hide();                           /* По закінченню анімації, в
даному слоті виконується перевірка
                                           * чи видно віджет, чи
його потрібно приховати
                                           * */

private:
    QLabel labelSender;                    // Label з відправником
    QLabel labelMsg;                       // Label з повідомленням
    QGridLayout layout;                    // Розміщення лайблів
    QPropertyAnimation animation;         // Властивість анімації для
спливаючого повідомлення
    double popupOpacity;                   // Властивість напівпрозорості віджету
    QTimer *timer;                         // Таймер, по зкінченню якого віджет буде
прихований
};

#endif // POPUP_H

```



```

        "margin-bottom: 2px;"
        "margin-left: 10px;"
        "margin-right: 10px;"
        "font: 12pt \"Calibri\"; }");

// Налаштування тексту повідомлення
labelMsg.setAlignment(Qt::AlignHCenter | Qt::AlignVCenter); //
Вирівнюємо по центру
// Задаємо форматування тексту
labelMsg.setStyleSheet("QLabel { color : white; "
        "margin-top: 2px;"
        "margin-bottom: 6px;"
        "margin-left: 10px;"
        "margin-right: 10px;"
        "font: 12pt \"Calibri\"; }");

// Виконуємо встановлення тексту у лейаут, ...
layout.addWidget(&labelSender, 0, 0);
layout.addWidget(&labelMsg, 1, 0);
setLayout(&layout); // який поміщаємо у віджет

// По сигналу таймера повідомлення буде приховане
timer = new QTimer();
connect(timer, &QTimer::timeout, this, &PopUp::hideAnimation);
}

void PopUp::paintEvent(QPaintEvent *event)
{
    Q_UNUSED(event)

    // Задаємо фон повідомлення
    QPainter painter(this);
    painter.setRenderHint(QPainter::Antialiasing); // Включаємо
згладжування

    // Визначаємо координати фону rect()

```

```

    QRect roundedRect;
    roundedRect.setX(rect().x() + 5);
    roundedRect.setY(rect().y() + 5);
    roundedRect.setWidth(rect().width() - 10);
    roundedRect.setHeight(rect().height() - 10);
    painter.setBrush(QBrush(QColor(0,0,0,180)));
    painter.setPen(Qt::NoPen);
    painter.drawRoundedRect(roundedRect, 10, 10);
}

```

```

void Popup::setPopupText(const QString &text, const QString &sender)
{
    labelSender.setText(sender + ":");
    labelMsg.setText(text);    // Задаємо текст повідомлення
    adjustSize();             // з перерахунком розміру віджета
}

```

```

void Popup::show(int lastPopupHeight)
{
    setWindowOpacity(0.0);

    animation.setDuration(150);
    animation.setStartValue(0.0);
    animation.setEndValue(1.0);

    int y;
    if (lastPopupHeight > 0)
        y = lastPopupHeight - 10 - height() + QApplication::desktop() ->
availableGeometry().y();
    else
        y = QApplication::desktop()->availableGeometry().height() - 36 -
height() + QApplication::desktop() -> availableGeometry().y();
    int x = QApplication::desktop()->availableGeometry().width() - 36 -
width() + QApplication::desktop() -> availableGeometry().x();

    setGeometry(x, y, width(), height());
}

```

```

    QWidget::show();

    animation.start();
    timer->start(3000);
}

void PopUp::hideAnimation()
{
    timer->stop();
    animation.setDuration(1000);
    animation.setStartValue(1.0);
    animation.setEndValue(0.0);
    animation.start();
}

void PopUp::hide()
{
    if(getPopupOpacity() == 0.0){
        QWidget::hide();
        emit hided();
    }
}

void PopUp::setPopupOpacity(double opacity)
{
    popupOpacity = opacity;

    setWindowOpacity(opacity);
}

double PopUp::getPopupOpacity() const
{
    return popupOpacity;
}
popupmanager.h
#ifndef POPUPMANAGER_H

```

```

#define POPUPMANAGER_H

#include <QObject>
#include <QLinkedList>

#include "popup.h"

class PopUpManager : public QObject
{
    Q_OBJECT
public:
    PopUpManager();
    virtual ~PopUpManager();

    void newPopUp(const QString &Message, const QString &sender);

private:
    QList<PopUp *> popUpsList;

private slots:
    void onMessegeHidded();
};

#endif // POPUPMANAGER_H

popupmanager.cpp
#include "popupmanager.h"

#include <QApplication>
#include <QDesktopWidget>

PopUpManager::PopUpManager()
{
}

PopUpManager::~~PopUpManager()
{
}

```

```
}
```

```
void PopUpManager::newPopUp(const QString &message, const QString  
&sender)  
{  
    int lastPopupHeight = 0;  
    if (!popUpsList.isEmpty())  
        if (popUpsList.last()->geometry().y() > QApplication::desktop()-  
>availableGeometry().height()/3)  
            lastPopupHeight = popUpsList.last()->geometry().y();  
    popUpsList.append(new PopUp);  
    connect(popUpsList.last(), &PopUp::hided, this,  
&PopUpManager::onMessegeHidded);  
    popUpsList.last()->setPopupText(message, sender);  
    popUpsList.last()->show(lastPopupHeight);  
}
```

```
void PopUpManager::onMessegeHidded()  
{  
    if (!popUpsList.isEmpty())  
        delete popUpsList.takeFirst();  
}
```

```
settings.h
```

```
#ifndef SETTINGS_H  
#define SETTINGS_H
```

```
#include <QDialog>  
#include <QSettings>  
#include <QMessageBox>
```

```
/* Defining */
```

```
#define ORGANIZATION_NAME "MySoft"  
#define ORGANIZATION_DOMAIN "mysoft.com"  
#define APPLICATION_NAME "Telehealth_v1.0"
```

```

#define SETTINGS_CLOSEtoTRAY "settings/tray"

namespace Ui {
class Settings;
}

class Settings : public QDialog
{
    Q_OBJECT

public:
    explicit Settings(QWidget *parent = nullptr);
    ~Settings();

signals:
    void settingsChaged();

private slots:
    void on_buttonBox_accepted();

    void on_buttonBox_rejected();

private:
    Ui::Settings *ui;
    void setControlsAccordingSettings();
};

#endif // SETTINGS_H

settings.cpp
#include "settings.h"
#include "ui_settings.h"
#include <QDebug>

Settings::Settings(QWidget *parent) :
    QDialog(parent),

```



```

    ui(new Ui::Settings)
{
    ui->setupUi(this);
    setControlsAccordingSettings();
}

Settings::~Settings()
{
    delete ui;
}

void Settings::setControlsAccordingSettings()
{
    QSettings settings(ORGANIZATION_NAME, APPLICATION_NAME);

    ui->checkBoxTray->setChecked(settings.value(SETTINGS_CLOSEtoTRAY,
true).toBool());

}

void Settings::on_buttonBox_accepted()
{
    QSettings settings(ORGANIZATION_NAME, APPLICATION_NAME);

    /* Information on the inclusion of the minimization function in the
tray
    * is saved in accordance with the state of the checkbox.
    * */
    settings.setValue(SETTINGS_CLOSEtoTRAY, ui->checkBoxTray-
>isChecked());

    //Saving settings
    settings.sync();
    QMessageBox::information(this,
                                tr("Save settings"),
                                tr("Settings saved successfully!"));
}

```

```

        emit settingsChaged();
        this->close();
    }

void Settings::on_buttonBox_rejected()
{
    setControlsAccordingSettings();
    this->close();
}

```

styledialog.h

```

#ifndef STYLEDIALOG_H
#define STYLEDIALOG_H

```

```

#include <QDialog>
#include <QColorDialog>
#include <QFontDialog>
namespace Ui {
class StyleDialog;
}

```

```

class StyleDialog : public QDialog

```

```

{
    Q_OBJECT

```

```

public:

```

```

    explicit StyleDialog(QWidget *parent = nullptr);
    ~StyleDialog();

```

```

signals:

```

```

    void styleChanged(QFont&, QColor&);

```

```

private slots:

```

```

    void on_selectFont_clicked();
    void on_selectColor_clicked();

```

```
void on_pushButton_clicked();
```

```
private:
```

```
Ui::StyleDialog *ui;
```

```
QFont userFont;
```

```
QColor userColor;
```

```
};
```

```
#endif // STYLEDIALOG_H
```

```
styledialog.cpp
```

```
#include"styledialog.h"
```

```
#include"ui_styledialog.h"
```

```
StyleDialog::StyleDialog(QWidget*parent)
```

```
:QDialog(parent),
```

```
ui(newUi::StyleDialog),
```

```
userFont(QFont("Gadugi",12)),
```

```
userColor(QColor(Qt::red))
```

```
{
```

```
ui->setupUi(this);
```

```
this->setFixedSize(this->size());
```

```
QStringpattern=tr("<fontcolor='%1'>%2</font>");
```

```
ui->Pattern->setText(pattern.arg(userColor.name()),"Pattern");
```

```
ui->Pattern->setFont(userFont);
```

```
ui->currentColor->setText("Currentcolor:"+userColor.name());
```

```
ui->currentFont->setText("Currentfont:"+userFont.family());
```

```
}
```

```
StyleDialog::~StyleDialog()
```

```
{
```

```
deleteui;
```

```
}
```

```
void
```

```
StyleDialog::on_selectFont_clicked()
```

```
{
```

```
bool ok;
```

```
userFont=QFontDialog::getFont(&ok,this);
```

```
ui->Pattern->setFont(userFont);
```

```
ui->currentFont->setText("Currentfont:"+userFont.family());
```

```
}
```

```
void
```

```
StyleDialog::on_selectColor_clicked()
```

```
{
```

```
userColor=QColorDialog::getColor(Qt::red,this);
```

```
QString pattern=tr("<fontcolor='%1'>%2</font>");
```

```
ui->Pattern->setText(pattern.arg(userColor.name(),"Pattern"));
```

```
ui->currentColor->setText("Currentcolor:"+userColor.name());
```

```
}
```

```
void
```

```
StyleDialog::on_pushButton_clicked()
```

```
{
```

```
emit styleChanged(userFont,userColor);
```

```
this->close();
```

```
}
```

