



БАҒДАРЛАМАЛЫҚ ИНЖЕНЕРИЯ
ПРОГРАММНАЯ ИНЖЕНЕРИЯ
SOFTWARE ENGINEERING

DOI 10.51885/1561-4212_2023_4_143

MFTAA 50.41.25

Л.В. Долматова¹, Г.М. Борамбаева², Г.А. Ким³, Е.В. Ушакова⁴

М. Қозыбаев атындағы Солтүстік Қазақстан университеті, Петропавл қ., Қазақстан

¹E-mail: ldolmatova@ku.edu.kz*

²E-mail: gmborambaeva@ku.edu.kz

³E-mail: gakim@ku.edu.kz

⁴E-mail: eushakova@ku.edu.kz

ПИРИНГТІК ЖЕЛІЛЕР ТЕХНОЛОГИЯСЫ АРҚЫЛЫ БЕЙНЕБАЙЛАНЫСТЫ ҰЙЫМДАСТЫРУ БОЙЫНША ҚОСЫМШАНЫ ӨЗІРЛЕУ

РАЗРАБОТКА ПРИЛОЖЕНИЯ ПО ОРГАНИЗАЦИИ ВИДЕОСВЯЗИ ПОСРЕДСТВОМ ТЕХНОЛОГИИ ПИРИНГОВЫХ СЕТЕЙ

DEVELOPMENT OF AN APPLICATION FOR ORGANIZING VIDEO COMMUNICATION THROUGH PEER-TO-PEER NETWORKING TECHNOLOGY

Аңдатпа. Бұл жұмыс пәндік саланың ақпараттық жүйелерін сипаттауға және модельдеуге, сондай-ақ жобалық іске асыру үшін қолданылған технологияларға арналған. Мақалада жобаның тиімділігі тұрғысынан әдістер қарастырылады, жалпыдан егжей-тегжейге ауысады, қажет болған жағдайда функцияларды бөледі және ыдыратады. Пирингтік желілерге ерекше назар аударылады, өйткені олардың негізгі мақсаты – ресурстарды бөлу және физикалық жағының ерекшелігі талап ететін белгілі бір тапсырмаларды орындау үшін құрылғылар арасында бірлескен жұмысты қамтамасыз ету. Жұмыстың нәтижесі жобаның түбірлік каталогында орналасқан барлық файлдар жиынтығы бар қосымшаның физикалық құрылымы болды.

Түйін сөздер: Бейне байланыс, пирингтік желі технологиялары, HTML, CSS, JavaScript, Node.js, AJAX, WebSocket, WebRTC, MongoDB, Mongoose, AJAX, JWT.

Аннотация. Данная работа посвящена описанию и моделированию информационных систем предметной области, а также технологиям, которые были применены для проектной реализации. В статье рассмотрены методы с точки зрения результативности проекта, двигаясь от общего к деталям, по мере необходимости разделяя и декомпозируя функции. Особое внимание уделено пиринговым сетям, так как основная их цель – разделение ресурсов и обеспечение совместной работы между устройствами для выполнения определенных задач, что и требует специфика физической стороны. Результатом работы явилась физическая структура приложения с совокупностью всех файлов, которые находятся в корневой директории проекта.

Ключевые слова: Видеосвязь, технологии пиринговых сетей, HTML, CSS, JavaScript, Node.js, AJAX, WebSocket, WebRTC, MongoDB, Mongoose, AJAX, JWT.

Abstract. This work is devoted to the description and modeling of information systems of the subject area, as well as technologies that were used for project implementation. The article discusses methods from the point of view of project effectiveness, moving from the general to the details, separating and decomposing functions as necessary. Particular attention is paid to peer-to-peer networks, since their main purpose is to share resources and ensure collaboration between devices to perform certain tasks, which is required by the specifics of the physical side. The result of the work was the physical structure of the application with the totality of all the files that are located in the root directory of the project.

Keywords: Video communication, peer-to-peer network technologies, HTML, CSS, JavaScript, Node.js, AJAX, WebSocket, WebRTC, MongoDB, Mongoose, AJAX, JWT.

Kipicne. Интернеттің пайда болуы мен дамуы адамдар арасындағы байланыс құралдарының өзгеруіне әсер етті. Егер бұрын адамдар телефон соғып, сұхбаттасушының дауысын ғана ести алатын болса, қазіргі мүмкіндіктер сізге көруге де мүмкіндік береді. Бұл үшін әркім қолдана алатын көптеген түрлі қызметтер бар, бірақ жеке компанияларда қолданылатын жергілікті қызметтер де бар [1].

Көптеген ірі компаниялар өз пайдаланушыларымен осындай байланыс орнатуға мүмкіндік береді. Оларға Skype, Zoom сияқты бағдарламалық өнімдер, сондай-ақ WhatsApp, Viber, Вконтакте және т.б. сияқты әртүрлі әлеуметтік желілердегі бейнеконференциялар кіреді.

Әр компания бейнеконференцияны ұйымдастыру және олардың қауіпсіздігін қамтамасыз ету үшін әртүрлі технологияларды қолданады. Мысалы, аудио және бейне деректерін ағынмен жіберуге және браузерді бейнеконференция терминалы ретінде пайдалануға мүмкіндік беретін WebRTC технологиясы бар.

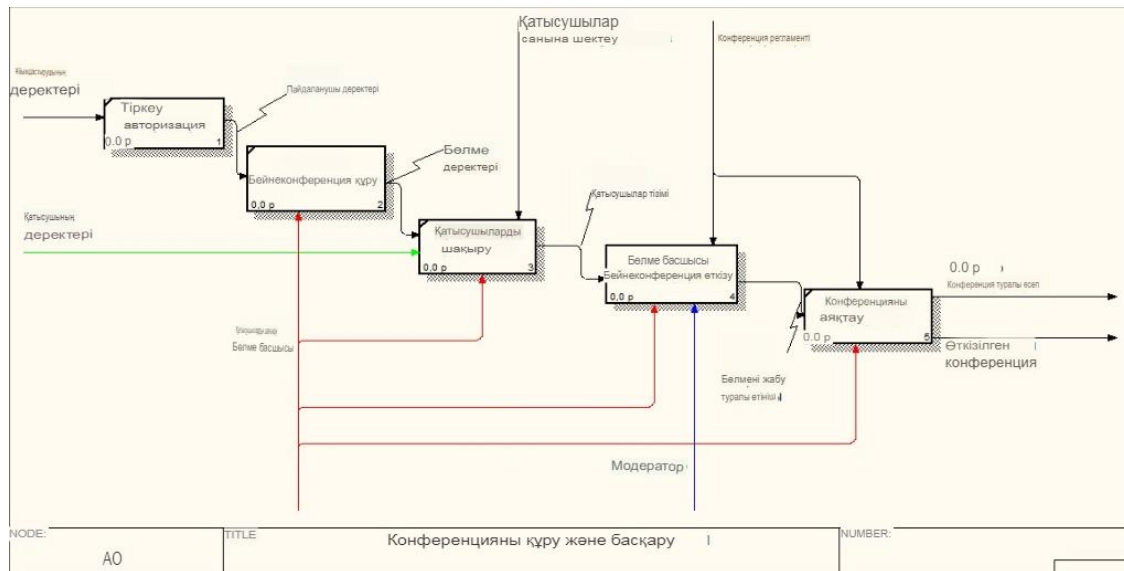
Бірақ ең қызықты технология - P2P хаттамасын немесе әртүрлі мүмкіндіктері мен ерекшеліктері бар пирингтік желілерді пайдалану болып табылады.

Бұл мақаланың *өзектілігі* пирингтік желілерді енгізу арқылы сапалы және қауіпсіз бейнебайланысты қамтамасыз ету үшін әртүрлі заманауи технологияларды қолдануға байланысты болып табылады.

Материалдар және зерттеу әдістері.

Пәндік саланы зерттеу, жобаның негізгі міндеттерін анықтау және іске асырылатын қосымшаның моделін әзірлеу мақсатында жоспарланып отырған объектілерді автоматтандыру үшін функционалдық модельдеу кезеңі қажет.

Кейбір модельдеу нотацияларын қарастырайық. IDEF0 жүйе жұмысының нәтижесінде кіріс ақпаратын, басқару элементтерін, механизмдерді және шығыс ақпаратын көрсетеді (1-сурет).

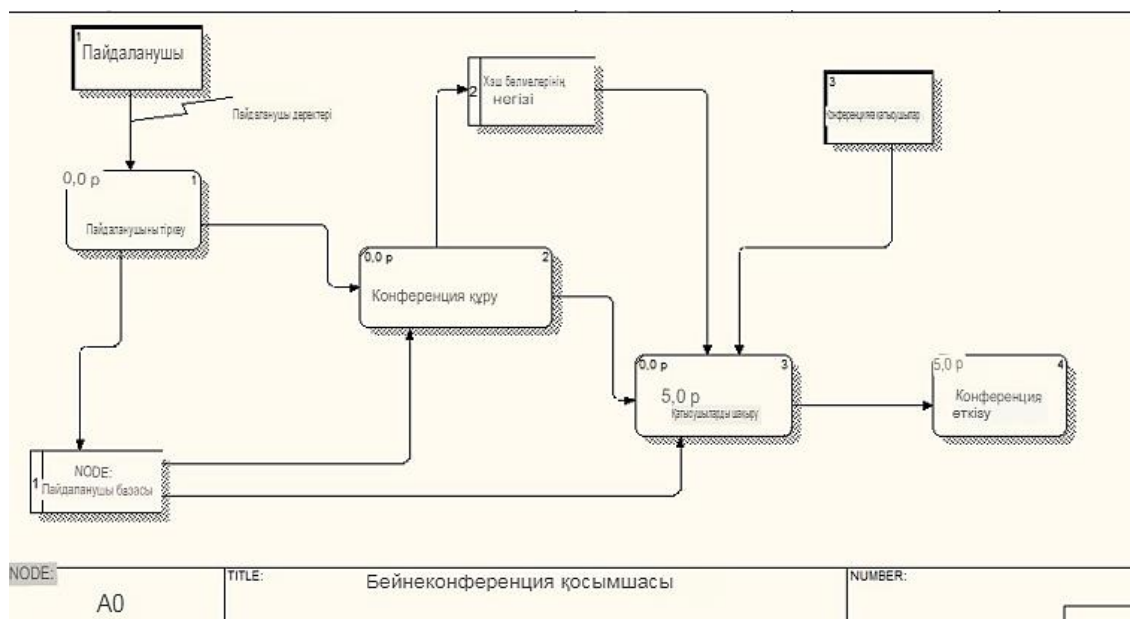


1-сурет. IDEF0 декомпозициясы

Диаграммада (1-сурет) кіріс мәліметі ретінде пайдаланушы мен конференцияға қатысушының деректері болып табылады. Механизмдері бөлме басшысы және модератор болып табылады. Сервер бүкіл процесті, конференция регламентін және қатысушылардың санын шектеуді бақылайды. Өткізілген немесе аяқталған конференция, сондай-ақ өткізу туралы есеп шығыс деректері болып табылады.

Бұл диаграммада 5 негізгі жұмысты атап өтуге болады. Біріншісі пайдаланушыны тіркеу. Тіркелгеннен кейін пайдаланушы қосымшаның мүмкіндіктеріне қол жеткізе алады. Бұл әрекетті бақылау сервердің көмегімен жүзеге асырылады. Келесі қадам – бейнеконференция құру болып табылады. Конференцияны құру кезінде басқа қатысушылар жиналатын бөлме құрылады. Бөлмені құру процесінде оны жасаған пайдаланушы бөлменің басшысы мәртебесін алады, бұл оған бейнеконференция процесін басқаруға мүмкіндік береді.

DFD – бұл деректерді манипуляциялау тұрғысынан ақпараттық жүйелерді модельдеуге арналған нотация. DFD нотациясы кез келген әрекеттерді, соның ішінде тауарды сату немесе жөнелту процесін, өтінімдермен жұмыс жасауды және басқа да әртүрлі процестерді сипаттай алады. DFD диаграммаларында үлкен процестерді құру және оларды іс-әрекеттің декомпозициясы сипаттамасымен ішкі процестерге бөлу мүмкіндігі қарастырылған. Бұл жағдайда DFD нотациясы бүкіл қосымшаның жұмыс процесін қарастырады және оның әртүрлі деректер қоймаларымен өзара әрекеттесуін сипаттайды. DFD нотациясындағы диаграмма 2-суретке сәйкес ұсынылған.



2-сурет. DFD нотациясындағы диаграмма

Бұл диаграммада барлық процестерді бастайтын негізгі нысан пайдаланушы болып табылады. Ол пайдаланушылар қорына жазылатын деректерді енгізу арқылы тіркеуден өтеді. Содан кейін ол конференция құра бастайды. Бұл процесс барысында құрылған бөлме туралы ақпарат бөлмелер қорына хэш-мәні ретінде жазылады.

Жобалық іске асыру мақсатында көптеген түрлі технологиялар мен құралдар қолданылады. Бірінші және ең негізгі технология – HTML болып табылады, немесе белгілеу тілі. HTML web-беттер мен қосымшалар үшін құрылым құруға мүмкіндік береді.

HTML-мен тығыз байланысты келесі технология – CSS. CSS мәтін түсі, қаріп мәнерлері, абзацтар арасындағы қашықтық, фондық суреттер, экранның әртүрлі өлшемдері үшін мазмұнды көрсету және басқа да көптеген мүмкіндіктер сияқты опцияларды басқаруға мүмкіндік береді. JavaScript, бұл жобаладағы ең маңызды технологиялардың бірі, өйткені оның негізінде қосымшаның негізі құрылады. Клиенттік JavaScript – бұл тілдің ең көп таралған түрі, бірақ бұл жобалада ол серверлік тіл ретінде де қолданылады. Шын мәнінде, тілдің өзі өзгермейді, тек қоршаған ортасы өзгереді [2, 3].

Қосымшаны толыққанды әзірлеу үшін оның серверлік бөлігін құру қажет. Бұл жобалада осы мақсаттар үшін Node.js таңдалды.

Node.js – бұл V8 қозғалтқышына негізделген JavaScript кодын орындауға арналған орта. Node.js қолдану арқылы backend' ретінде әзірлеуші JavaScript-те толыққанды әзірлеудің барлық артықшылықтарын алады, атап өтетін болсақ:

- жақсартылған әзірлеуші тиімділігі мен өнімділігі;
- кодты бөлісу және қайта пайдалану;
- жылдамдық;
- өнімділік;
- құралдардың үлкен қорының болуы.

Демек, әзірлеу аз уақытты алады және нәтижесінде жылдам және сенімді бағдарламалық жасақтама жасалады.

Әдетте, сервер мен клиент арасында деректер алмасу кезінде бет толығымен қайта жүктеледі. Кейбір қосымшалар үшін бұл мәселе болып табылады, және оның шешімі бар ол – AJAX. AJAX – бұл бағдарламалау тілі немесе құралы емес, ол динамикалық қосымшаларды жасауды жеңілдететін интуитивті түсінікті әдіс болып табылады. AJAX асинхронды JavaScript және XML дегенді білдіреді. Осы екі технологиядан басқа, ол DOM және DHTML біріктіреді. Негізінде, AJAX бүкіл бетті жаңартпай, фондық режимде серверге деректерді жіберуге және алуға мүмкіндік беретін сценарий болып табылады [4, 5].

Жалпы қосымшаларда деректер серверге және серверден синхронды сұраулар көмегімен жіберіледі. Бұл процесс аяқталғанда, пайдаланушы жаңа деректері бар басқа бетке қайта бағытталады. AJAX әдісін қолдану барсында сұраныс кезінде арнайы объект құрылады, ол мәліметтерді XML түрінде серверге жібереді. Сұрауды өңдеп, жауап алғаннан кейін ағымдағы беттің мазмұны жаңартылады. Бүкіл процесс фондық режимде жүргізілгендіктен пайдаланушы осы сұраныстың жасалғанын білмейді [6].

Бұл жобаны әзірлеу кезінде функционалдылықтан басқа, қауіпсіздікті де ескеру қажет. JWT немесе JSON web Token технологиясы пайдаланушының қауіпсіз тіркелуі мен аутентификациясын қамтамасыз ету үшін қолданылады. JWT - тараптар арасында JSON форматында деректерді қауіпсіз тасымалдауға арналған RFC 7519 белгісі бар стандарт. Ол жинақы, оқуға оңай, кілттердің екі түрін қолдану мүмкіндігі бар: жеткізушіден жабық немесе екі ашық куәлік (IdP). Сондықтан токеннің тұтастығы мен түпнұсқалығын үшінші тараптар тексере алады.

Ақпаратты сақтау үшін қосымша әдетте мәліметтер қорын қолданады. Бұл жобалада MongoDB осындай сақтау орны ретінде таңдалды. MongoDB – бұл мәліметтермен тиімді жұмыс істеуге мүмкіндік беретін, құжаттарға бағытталған және ашық бастапқы коды бар мәліметтер қоры. MongoDB-ді NoSQL форматына жатқызады, яғни мәліметтерді сақтау кесте түрінде жүзеге асырылмайды.

MongoDB мәліметтер қоры PHP, Python, Ruby, C, C++, Go, Java және басқа да көптеген танымал бағдарламалау тілдеріне қолдауды қамтамасыз етеді. Осылайша, осы тілдердің кез келгенінде MongoDB-ді қолданатын қосымшаны құруға болады. Қазіргі

уақытта оны Facebook, Google, Adobe және тағы басқалары сияқты ірі технологиялық алыптар қолданады.

MongoDB және Node.js қолданған кезде әр түрлі қиындықтар туындауы мүмкін. Оларды жеңу үшін әртүрлі құралдар бар. Бұл жобалада Mongoose осындай құрал ретінде әрекет етеді.

Әзірленіп жатқан қосымша нақты уақыт режимінде мәліметтердің тұрақты берілуін қамтамасыз етуі керек болғандықтан, осыны ұйымдастыратын арнайы хаттамаларды қолдану қажет. Осы мақсаттар үшін WebSocket пайдаланылады.

Бұл жобаның негізгі ерекшеліктерінің бірі – пирингтік желілерді пайдалану. Әзірлеуді дәлірек сипаттау үшін негізгі ұғымдарды анықтау қажет. P2P желісі немесе бір рангілі желісі таратылған архитектураны қолданады және орталықтандырылған желілерден, ең алдымен, құру тәсілімен ерекшеленеді [7, 8].

P2P желісінде оның құрамына кіретін қатысушылар бір рангілі түйіндер деп аталады және олар берілген жүктемені бірге қолданады. Түйіндердің ешқайсысында артықшылықтары жоқ және әкімшілік орталығы болмайды.

Белгілі бір мағынада пирингтік желілер компьютерлік әлемдегі эгалитаризмді білдіреді. Қатысушылардың әрқайсысы басқа қатысушылардың алдында тең және бірдей міндеттемелерді орындайды. Бір рангілі түйіндер бір уақытта клиент және сервер болып табылады. Бұл желідегі әрбір ресурс орталықтандырылған бақылаусыз бірге пайдаланылады. Жалпы ресурстар ретінде процессорға жүктеме, желі мүмкіндіктерін пайдалану немесе дискілік жад болуы мүмкін.

Бір рангілі желілерді пайдаланудың негізгі мақсаты ресурстарды бөлу және белгілі бір тапсырмаларды орындау үшін құрылғылар арасында бірлескен жұмысты қамтамасыз ету болып табылады. Жоғарыда айтылғандай, пирингтік желілерді дискілік кеңістікте ақпаратты сақтау және беру үшін пайдалануға болады. Бірақ бір рангілі желілерді қолданудың ең танымал нұсқасы – файлдарды ғаламдық желіде пайдалану. P2P желілері файлдарды бөлісу үшін өте қолайлы, өйткені олар ақпаратты бір уақытта алуға және беруге мүмкіндік береді. Бұл принцип әртүрлі торрент-трекерлердің жұмысында, желінің өткізу қабілеттілігіне және қажетті ақпаратты беретін құрылғылардың санына байланысты жүктеу жылдамдығында қолданылады.

Жалпы, пирингтік желілер осы жоба үшін оның артықшылықтары мен құндылықтарына қарай таңдалды.

Жоғарыда сипатталған барлық технологиялар жобалада қолданылады, бірақ олардың бастысы – WebRTC. WebRTC – бұл браузер арқылы нақты уақыт режимінде сөйлесуге мүмкіндік беретін революциялық технология [9, 10].

WebRTC болмаса, құрылғылар арасында аралық сервер болуы керек. Құрылғылардың бірі серверге хабарлама жібереді, ал сервер оны басқа құрылғыға жібереді. Тағы бір міндетті шарт – құрылғылардың әрқайсысында бір плагин немесе қосымша болуы керек. Мұндай байланыстың танымал мысалдарының бірі – Skype, ол сапалы бейне байланыс қосымшасы болып табылады, бірақ екі құрылғыда Skype қосымшасы болған жағдайда ғана жұмыс жасайды. WebRTC бұл шектеуден арылуға және браузер арқылы бір-бірімен тікелей байланысуға мүмкіндік береді. Жалпы, бұл технология бүкіл жобаның тірек нүктесі болып табылады. Жоғарыда сипатталған көптеген технологиялардың баламасы болуы мүмкін. Бірақ, WebRTC іс жүзінде аудио және бейне ағындарын жіберудің бастамашысы болып табылады және де ол осы жоба аясында үлкен маңызға ие [11, 12].

Нәтижелері және оларды талқылау. Зерттеудің нәтижесі – жобаның түбірлік директориясындағы барлық файлдар жиынтығы бар қосымшаның физикалық құрылымы

болып табылады. Әзірлеуге ыңғайлы болу үшін барлық файлдар тиісті бумаларға орналастырылған. Бұл құрылымды неғұрлым нақты көрсету үшін оны ең түсінікті түрде ұсынатын әртүрлі бағдарламалық жасақтама бар. Бұл жобадан физикалық құрылымды сипаттау үшін MiniTool Partition Wizard бағдарламасы қолданылды. Қосымшаның физикалық құрылымы 3-суретке сәйкес ұсынылған.

| Name | Percent of Parent | Size | Extension | File Type | Percent |
|---------------------------------------|-------------------|----------|------------|--------------------|---------|
| C:\Users\Professional\Desktop\Project | 100.00% | 16.21 MB | .js | Файл JavaScript | 55.19% |
| node_modules | 99.49% | 16.13 MB | .md | Файл "MD" | 12.67% |
| [5 Files] | 0.38% | 63.42 KB | .map | Linker Address Map | 12.44% |
| package-lock.json | 91.50% | 58.03 KB | .json | Файл "JSON" | 7.53% |
| server.js | 7.40% | 4.69 KB | .ts | Видео (TS) | 6.59% |
| package.json | 0.90% | 584 B | .mem | Файл "MEM" | 2.47% |
| .env | 0.18% | 119 B | .lock | Файл "LOCK" | 0.83% |
| .gitignore | 0.02% | 12 B | .html | Файл "HTML" | 0.19% |
| views | 0.06% | 10.35 KB | .patch | Файл "PATCH" | 0.15% |
| room.ejs | 30.12% | 3.12 KB | .xml | Файл "XML" | 0.15% |
| register.ejs | 28.20% | 2.92 KB | .ejs | Файл "EJS" | 0.06% |
| login.ejs | 24.90% | 2.58 KB | .txt | Текстовый докум... | 0.05% |
| profile.ejs | 16.78% | 1.74 KB | .yml | Файл "YML" | 0.04% |
| public | 0.06% | 9.64 KB | .info | Файл "INFO" | 0.04% |
| script.js | 50.28% | 4.85 KB | .eslintrc | Файл "ESLINTRC" | 0.04% |
| style.css | 22.12% | 2.13 KB | .markdown | Файл "MARKDOWN... | 0.04% |
| my-login.js | 15.86% | 1.53 KB | .out | Файл "OUT" | 0.02% |
| my-login.css | 11.73% | 1.13 KB | .ps1 | Сценарий Windo... | 0.02% |
| models | 0.01% | 922 B | .css | Файл "CSS" | 0.02% |
| Room.js | 54.01% | 498 B | .jshintrc | Файл "JSHINTRC" | 0.02% |
| User.js | 45.99% | 424 B | .npmignore | Файл "NPMIGNO... | 0.02% |
| | | | .cmd | Сценарий Windows | 0.01% |
| | | | .flow | Файл "FLOW" | 0.01% |
| | | | .sh | Файл "SH" | 0.01% |
| | | | .iml | Файл "IML" | 0.01% |
| | | | .kef | Файл "KEF" | 0.00% |

3-сурет. Қосымшаның физикалық құрылымы

Бұл бағдарламада жобаның әртүрлі файлдарының арақатынасын анықтауға болады. Қосымшаны толық сипаттау үшін ең қажетті файлдардың құрылымы талданады.

Бірінші, кілттік болып табылатын «server.js» файлы, өйткені ол бүкіл жобаны іске қосады. Файлдың басында қосымшаның ғаламдық параметрлеріне жауап беретін тұрақты айнымалылар жарияланады. Бұл айнымалылар 4-суретке сәйкес ұсынылған.

```

1  const express = require('express')
2  const app = express()
3  const server = require('http').Server(app)
4  const io = require('socket.io')(server)
5  const { ExpressPeerServer } = require('peer')
6  const peerServer = ExpressPeerServer(server, {
7    debug: true
8  })
9  const { v4: uuidV4 } = require('uuid')
10 const session = require('express-session')
11 require('dotenv').config()
12 const bodyParser = require('body-parser')
13 const mongoose = require('mongoose')

```

4-сурет. «server.js»-тің негізгі айнымалылары

Бұл айнымалылар әртүрлі технологияларды қосуға жауап береді. Ең маңыздылары 4, 5, 6, 10 жолдардағы айнымалылар. Төртіншісінде socket.io қосылады. Socket.io – бұл JavaScript кітапханасы, ол клиент пен сервер арасында екі бағытты байланыс жасайды.

Шын мәнінде, бұл кітапхананың көмегімен WebSocket арқылы байланыс жасалады [13].

Peer.js кітапханасы 5 және 6 жолдарда инициализацияланады. Peer.js – бұл WebRTC технологиясын жүзеге асыруға арналған қабықша, бұл жобаның ең негізгі технологиясы. Бұл кітапхананы пайдалану берілген технологиямен өзара әрекеттесуді жеңілдетеді. Оныншы жол бірегей сәйкестендіру нөмірін генерациялауға жауапты [14, 15].

Келесі «server.js»-тің маңызды бөлігі болып мәліметтер қорын қосу болып табылады. Бұл әрекет сурет 5-ке сәйкес ұсынылған кодтың көмегімен жүзеге асырылады. Кодтың бұл бөлігінде MongoDB мәліметтер қорына қосылу сипатталады.

```
36 mongoose.connect(process.env.DATABASE_URL, {
37   useNewUrlParser: true,
38   useUnifiedTopology: true,
39   useCreateIndex: true,
40   useFindAndModify: false
41 })
42 const db = mongoose.connection
43 db.on('error', (error) => console.log(error))
44 db.once('open', () => console.log('Connected'))
45
```

5-сурет. Мәліметтер қорын қосу

Дұрыс және сенімді қосылуды қамтамасыз ету үшін сеанстармен жұмыс жасау қажет. Пайдаланушы үшін «server.js» сессиясында 6-суретке сәйкес ұсынылған функция бар.

```
app.use(function(req, res, next) {
  if (req.session && req.session.user) {
    User.findOne({ 'email': req.session.user.email }, function(err, user) {
      if (err) {
        res.redirect('/login')
      }
      if (!user) {
        req.session.destroy();
        res.redirect('/login')
      } else {
        req.user = user;
        req.session.user = user
        res.locals.user = user
      }
      next()
    })
  } else {
    res.locals.user = null
    next()
  }
})
```

6-сурет. «server.js»-те сессияны тексеру

Кодтың бұл бөлігінде сессияны тексеру жүргізіледі. Егер қосылған пайдаланушы оған тиесілі болса, онда ол жаңартылады. Сондай-ақ, бұл кодта қателерді өңдеу жасалады.

Қорытынды. Бұл жұмыста жобаның тиімділігі тұрғысынан әдістер қарастырылды, жалпыдан бөлшектерге ауысу арқылы, қажет болған жағдайда функцияларды бөледі және декомпозициялайды. Қарастырылған заманауи технологияларды пайдалану, серверді қолданбай пайдаланушыларды қосу мүмкіндігін беретін пирингтік желілерді енгізу арқылы сапалы және қауіпсіз бейнебайланысты қамтамасыз етеді. Бұл ұсынылған зерттеу үшін технологияны таңдауға әсер етті.

1. Application Usage & Threat Report. Available at: <http://researchcenter.paloaltonetworks.com/appusage-risk-report-visualization/> (04.2023);
2. А. Хрусталеv, А. Кириченко "HTML5 + CSS3. Основы современного WEB-дизайна", Наука и Техника, 2018. – С.100-256;
3. Минник Крис, Титтел Эд. HTML5 и CSS3 для чайников. – М.: Диалектика, 2019. – С.150-226;
4. Баялы Э.Т., Абрахманов Р.Б., Садыбеков Р.Ш. PHP және MySQL-де WEB қосымшаларды құру негіздері [Текст]: оқу-әдістемелік құрал / – Алматы: ТехноЭрудит, 2019. – Б. 55-102.
5. Ибрагимов, О.М. Web- бағдарламалауға кіріспе [Текст]: оқу құралы / О.М. Ибрагимов. – Алматы: ТехноЭрудит, 2019. – Б. 33-145.
6. Заяц, А.М. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js: Учебное пособие / А.М. Заяц, Н.П. Васильев. - СПб.: Лань, 2019. – С. 20-120.
7. AJAX Introduction – W3Schools [Электрондық ресурс]. – Қол жеткізу режимі: <https://developer.mozilla.org/ru/docs/Web/Guide/AJAX> (03.2023).
8. Эспозито, Д. Разработка веб-приложений с использованием ASP.NET и AJAX / Д. Эспозито. – СПб.: Питер, 2019. – С. 100-240.
9. JSON Web Tokens Түсінуге болатын бес қарапайым қадам [Электрондық ресурс]. – Қол жеткізу режимі: <https://habr.com/ru/post/340146/> (03.2023).
10. Руководство по MongoDB [Электрондық ресурс]. – Қол жеткізу режимі: <https://metanit.com/nosql/mongodb/> (03.2023).
11. Express оқулығы 3 бөлім: Мәліметтер қорын пайдалану [Электрондық ресурс]. – Қол жеткізу режимі: <https://code.tutsplus.com/ru/articles/an-introduction-to-mongoose-for-mongodb-and-nodejs--cms-29527> (04.2023).
12. Peer-to-peer технологиясы. P2P желілері [Электрондық ресурс]. – Қол жеткізу режимі: <https://ru.bitcoinwiki.org/wiki/Peer-to-peer> (04.2023).
13. Ji, M., Caire, G., Molisch, A.F. Wireless device-to-device caching networks: Basic principles and system performance. Available at: <http://arxiv.org/abs/1305.5216> (04.2023).
14. P2P протоколы обмена данными [Электрондық ресурс]. – Қол жеткізу режимі: <https://habr.com/ru/articles/596983/> (04.2023).
15. WebRTC 1.0: Real-Time Communication Between Browsers [Электрондық ресурс]. – Қол жеткізу режимі: <https://www.w3.org/TR/webrtc/> (04.2023).

References

1. Application Usage & Threat Report. Available at: <http://researchcenter.paloaltonetworks.com/appusage-risk-report-visualization/> (04.2023);
2. A. Khrustalev, A. Kirichenko "HTML5 + CSS3. Fundamentals of modern WEB design", Science and Technology, 2018. - P.100-256;
3. Chris Minnick, Ed Tittel. HTML5 and CSS3 for dummies. – М.: Diagnostics, 2019. – С.150-226;
4. Bayali, A. T. Basics of creating WEB applications in PHP and MySQL [Text]: educational and methodological tool / A. T. Bayali, R. B. Abrahamov, R. Sh. Sadybekov. – Almaty: TechnoErudit, 2019. - B. 55-102;
5. Ibragimov, O.M. Introduction to Web-programming [Text]: educational tool / O.M. Ibrahimov. – Almaty: TechnoErudit, 2019. – B. 33-145;
6. Zayats, A.M. Design and development of WEB-applications. Introduction to frontend and backend development in JavaScript and node.js: Tutorial / A.M. Hare, N.P. Vasiliev. – St. Petersburg: Lan, 2019. – S. 20-120;
7. AJAX Introduction - W3Schools [Electronic resource]. – Access mode: <https://developer.mozilla.org/ru/docs/Web/Guide/AJAX> (03.2023);
8. Esposito, D. Development of web applications using ASP.NET and AJAX / D. Esposito. – St. Petersburg: Peter, 2019. – S. 100-240;
9. Five Simple Steps to Understand JSON Web Tokens [Electronic resource]. – Mode of access: <https://habr.com/ru/post/340146/> (03.2023);
10. MongoDB Guide [Electronic resource]. – Mode of access: <https://metanit.com/nosql/mongodb/> (03.2023);
11. Express Tutorial Part 3: Using Databases [Electronic resource]. – Access mode: <https://code.tutsplus.com/ru/articles/an-introduction-to-mongoose-for-mongodb-and-nodejs--cms-29527> (04.2023);
12. Peer-to-peer technology. P2P networks [Electronic resource]. — Mode of access:

- <https://ru.bitcoinwiki.org/wiki/Peer-to-peer> (04.2023).
13. Ji, M., Caire, G., Molisch, A.F. Wireless device-to-device caching networks: Basic principles and system performance. Available at: <http://arxiv.org/abs/1305.5216> (04.2023);
 14. P2P communication protocols [Electronic resource]. – Access mode: <https://habr.com/ru/articles/596983/> (04.2023);
 15. WebRTC 1.0: Real-Time Communication Between Browsers [Electronic resource]. – Mode of access: <https://www.w3.org/TR/webrtc/> (04.2023).