

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
«АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ»
коммерциялық емес акционерлік қоғамы
IT-инжиниринг кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

Кафедра меңгерушісі

PhD, доцент

_____ Т.С. Картбаев

« ____ » _____ 2019 ж.

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: ЖОО арналған «OnLine Өтініш» программалық қамтамасын әзірлеу

Мамандығы: 5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

Орындаған: Жанғазы М.С. Тобы: ВТк-15-1

Ғылыми жетекші: аға оқытушы Мукапил К.

Кеңесшілер:

Экономикалық бөлім: э.ғ.к., профессор _____ Ж.Г. Аренбаева
«24» _____ 05 _____ 2019 ж.

Өміртіршілік қауіпсіздігі: т.ғ.д., аға оқытушы _____ Ш.Ш. Бекбасаров
«27» _____ 05 _____ 2019 ж.

Есептеу техникасын қолдану: аға оқытушы _____ Ж.С. Айткулов
«20» _____ 05 _____ 2019 ж.

Норма бақылаушы: аға оқытушы _____ К. Мукапил
«29» _____ 05 _____ 2019 ж.

Сын-пікір беруші: т.ғ.к., доцент _____ Л.Ш. Балгабаева
« ____ » _____ 2019 ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
«АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ»
коммерциялық емес акционерлік қоғамы

Басқару жүйелері және ақпараттық технологиялар институты

IT-инжиниринг кафедрасы

Мамандығы 5B070400 – «Есептеу техникасы және
бағдарламалық қамтамасыз ету»

Дипломдық жобаны орындауға берілген
ТАПСЫРМА

Білім алушы: Жанғазы Медет Сәрсенұлы

Жобаның тақырыбы: ЖОО арналған «OnLine Өтініш» программалық қамтамасын әзірлеу

2018 жылғы «26» қазан № 124 университет бұйрығымен бекітілген.

Аяқталған жобаны тапсыру мерзімі: «24» мамыр 2019ж.

Дипломдық жобаның бастапқы мәліметтері (зерттеу (жоба) нәтижелерінің талап етілген параметрлері мен объектінің бастапқы мәліметтері): Ұсынылып отырған дипломдық жобада ЖОО арналған «OnLine Өтініш» программалық қамтамасын әзірлеу. Жобаны орындау барысында MySQL МҚБЖ және PHP тілін қолданамын.

Дипломдық жобада қарастырылған мәселелер тізімі немесе дипломдық жобаның қысқаша мазмұны:

- талдау бөлімі;
- жобалау бөлімі;
- жүзеге асыру және тестілеу бөлімі;
- экономикалық бөлім;
- өміртіршілік қауіпсіздігі;
- А қосымшасы. Техникалық тапсырма;
- Ә қосымшасы. Программа листингі;
- Б қосымшасы. Ендіру актісі.

Графикалық материалдар тізімі (міндетті сызбалар дәл көрсетілуі тиіс):
9 кесте, 32 сурет ұсынылған.

Ұсынылатын негізгі әдебиеттер:


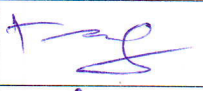
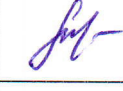

1 Хассон, У. Критерии качества дистанционного образования / У. Хассон. - М.: Высшее образование, 2002. -365 с.

2 Яковлев А.А. Раскрутка сайтов. Основы, секреты, трюки / А.А. Яковлев, В.Г. Ткачев. - СПб. : БХВ-Петербург, 2010. - 344 с.

3 Скляр, Д. PHP. Рецепты программирования / Д. Скляр. - СПб.: БХВ - Петербург, 2007.-736 с.

4 Бенкен Е.С. PHP, MySQL, XML. Программирование для Интернета / Е.С. Бенкен. - СПб. : БХВ-Петербург, 2007. - 310 с.

Дипломдық жобаның бөлімдеріне қатысты белгіленген кеңес берушілер

Бөлімдер	Кеңесшілер	Мерзімі	Қолы
Экономикалық бөлім	Аренбаева Ж.Г.	04.03.2019 - 24.05.2019	
Өміртіршілік қауіпсіздігі	Бекбасаров Ш.Ш.	20.03.2019 02.05.2019	
Программалық қамтама	Айтқулов Ж.С.	29.04.2019 19.05.2019	
Норма бақылау	Мукапил К.	04.04.2019 10.05.2019	


Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Талдау бөлімі	29.10.18 - 28.12.18	орындағандық
Жобалау бөлімі	02.01.19 - 05.02.19	орындағандық
Жүзеге асыру және тестілеу бөлімі	28.02.19 - 12.04.19	орындағандық

Тапсырманың берілген күні «28» қазан 2018ж.

Кафедра меңгерушісі _____ Т.С. Картбаев

Жобаның ғылыми жетекшісі  _____ К. Мукапил

Тапсырманы орындауға алған білім алушы  _____ М.С. Жанғазы

Аңдатпа

Бұл дипломдық жоба университет ішінде құжаттарға немесе қызметтерге сұраныс беруге арналған веб-қосымшаны әзірлеу. Дипломдық жоба PHP тілінде жүйені құруға бағытталып жасалынды.

Жобаны дайындау барысында тестілеу жүйесін жасауға арналған қосымшалар: HTML, PHP, JavaScript, Mysql программалық құралдары пайдаланылды. Веб-бағдарлама университет аумағында студенттердің, оқытушылардың және қызметкерлердің сұраныс алмасуларын жеңілдетуге мүмкіндік береді.

Өміртіршілік қауіпсіздігі мәселесі тексеріліп, ақпараттық жүйені әзірлеу барысында жарықтандыруды есептеу коэффициенті есептелінді. Сондай-ақ, экономикалық бөлімінде жұмыстың орындалу құны, еңбек бағасы және жұмыстың тиімділігі анықталды.

Аннотация

Это разработка веб-приложения для запроса документов или услуг внутри университета дипломного проекта. В данном дипломном проекте был разработан проект, направленный на создание системы на языке PHP.

При разработке проекта использовались приложения для создания системы тестирования: HTML, PHP, JavaScript, Mysql. Веб-программа позволяет облегчить запросы студентов, учащихся и сотрудников на территории университета.

Была проверена проблема безопасности жизнедеятельности, в ходе разработки информационной системы был рассчитан коэффициент расчета освещения. Также в экономической части была определена стоимость выполнения работ, оценка труда и эффективность работы.

Annotation

It is the development of a web application to request documents or services within the University diploma project. In this graduation project was developed a project aimed at creating a system in PHP.

When developing the project, applications were used to create a testing system: HTML, PHP, JavaScript, Mysql. The web-based program facilitates the requests of students, students and staff at the University.

The problem of life safety during the development of the information system was verified and the lighting was calculation coefficient. Also in the economic part was determined the cost of work, labor assessment and work efficiency.

Мазмұны

	Кіріспе	9
1	Талдау бөлімі	10
1.1	Жобаның негізгі мақсаты	10
1.2	Онлайн өтініш жүйесіне шолу	10
1.3	Электронды құжат айналымы жүйесі	11
1.4	Электрондық құжат айналымының негізгі ұғымдары мен анықтамалары	14
1.5	Құжат айналымы ұғымы	14
1.6	Қағаз құжат айналымынан электронды құжаттан артықшылығы	18
1.7	Электрондық құжаттарды басқаруды енгізу	19
1.8	Электрондық құжаттарды басқару үшін не қажет	19
1.9	Құжаттарды дайындауды қамтамасыз ету	21
2	Жобалау бөлімі	23
2.1	Құжаттарды ұйымдастыру	23
2.2	Жүйеге қойылатын функционалдық талаптар	23
2.3	Электрондық құжаттарды басқару жүйесі	24
2.4	Электрондық құжат айналымы жүйесіне арналған типтік талаптар	26
2.5	Microsoft Visio 2016 ортасы	27
2.6	Прецеденттер диаграммасы	27
2.7	Тізбек диаграммасы	28
2.8	Күй диаграммасы	29
2.9	ER диаграммасы	30
2.10	MySQL технологиясы	31
2.11	HTML тілі туралы түсінік	32
2.12	YII2 тірегі	32
2.13	PHP программалау тілі	34
3	Жобаны жүзеге асыру және тестілеу бөлімі	37
3.1	Жүйенің архитектурасы	37
3.2	Пайдаланылған бағдарламалық жасақтамалар	37
3.3	OnLine өтініш жұмысының жалпы схемасы	54
4	Экономикалық бөлім	60

4.1	Экономикалық бөлімнің мақсаты	60
4.2	Бағдарламалық өнімдерді әзірлеудің күрделілігін есептеу	60
4.3	Бағдарламалық өнімді әзірлеуге кеткен шығындарды есептеу	60
4.4	Еңбекақы төлеу шығындарын есептеу	62
4.5	Бағдарламалық өнімнің мүмкін бағасын анықтау	66
4.6	Бағдарламалық өнімнің экономикалық бөлімі бойынша қорытынды	67
5	Өміртіршілік қауіпсіздігі	68
5.1	Табиғи жарықтандыру	68
5.2	Жасанды жарықтандыруды тексеру	72
5.3	Өміртіршілік қауіпсіздігі бөлімі бойынша қорытынды	76
	Қорытынды	77
	Әдебиеттер тізімі	78
	А қосымшасы. Техникалық тапсырма	79
	Ә қосымшасы. Программа листингі	86
	Б қосымшасы. Ендіру актісі	91

Кіріспе

Қазіргі технология заманында құжаттарды, сұраныстарды және ақпарат алмасуды заманауи құрылғылардың көмегімен оңайлату, қауіпсіз және ыңғайлы ету өте маңызды. Интернеттің адамдар арасында кең қолданысқа ие болуы және интернеттің жылдамдығының артуына байланысты веб сайттар үлкен сұранысқа ие. Әрбір адамның қолында смартфондардың болуы интернет арқылы адамдарға қызмет көрсетуді өте ыңғайлы етеді. Университет аумағында студенттердің, оқытушылардың және қызметкерлердің сұраныс алмасуларын жеңілдету мақсатында дипломдық жұмысымды сұраныс жасау бағытында алдым. Студенттер, оқытушылар қашықтықтан өзіне қажетті құжаттарға сұраныс жасап, құжаттың дайын болғандығы жөнінде ақпаратты біле алады. Оқытушылар сұраныс арқылы кабинеттегі бұзылған құралдар жайында инженерлерге хабарлай алады. Дипломдық жұмыс университет ішіндегі барлық сұралымдарды қамтиды. Сұраным сайты оқытушылар мен студенттердің уақытын үнемдеп барлық сұралымдардың тез әрі сапалы орындалуына қол жеткізеді.

Жобамның техникалық жағынан іске асуы үшін php бағдарламалау тілін таңдадым. Php тілінің таңдау себебім бұл тіл өте икемді және күшті тіл, сондықтан кез келген масштабтағы жобаларда кеңінен танымал. php тілі қауіпсіздігімен, оңайлылығымен, пайдаланушылар мен әзірлеушілердің үлкен қауымдастығы болуымен маған ұнады.

1 Талдау бөлімі

1.1 Жобаның негізгі мақсаты

Жобаның негізгі мақсаты университет ішінде құжаттарға немесе қызметтерге сұраныс беруді веб қосымшасын қолдана отырып ыңғайлы, әрі жылдам ету. Бұл тақырыпты таңдау себебім дәстүрлі түрде жасалатын сұранымдар адамның уақытын алады және көп қағаз айналымын талап етеді. Оқытушылар немесе студенттер өзіне қажетті құжатқа онлайн сұраным жасап, смартфон немесе компьютер арқылы құжатының немесе сұралымының дайын болғандығы туралы хабарлама ала алады. Сайтты қолдану барысында қағаз айналымы азайып университеттен берілген құжаттардың толық тізімінің электрондық базасы пайда болады. Құжат айналымын толық бақылауға алып сұраныстың орындалу уақытына қарап жұмысшылардың жұмыс істеу сапасын анықтауға және арттыруға мүмкіншілік туады. Сұранымдарға электронды түрде жауап беру арқылы университеттің қызмет көрсету сапасы артады. Инженерлер істен шыққан немесе жұмыс істеуінде қателігі бар аппараттар туралы жылдам хабар алып бұл мәселелерді шешу уақыты да қысқарады. Келіп түскен сұралымдарды іріктеп тиісті адамына табыстау арқылы жұмысты ұйымдастыру сапасы артады. Сұраныстардың үлкен базасы пайда болған кезде , керек мәліметтерді жылдам іздеп табуға мүмкіндік туады.

Жобаны веб сайт түрінде жасау себебім бұл пайдалану үшін өте ыңғайлы. 24 сағат және 365 күн осы бағдарламаларға қатынасуға болады. Олар кез келген уақытта әлемнің кез келген бөлігінде пайдаланылуы мүмкін. Барлық деректер қоры үнемі қол жетімді. Қажет болған жағдайда қажетті деректерге кіру үшін компьютерді немесе телефонды пайдалануға болады. Үстелдік қосымшалардан айырмашылығы, оларды орнатудың қажеті жоқ, өйткені олар веб-серверде жұмыс істейді. Бағдарламалық жасақтаманы орнату үшін қажетті уақыт пен проблемалар да аз болады. Сонымен қатар, қызмет көрсету және ақауларды жою проблемалары іс жүзінде пайда болмайды. Дискілік кеңістікте істеу керек үлкен резервтік көшірмелерден бас ауруы арқылы өту қажет емес және т. б. Үстел қосымшаларымен салыстырғанда олар экономикалық тиімді. Сонымен қатар, олар компьютерде орын жоғалтпайды, өйткені бағдарламалық қамтамасыз ету серверден іске қосылады. Қызмет көрсету үшін төлеудің қажеті жоқ.

1.2 Онлайн өтініш жүйесіне шолу

Электронды өтініш келесідей тапсырмаларды орындайды:

- интернетпен сұраныс қабылдайды;
- қажетті сұраныс туралы толық мәлімет бере алады (тақырыбы, аты, т.с.с);
- on-line жүйесінде сұранысты рәсімдейді;
- өтініш беруші мен алушы қатынасын құрады;

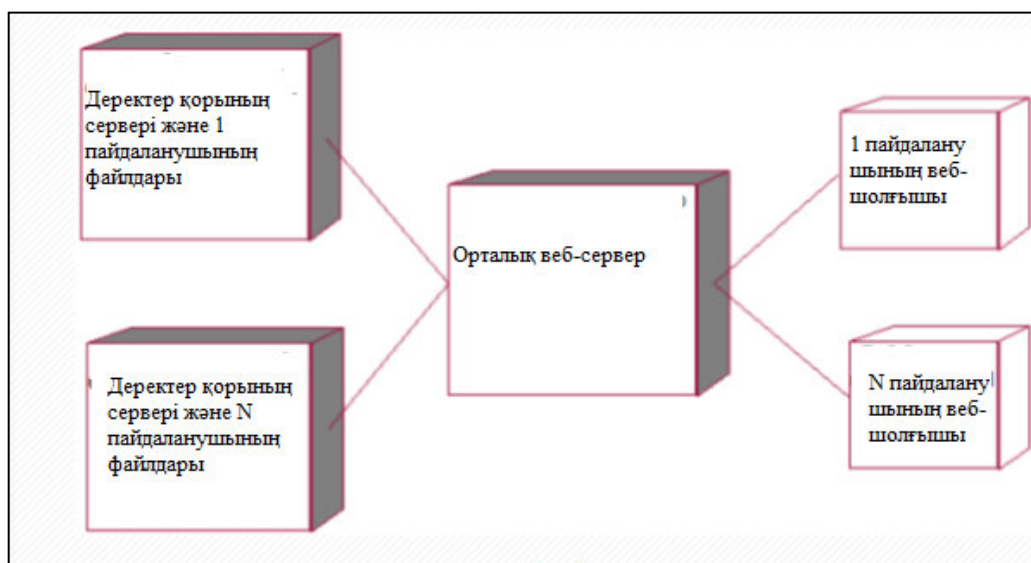
- Өтініш сайты администраторы болса жаңа өтініш енгізе алады;
- интернет дүкенің администраторы болса электронды дүкендегі керексіз тауарларды өшіре алады;
- алдыңғы жасалған сұраныстар тарихын қарастырады.

1.3 Электронды құжат айналымы жүйесі

Электрондық құжат айналымы жүйесі (ЭҚЖ) – компьютерлік желілерде электрондық құжаттарды құру, басқару және тарату процесін қамтамасыз ететін, сондай-ақ ұйымдағы құжаттар ағымына бақылауды қамтамасыз ететін ұйымдастыру-техникалық жүйе. Электрондық құжат айналымы жүйесін (ЭҚЖ) қазіргі заманғы ұйымдастыруда IT-инфрақұрылымның міндетті элементі болып табылады. Олардың көмегімен коммерциялық компаниялар мен өнеркәсіптік кәсіпорындар қызметінің тиімділігін арттырады, ал мемлекеттік мекемелерде Электрондық құжат айналымы технологиясы негізінде ішкі басқару, ведомствоаралық өзара іс-қимыл және халықпен өзара іс-қимыл міндеттері шешіледі. Бастапқыда бұл кластың жүйелері классикалық іс жүргізу міндеттерін автоматтандыру құралы ретінде ғана қарастырылды, бірақ уақыт өте келе міндеттердің кең ауқымын қамти бастады. Бүгінде ЭҚЖ әзірлеушілері өз өнімдерін хат-хабармен және ЖІҚ-мен (ұйымдастыру-өкімдік құжаттармен) ғана емес, сонымен қатар түрлі ішкі құжаттармен (шарттар, нормативтік, анықтамалық және жобалық құжаттамалар, кадр қызметі бойынша құжаттар және т.б.) жұмысқа бағыттайды. ЭҚЖ сондай-ақ электрондық құжаттармен жұмыс істеу маңызды құрамдас бөлігі болып табылатын қолданбалы міндеттерді шешу үшін қолданылады: клиенттермен өзара іс-қимылды басқару, азаматтардың өтініштерін өңдеу, сервистік қызмет жұмысын автоматтандыру, Жобалық құжат айналымын ұйымдастыру және т. б. Іс жүзінде электрондық құжат айналымы жүйесі электрондық құжаттармен жұмыс істеуді қамтамасыз ететін кез келген ақпараттық жүйені атайды.

Қазіргі уақытта көптеген ЭҚЖ бар. Олар көптеген заманауи платформалар үшін жазылған: Microsoft Windows, Unix-ұқсас, BSD. Сол платформаның астында бірнеше түрлі нұсқаларда жазылған қалың және жұқа клиенттерден клиенттік қосымшалардың түрлі түрлері бар. Бірақ барлық қолданыстағы ЭҚЖ елеулі кемшіліктері бар, ол электрондық құжат айналымын жаһандық деңгейге шығуға мүмкіндік бермейді. Кемшілік-компанияның өзара электронды түрдегі түрлі құжаттармен жедел алмасуға мүмкіндігі жоқ. Бұл мәселені шешу электрондық құжат айналымын сапалы жаңа деңгейге шығарады. Шешім web-сайт принципі бойынша ұйымдастырылған бірыңғай ЭҚЖ құру болып табылады. Web-қосымша түріндегі ЭҚЖ ұйымдастыру барлық web-қосымшаларға тән бірқатар артықшылықтарға ие: жаһандық қолжетімділік; орталықтандырылған әкімшілік ету; іркілістерден қорғалудың жоғары дәрежесі. Бұдан басқа, компаниялар осындай жаһандық ЭҚЖ-ны пайдалана отырып, үлкен ұйымдастырушылық және экономикалық артықшылықтарға ие болады: жұмыс станцияларында күрделі және қатты ЭПЖ-ны өрістету және

қолдау үшін білікті мамандарға қажеттіліктің болмауы; серверлік бөліктердің жұмыс істеуі үшін қымбат тұратын жабдықтар мен бағдарламалық қамтамасыз етудің қажеттілігінің болмауы; web-бағытталған ЭҚЖ архитектурасын үш функционалдық блоктарға бөлуге болады: орталық web-сервер, деректер қорының сервері; клиенттік қосымша (web-браузер). Орталық web-сервер жүйенің жүрегі болып табылады және жұмыс логикасын, аутентификация механизмдерін, қол жеткізу құқығын шектеуді, деректерді резервтеуді, ДББЖ мен файлдармен жұмыс істеу ережелерін іске асырады. Деректер қорының сервері деректерді сақтауды, резервтеуді, белгілі бір шамада қолжетімділікті бақылауды, қорғауды жүзеге асырады. Ұсынылған әдістің архитектурасы 1.1-суретте көрсетілген.



1.1-сурет – Деректер қорының сервері

Жаһандық желілерде ақпарат берудің қазіргі жағдайларында Web-технологияларды пайдаланбай іс жүзінде мүмкін емес. Мұндай технологияларды пайдалану жай Web-браузерлер арқылы деректермен жұмыс істеуге мүмкіндік береді, ал олар әртүрлі клиенттік платформаларда орналастырылуы мүмкін. Сол арқылы гетерогенді желілік ортада жұмыс істеу проблемасы шешілді. Web-ортада жұмыс істеу үшін TCP/IP протоколдарын қолдайтын желінің болуы қажет. TCP / IP бар болуы жүйені басқа ақпараттық сервистермен, мысалы, электрондық поштамен оңай біріктіруге мүмкіндік береді. Сонымен қатар, масштабтау проблемасы автоматты түрде шешіледі, өйткені TCP/IP технологиялары үшін пайдаланушының ресурстары мен интерфейсі орналасқан жерде ешқандай айырмашылық жоқ: жергілікті, корпоративтік желі шеңберінде немесе жаһандық желі бойынша бөлінген. Web-технологияны пайдаланған кезде ЭҚЖ-де құжаттарға қол жеткізу үшін әдеттегі Web-браузер арқылы жауап беретін серверлік компоненттер пайда болады. Жоғарыда айтылғандарды негізге ала отырып, электрондық кітапханалардың талаптарына жауап беретін және жұмысты Интернет арқылы жүзеге асыруға

мүмкіндік беретін мамандандырылған функциялары бар құжаттарды басқару жүйесін әзірлеу қажеттілігі туралы қорытынды жасауға болады.

Құжаттарды басқару жүйесінің негізгі функцияларын жүзеге асыру әдістерін анықтаймыз және жүйені құру тұжырымдамасын тұжырымдаймыз. Жүйе келесі функцияларға ие болуы тиіс: деректерді қысу алгоритмдерін қолдана отырып, ақпаратты қалыптастыру және сақтау; серверде деректер қоймасын ұйымдастыру; құжаттарға қол жеткізуді қамтамасыз ету; жүйеге ақпаратты қосу, жүйеден жою; құжат мәтінінен түйінді сөздерді автоматты түрде бөлу; кілт сөз бойынша деректерді жылдам іздеу; каталогты ұйымдастыру үшін деректерді автоматты түрде жіктеу. Жүйе болуы тиіс функцияларды қарастырудан сот жобалау кезінде шешуі қажет негізгі міндеттер мыналар болып табылады: жүйеде ақпарат беру, Ақпарат қоймасын ұйымдастыру, каталогты ұйымдастыру және пайдаланушының ақпараттық қажеттіліктеріне сәйкес деректерді іздеу. Жүйенің ішкі ақпаратын индекстер, оған құжаттар мен басқа да қосымша ақпаратты енгізу туралы мәліметтері бар каталогтың құрылымы сияқты сақтау үшін деректер қорын басқару жүйесін (ДБЖ) пайдаланған жөн. Таңдау реляциялық деректер қорының осындай сипаттамасының болуымен түсіндіріледі, миллиондардан нақты жазбаны таңдаудың аз ғана уақыты сияқты, бұл кітапханалардағы ақпараттың көп саны және жүйенің жылдамдығына қойылатын жоғары талаптар жағдайында өте маңызды. Деректер базасынан іріктеудің аз уақытына кестеге оның бір немесе бірнеше өрістері бойынша индексті құру жолымен қол жеткізіледі. Индекстер ДБЖ арқылы жасалады және әдетте теңдестірілген екілік ағаш алгоритмін қолдана отырып жүзеге асырылады. Сондай-ақ ақпараттың өзін сақтау проблемасын шешу қажет. Бүгінгі таңда электрондық құжаттарды сақтауды ұйымдастырудың екі тәсілі қолданылады. Біріншісі құжаттардың денесі файлдық жүйеде сақталады, екіншісі құжаттарды реляциялық немесе мамандандырылған деректер базасында сақтауды қарастырады. Екінші тәсіл құжаттарды қорғаудың үлкен дәрежесіне ие болғанымен, бірақ келесі негізгі кемшіліктерге ие: қатты дискілерден ерекшеленетін ақпарат тасығыштарды қолдаудағы қиындықтар (сәл ДБЖ магнитооптикалық және басқа да жинақтауыштарды қолдайды) және гетерогенді сақтау жүйелерін құрудың практикалық мүмкін еместігі; электрондық құжаттар жасалатын және өзгертілетін қосымшалармен жұмыс істеген кезде, құжаттардың денелері кез келген жағдайда файлдық жүйе арқылы өтеді, себебі қосымша деректер қорымен тікелей жұмыс істей алмайды, бұл қатты дискіден жазу және оқу операцияларының санын екі есе көбейтуді білдіреді. Құжаттардың үлкен мөлшері болса, бұл жұмыс жылдамдығына қатты әсер етеді. Осылайша, ақпаратты сақтауды құжатты ұсыну форматын әзірлеумен файлдық жүйеде жүзеге асыру қажет. Құжаттарды басқару жүйесінде құжаттарды тиімді іздестіруді қамтамасыз ету үшін ақпараттық-іздестіру кіші жүйесінің болуы қажет. Оны жүзеге асыру үшін классикалық ақпараттық-іздеу жүйелерінде қолданылатын әдістерді қолдануға болады. Негізгі сөздер бойынша жылдам іздеу үшін сөздік түріндегі ақпараттық-іздеу жүйелерінің әдістерін пайдалануға

болады. Барлық ақпарат негізгі сөздер бойынша индекстелуі тиіс, және де кілт сөздерді бөлу автоматты түрде жүргізілуі тиіс. Деректерді іздеу индексі бойынша жүзеге асырылады. Логикалық құжаттар тақырыптық каталог түрінде ұсынылуы тиіс (айдарларға бөлінген). Айдарлар бойынша құжаттарды жіктеуді жасанды нейрондық желілердің мүмкіндіктерін пайдалана отырып, автоматты түрде жүргізген жөн. Құжаттарды басқару жүйесі схемалық түрде 2-суретте көрсетілген түрінде көрсетілуі мүмкін.



1.2-сурет – Құжаттарды басқару жүйесі

1.4 Электрондық құжат айналымының негізгі ұғымдары мен анықтамалары

Электрондық деректер алмасу-бүгінгі таңда әркім тап болатын шындық. Ол Ақпараттық жүйелер, Компьютерлік желілер, Интернет, электрондық пошта және басқа да көптеген құралдар арқылы жүзеге асырылады.

Соңғы онжылдықта басқару процестерін тиімді қамтамасыз етудің жаңа аспаптық құралдары пайда болды және кеңінен таралды. Оның ішінде сөз басқару құжаттарын өңдеуге арналған бағдарламалық қамтамасыз ету туралы болып отыр. Атап айтқанда, электрондық құжат, электрондық-цифрлық қолтаңба, электрондық құжат айналымы жүйесі сияқты ұғымдар пайда болды.

1.5 Құжат айналымы ұғымы

Құжат айналымы-ұйымдағы құжаттардың жасалған немесе алынған сәттен бастап орындау немесе жөнелту аяқталғанға дейін қозғалысы. Құжаттармен жұмыс кешені: қабылдау, тіркеу, тарату, орындалуын бақылау, істерді қалыптастыру, құжаттарды сақтау және қайта пайдалану, анықтамалық жұмыс.

Электрондық құжат айналымы (ЭДО) "қағазсыз іс жүргізу" тұжырымдамасын іске асыра отырып, электрондық түрде ұсынылған құжаттармен жұмыс істеу бойынша бірыңғай механизм болып табылады.

Электрондық түрде ұсынылған құжат немесе электрондық құжат (ЭД) – Ақпаратты компьютерлік өңдеу құралдарының көмегімен жасалған, электрондық цифрлық қолтаңбамен қол қойылуы және тиісті форматтағы файл түрінде машиналық тасығышта сақталуы мүмкін құжат.

Электрондық цифрлық қолтаңба (ЭЦҚ) – электрондық құжаттардың тұтастығын бақылау және түпнұсқалығын растау мүмкіндігін қамтамасыз ететін ақпаратты қорғау құралы болып табылатын өз қолымен қойылған қолдың аналогы.

Бит кибернетикадағы ақпараттың бірлігі болып табылатындығына ұқсас, құжат құжат Құжат айналымы жүйесіндегі ақпараттың бірлігі болып табылады. Құжат айналымы жүйесі құжаттарды сақтайды, олардың тарихын жүргізеді, олардың ұйым бойынша қозғалысын қамтамасыз етеді, осы құжаттардың қатысы бар процестердің орындалуын қадағалауға мүмкіндік береді. В

Құжат айналымы жүйесі енгізілген ұйымдар құжат басқарудың негізгі құралы болып табылады. Бұл жерде тек шешімдер, тапсырмалар немесе бұйрықтар жоқ – осы бұйрықтарды, шешімдерді, тапсырмаларды және т.б. қамтитын құжаттар бар: ұйымдағы барлық басқару құжаттар арқылы жүзеге асырылады. Ұйымның құжат айналымы деп тиісті лауазымды тұлғалардың жұмыс барысында жасаған құжаттардың реттелген орын ауыстыруын түсіну қабылданған.

МЕМСТ Р 51141-98 сәйкес - құжат айналымы – ұйымдағы құжаттардың құрылған немесе алған сәттен бастап орындау немесе жөнелту аяқталғанға дейін қозғалысы. Құжаттармен жұмыс кешені: қабылдау, тіркеу, тарату, орындалуын бақылау, істерді қалыптастыру, құжаттарды сақтау және қайта пайдалану, анықтамалық жұмыс.

1.5.1 Құжат айналымы түрлері

Бастапқы құжаттардың орнын ауыстыру және есепке алу жүргізілетін ұйымдардың шаруашылық операцияларының сипатына байланысты құжат айналымының мынадай түрлері ажыратылады:

Басқарушылық. Негізінде стратегиялық шешімдер қабылдау және олардың уақтылы орындалуын бақылау жүзеге асырылатын жедел, дұрыс және көрнекі басқару есептерін жасау үшін ақпаратты жинау, талдау және қорыту кезеңдерін қамтиды;

Қаржылық. Іс жүргізудің барлық операцияларын қамтиды және ұйымға бизнес-процестерге қатысатын құжаттаманы (шарттар, хат-хабарлар, жоспарлы және талдамалық есептер, нормативтік-заңнамалық актілер және т. б.) басқаруға мүмкіндік береді;

Кадрлық. Ұйымның кадрларын есепке алумен байланысты ақпаратты жинау мен тіркеуді білдіреді: қызметкерлерді қабылдау, ауыстыру, лауазымын

өзгерту және жұмыстан босату, жұмыстан босату және жұмыстан босату, демалыс және тәртіптік жазалар, демалыс кестелері, штаттық кестелер, лауазымдық нұсқаулықтар және т. б. туралы.

Техникалық. Жобалық құжаттаманың өмірлік циклін дайындау және қолдау үшін (күру, ресімдеу, келісу, бекіту, түзетулер енгізу және мұрағаттау) ақпараттық ағындарды үйлестіреді);

Мұрағат. Іс жүргізу барысында құжаттарды мұрағат сақтауға уақтылы беруді және жыл сайын қайта қаралатын істер номенклатурасына сәйкес олардың есебін қамтамасыз етеді.

1.5.2 Құжат айналымын ұйымдастыру тәсілдері

Құжаттаманы есепке алудың келесі тәсілдері бар:

Қағаз – барлық қаржы-шаруашылық операциялар деректемелердің міндетті түрде көрсетіле отырып, тек қағаз тасығышта ресімделетін және құрастырушылардың қолдарымен куәландырылады. Мұндай әдіс маңызды құжаттарды жасауға, айналысқа жіберуге және сақтауға арналған Уақыт пен материалдық ресурстардың шығындарын өтеу.

Сондай-ақ материалдық тасығыш жойылған жағдайда құжатты біржола жоғалту қаупі бар.

Электрондық-барлық бизнес-процестер қағаз тасығышта түпнұсқасы жоқ құжаттарда тек электрондық түрде тіркелуді білдіреді. Құжаттарды заң күшімен қамтамасыз ету үшін плагиат пен құжаттаманы бұрмалауды қорғайтын электрондық-сандық қолдар қолданылады. Мұндай есептеу тәсілі құжаттарды жасауға уақытты азайтуға және олардың орнын ауыстыруға бақылауды күшейтуге мүмкіндік береді.

Аралас-құжаттарды келісу және бекіту қағаз тасығышта жүзеге асырылады, ал құрылымдық бөлімшелер немесе қызметкерлер арасында дайындау және беру компьютерлер көмегімен орындалады деп болжайды. Бұл ретте қағаз тасығыштағы құжаттың түпнұсқасы заңды күшіне ие. Бұл әдіс кәсіпорындағы құжаттама қозғалысының кестесін оңтайландыруды қамтамасыз етеді.

1.5.3 Электрондық құжат айналымының негізгі қағидаттары

Құжатты бір жақты сәйкестендіруге мүмкіндік беретін құжатты бір рет тіркеу.

Құжаттардың қозғалыс уақытын қысқартуға және олардың орындалу жеделдігін арттыруға мүмкіндік беретін операцияларды қатар орындау мүмкіндігі.

Құжаттың (процестің) өмір сүру уақытының әрбір сәтінде құжатты (міндетті) орындауға жауапты тұлғаны сәйкестендіруге мүмкіндік беретін құжат қозғалысының үздіксіздігі.

Құжаттардың қайталану мүмкіндігін болдырмауға мүмкіндік беретін

құжаттық ақпараттың бірыңғай (немесе келісілген бөлінген) базасы.

Құжатты табуға мүмкіндік беретін, ол туралы ең аз ақпаратқа ие болатын құжатты іздеудің тиімді ұйымдастырылған жүйесі.

Құжат айналымы процестері бойынша құжаттардың қозғалысын бақылауға және есептердегі деректерге негізделі отырып, басқарушылық шешімдер қабылдауға мүмкіндік беретін құжаттардың әртүрлі дәрежелері мен атрибуттары бойынша есептіліктің дамыған жүйесі.

Басқармада құжаттарды жасаудың және өндеудің электрондық құралдары кеңінен қолданылады, сондықтан құжат айналымы басқаруды ақпараттық қамтамасыз етудің құрамдас бөлігі ретінде қарастырылады және ұйымның ақпараттық ағындарының қозғалысын – оларды алуды, өндеуді, өндеуді, пайдалануды қамтиды.

Егер соңғы уақытқа дейін іскерлік әдебиетте бір-бірінен дәстүрлі және электрондық құжат айналымын бөлу қабылданған болса, енді дәстүрлі іс жүргізудің ең үздік сапасын құжаттарды құрудың, өндеудің және берудің қазіргі заманғы технологияларымен үйлестіретін құжат айналымының жаңа, қазіргі заманғы түрінің пайда болуы туралы айтуға болады.

Қазіргі уақытта ресейлік құжаттарда жұмыс процесінің екі түрінің комбинациясы бар, ол ірі әкімшілік және өнеркәсіптік орталықтарда электронды басымдыққа ие.

Дәстүрлі жұмыс процесі офис жұмысының нақты бағыттарында, атап айтқанда, құпия құжаттарға (шектеулі пайдалану құжаттарына) қатысты өз позициясын сақтайды.

Осылайша, ұйымның қазіргі заманғы құжаттарды басқару құжаттармен жұмыс істеу үшін электрондық технологияларды басымдықты пайдалану негізінде аралас құжат айналымы ретінде қарастырылуы керек.

Соңғысы құжаттардың, әдетте, қазіргі заманғы технологиялық құралдардың (офистік техниканың, материалдардың және т.б.) көмегімен жасалғанын білдіреді және олар электронды түрде және қағаз түрінде жіберіледі, өңделеді және сақталады.

Ұйымның қазіргі заманғы құжатты басқаруы үш тәуелсіз ақпараттық ағындарды ұйымдастырылған «біріктіру» нәтижесінде ішкі құжаттар, кіретін құжаттар және «сапар» шығатын құжаттар,

Сыртқы ақпарат ағыны кіріс және шығыс құжаттарының қозғалысын қамтамасыз етеді, ал ішкі ақпараттық ағындар кәсіпорын ішіндегі құжаттардың қозғалысын қамтамасыз етеді.

Ішкі құжаттар, әдетте, ұйымның (оның құрылымдық бөлімшелерінің) мақсаттары мен қажеттіліктеріне және құжаттарды әзірлеудің ішкі ережелеріне сәйкес, лауазымды тұлғалар әзірлеген құжаттарды құпияландырады. Басқаша айтқанда, бұл кәсіпорынның ішкі қызметінен туындайтын құжаттар.

Шығыс құжаттары-бұл ұйымның ішінде әзірленген және төмен тұрған басқару органдарына (бағынышты лауазымды тұлғаларға, мекемелер мен ұйымдарға) басшылық ету үшін немесе жоғары инстанцияның (аға бастықтардың) жазбаша нұсқауларын (басшылық құжаттарын) орындау үшін

одан тыс жерлерге жіберілген құжаттар.

1.6 Қағаз құжат айналымынан электронды құжаттан артықшылығы

Көптеген кәсіпорындар электрондық құжат айналымының артықшылықтарын мойындайды, ол келесі артықшылықтарға ие:

- көпкритериалды құжаттарды іздеу;
- құжаттардың орындалуын бақылау;
- құжаттарды тіркеу;
- құжаттарға қарар енгізу;
- желіде құжаттарды үлестірілген өңдеу;
- әр түрлі құжаттарға және жүйенің функцияларына қол жеткізу құқықтарын бөлу;
- құжаттардың бірнеше картотекасын жүргізу;
- құжат жобаларымен жұмыс;
- орындалуына байланысты "папкалар" бойынша орындалған құжаттарды бөлу
- құжаттың орындалу сатылары: түскендер, орындалуда, бақылауда және басқалар;
- стандартты есептерді қалыптастыру;
- электрондық пошта арқылы құжаттар алмасу;
- құжаттарды іске есептен шығару;
- қағаз түпнұсқалар мен құжаттар көшірмелерінің орнын ауыстыруды қадағалау, жүргізу
- құжаттарды ішкі беру тізілімдері;
- лауазымды тұлғалардың, ұйымдардың пайдаланушылық тізімдерін, тақырыптық
- рубрикалар, құжаттар топтары;
- Шығыс баспа пішіндері үлгілерін өңдеу

Электрондық құжат айналымын енгізу құжаттармен жұмыс істейтін қызметтердің (курьерлер, кеңсе қызметкерлері және т.б.) санын азайтуға мүмкіндік береді. Төменде суретте қағаз процесін цифрларға ауыстыру кезінде құжаттармен жұмыс істеудің жекелеген кезеңдерінің уақыты қаншалықты қысқаратыны көрсетілген.

Электрондық құжат айналымы жағдайында сыртқы шарттарды, мысалы, есептілік нысанын өзгерту жөніндегі талаптарды өзгерту кезінде құжат айналымын қайта құруға кететін шығындардан әлдеқайда аз талап етіледі.

1.7 Электрондық құжаттарды басқаруды енгізу

Электрондық құжат айналымы – бұл үкімет пен жергілікті өзін-өзі басқарудың тиімділігін айтарлықтай жақсарту үшін жоғары технологиялық және прогрессивті тәсіл.

Табысты бизнестің кілті әрдайым өз қызметкерлерінің тиімді жұмысы болып табылады. Бірақ, өткенге қатысты ақпаратты өңдеу әдістері

азаматтардың сапалы қызмет көрсету сапасы үшін жақсы емес. Бүгінгі күні ақпараттық ресурстарды пайдаланып, азаматтардың жоқтығына байланысты уақытты қысқарту қажет.

Құжаттарды қолмен көшіру, ұйымдағы қағаз құжаттарының қозғалысын қадағалау және құпия ақпаратты беруді бақылау қызметкерлер құрамының жүктемесін айтарлықтай төмендетеді. Құжаттарды өңдеудің барлық сатыларында атқарушы орындаушылардың жұмысын автоматты түрде жақсартады және құжаттарды неғұрлым дәл және басқарылатын болуы мүмкін уақытты анықтайды.

Электрондық құжат айналымы жүйелерін және ақпаратты сақтау құралдарын бөлісу оның талдауын және есептілігін жеңілдететін ақпаратты жүйелендіруге және үйлестіруге мүмкіндік береді. Үлкен деректер жинақтарында жасырын үлгілерді іздеу үшін ақпарат алу үшін тиісті технологияларға негізделген тиімді шешімдер мен әрекеттерді жасауға болады.

Осының барлығы тек электрондық құжаттарды басқарудың толық жүйесі арқылы жасалуы мүмкін.

Бейресми емес қағазсыз технологиялар ақпараттық басқаруды жеңілдетеді. Олар автоматтандырылған және орталықтандырылған ақпаратпен алмасу және барлық қол жетімді дереккөздерден қажетті ақпаратты ғана шығаратын шешімдер үшін негіз болады.

Электрондық кеңсе және кеңсе менеджменті жүйесі қызметкердің жұмысын оңай, жағымды және мазмұнды ету арқылы жаңа ұйымдастырушылық мәдениетті қалыптастыруға көмектеседі. Ақпараттық технологиялар қызметкерлерге ішкі тапсырмаларды орындауға ғана емес, сондай-ақ кең ауқымды мәселелерді шешу үшін бірге жұмыс істеуге мүмкіндік береді. Ақпараттық технологиялар клиентпен емес, клиенттермен қарым-қатынаста жаңа деңгейге көшу ретінде азаматтардың шағымдарына тікелей жауап беретін катализатор ретінде әрекет етуі мүмкін.

1.8 Электрондық құжаттарды басқару үшін не қажет

Электрондық құжаттарды басқару – бұл айтарлықтай оңтайландыратын, сонымен қатар бизнесті түбегейлі өзгертетін технологиялардың үйлесімі. Мұндай кең ауқымды жүйе нөлге тең емес, өйткені іс-әрекеттегі ұзақ мерзімді параличке әкелуі мүмкін іс-әрекеттің барлық маңызды аспектілеріне әсер етеді. Осылайша, электрондық құжат айналымы мен басқару жүйесін енгізе бастағанға дейін, әрқайсысы жаңа сапалы жаңа технологияларға дайын екендігімен мұқият қарау маңызды.

Электрондық құжат айналымын енгізу құжаттармен жұмыс істеуге тартылған барлық қызметкерлер үшін төмендегілерді талап етеді:

– бағдарламалық қамтамасыз етуді орындау үшін жеткілікті компьютерлік техника;

– аппараттың 100% барлық дербес компьютерлердің қызметкерлері үшін міндетті болып табылады;

– барлық жұмыс аймақтары арасында жеткілікті өткізу қабілеті бар коммуникациялық құралдар;

– электронды құжат айналымына ауысуға мүмкіндік беретін бағдарламалық жасақтаманы пайдаланатын автоматтандырылған кеңсе басқару жүйесі;

– менеджерлердің электрондық қолтаңбаларын құжатта қолжазба қолымен психологиялық оқыту;

– электрондық қағаз құжаттарын аударуға арналған қызметтер мен техникалық мүмкіндіктер.

Электронды құжаттарды электронды құралдармен тұрақты қол жеткізе алмайтын барлық қызметкерлер электронды құжат айналымын таратуға барлық негіздеме жұмысқа нақты қатысудан бас тартуы керек. Құжатқа келген барлық қызметкерлер электронды түрде жарияланымдар тек олармен таныса алады. Құжаттамаға қатысу, тіпті күнделікті немесе пікірталасқа қатысу автоматты түрде электронды құжат айналымы жүйесі бар дербес компьютерлік техниканың болуын талап етеді.

Бірыңғай ақпараттық жүйені енгізу аппараттық және бағдарламалық қамтамасыз етуді орнатуды ғана емес, сондай-ақ бизнес-процестерді қайта ұйымдастыруды және кадрларды қайта даярлауды талап етеді. Жүйені енгізу кезінде бизнес процесінің көлемі неғұрлым аз болса, бағдарламаның орындалуын күту неғұрлым ықтимал болса, жүйенің өнімділігі қаншалықты жоғары болады. Сондықтан күрделі ақпараттық жүйелер әрдайым бөлімшелерде жүзеге асырылады, содан кейін қызметтің салыстырмалы түрде жабық әрекеттерін автоматтандыруға, содан кейін оларды интеграциялық жүйеге біріктіруге мүмкіндік береді.

Технологиялық түрде электрондық құжат айналымы жүйесі оқырмандар мен құжаттарды дайындауды және олардың сыртқы ортасын қамтитын кешенді жүйе болып табылады. Осылайша, автоматтандыруды уақытында ұлғайту үшін ұжымдық жұмыстарды ұйымдастыру үшін автоматтандырылған кеңсе менеджменті жүйелері мен құралдарын енгізу қажет.

Офисті автоматтандыру және электронды құжат айналымы жүйесін бастапқыда енгізу және пайдалану электрондық құжаттарды басқаруды және кеңсе қызметін басқаруды жеңілдетумен қатар, электрондық құжат айналымын іс жүзінде енгізуге мүмкіндік береді, электрондық құжат айналымы мен автоматтандырылған басқару жүйелерін жібереді. Жалпы алғанда, психологиялық тренинг қажетті техникалық көмекке ие болудан гөрі маңызды. Технологиялық тұрғыдан алғанда, екі жүйеде де өндіруші дайындаған заманауи бағдарламалық қамтамасыз етуді пайдалану бизнеске бизнес-процестерді толық электрондық форматқа көшіруге мүмкіндік береді.

Электрондық құжат айналымына көшу процесінде қызметкерлердің негізгі технологиялық мәселелері – құжаттардағы электрондық қолтаңбалардың электрондық қолтаңбаларын пайдалану. Бұл технологияны түсінбей немесе жүзеге асырмай, қағазды өңдеуге ауысу мүмкін емес.

Қағазсыз технологияға кіріс және шығыс құжаттардың толық аудармасы талап етілмейді. Мұндай шектеу компьютерлік техниканың және Интернет емес пайдаланушылардың құқықтарын бұзуы мүмкін. Өтініш берушілердің дәстүрлі қолтаңбаларымен қабылданған қағаз құжаттарын тіркеудің классикалық ерекшеліктері электрондық құжат айналымы жүйесінің ерекшелігі болып табылады. Бұл «ақпараттық теңсіздік» алынып тасталуы керек. Егер сіз осы мәселені шешуге жеткілікті көңіл бөлмесеңіз, компьютер техникасы мен интернетті жоғалтқандар ақпараттық қоғамға әсер етеді.

1.9 Құжаттарды дайындауды қамтамасыз ету

Электронды құжат айналымы процесінде автоматтандырылған құжаттарды сол процестерден басқару үшін құжат құжаттарын дайындаудағы айырмашылық жақсы емес. Жұмыс процесін автоматтандыруға арналған шешімнің сипаттамасында, құжаттарды жасау бойынша ағымдағы жұмыс электрондық түрде жүзеге асырылған деп айтылды. Бұл процедура қызметкерлер үшін әдеттегі болып табылады және шешімнің мақсаты жұмыс ұйымдастырудың заманауи әдістерін ұсыну болып табылады.

Автоматтандырылған құжат айналымындағы ең заманауи шешім пайдаланылды және іріктеу критерийлері тек құжат айналымымен үйлесімді және онлайн жұмыс үрдісінде жаңа ештеңе жоқ.

Электрондық құжат айналымы және жұмыс процесі құжаттарды дайындау үдерісін реттейді – құжаттарды куәландыру және бекіту процесі, соның ішінде электрондық аналогты қолтаңбаларды пайдалану.

Сонымен қатар, электрондық құжат айналымы жүйесі сізге «ресми» құжат ағындары мен мұрағаттар арқылы өңделетін бейресми ақпарат пен құжаттары бар объектілердің бір қоймасымен жұмыс істеуге мүмкіндік береді.

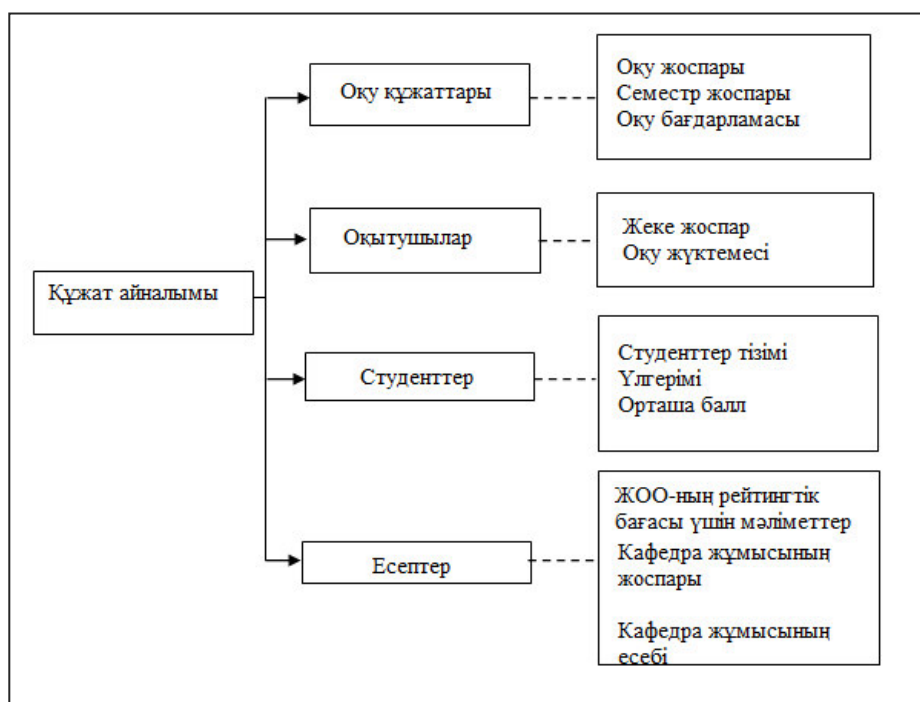
Осының барлығы құжаттардың тиімділігін айтарлықтай жақсартады, себебі барлық ақпараттық сайттар деректерді іздеу және талдау үшін бірегей құралдарға ие. Жаңа құжаттарды дайындау кезінде фрагменттер немесе аналитикалық материалдар тек қызметкерлердің жеке немесе ұжымдық тәжірибесіне ғана емес, сондай-ақ барлық қолданыстағы құжаттарды (деректерге қол жеткізуді абсолютті бақылау) алынған мәліметтер негізінде оңай пайдалануға болады. Бұл дайындалған құжаттардың жұмысын және сапасын айтарлықтай жақсартады. Құжаттарды электронды түрде беру осы терминге сәйкес «білім менеджменті» тұжырымдамасында бірыңғай білім базасын құруға және басқаруға мүмкіндік береді.

2. Жобалау бөлімі

2.1 Құжаттарды ұйымдастыру

Құжаттармен толыққанды жұмыс істеу үшін ЖОО кафедрасы үшін құжат айналымын автоматтандыру жүйесін веб-қосымша түрінде әзірлеу қажет. Жүйені Алматы энергетика және байланыс университетінің кафедрасына пайдалану жоспарланып отыр, кейіннен электрондық құжат айналымы жүйесін кез келген мемлекеттік жоғары оқу орнының кез келген кафедрасында пайдалануға болады. Әзірленетін веб-қосымшаға ауысуға болатын сілтеме кафедраның сайтында орналастырылады.

Кафедра қызметкерлерінің жұмысын құжаттармен қарастыру нәтижесінде құжат айналымының келесі сызбасы әзірленді:



2.1-сурет – Кафедра құжат айналымы сызбасы

2.2 Жүйеге қойылатын функционалдық талаптар

Әзірленетін жүйеге және оның функционалдығына қойылатын негізгі талаптарды қарастырайық:

- пайдаланушы атқаратын қызметіне байланысты құқық беру;
- құжатты қосу және жою;
- құжаттың стандартты үлгісін қосу және жоюды қамтамасыз ету;
- құрал құрылған құжаттың немесе үлгінің деректер базасына сақталуын қамтамасыз ету;
- файлдық каталогты құжаттармен қарау мүмкіндігі;

2.3 Электрондық құжаттарды басқару жүйесі

Электрондық құжаттарды басқару жүйесі – ақпараттық технологиялармен жұмыс жасау процесін автоматтандыру жүйесі (жасау, өзгерту, сақтау, іздеу, жіктеу және т.б.), сондай-ақ қызметкерлер арасында. Бұл жағдайда құжаттар ең алдымен құрылымдық емес электронды құжаттарды (Word, Excel файлдары және т.б.) білдіреді. Әдетте, EDMS электронды құжаттарды мұрағаттауды және бизнес үдерісін автоматтандыруды және кеңселік функционалды қолдау жүйесін қамтиды.

Бүгінгі күні электрондық құжат айналымы жүйелерінің көп саны бар. Сонымен қатар, «құжат айналымы» мен «есепке алу» арасындағы айырмашылық қажет. Жазбаларды басқару - бұл кеңсе тәжірибесінде құжатпен жұмыс істеу үшін ресми ережелер жиынтығын қолдану үшін қолданылатын термин. Кеңсе жұмысының маңыздылығына сәйкес құрылуға болатын құжаттарды басқару жүйесі бар. Дегенмен, алдымен осы мақсаттарды қолдауға бағытталған жүйелер бар және жұмыс ағыны жүйесі деп аталатын жалпы функциялар жеткіліксіз.

Кез-келген іскерлік жүйе төменде көрсетілген санаттардың әрқайсысының элементтерін қамтуы мүмкін, бірақ олардың көпшілігі өнімнің орналасуымен байланысты болатын салалардың бірінде айқын аспекті бар.

Электрондық құжаттарды басқару жүйелерінің классификациясы:

– бүкіл әлем бойынша «брондалған» электрондық құжат айналымы жүйесі:

- стандартты ерекшеліктер жиынтығы;
- белгілі бір ұйымның қажеттіліктерін толық қанағаттандыра алмау;
- сатып алу мен монтаждаудың төменгі уақыттық шығындары;
- салыстырмалы төмен шығындар;
- әрбір іске асырылған жұмыс орнына лицензияны алу қажеттілігі.
- максималды жекеленген жүйе;
- үлкен уақыттық шығындар;
- дамудың жоғары құны;
- ілеспе шығындар: қызметкерлерді оқыту құны, жаңа жабдықтар мен бағдарламалық қамтамасыз ету сатып алу.

Біріккен электронды құжаттарды басқару жүйесі:

- қажетті қосымша модульдерді әзірлеуге арналған негізгі платформа;
- компанияның қажеттіліктеріне толық сәйкес келуі;
- даму және іске асыру үшін төмен шығындар;

Бағаға кіреді:

- Тапсырыстың күрделілігіне, негізгі платформа құны мен жеке бейімделу құнын;
- Клиентке өнімге құқықты беру;
- Даму және қолжетімділік;
- Ыңғайлы интерфейс;

– Қолданыстағы офистік қосымшалармен өзара әрекеттесу.

Орындалған функциялардың санына сәйкес электрондық құжат айналымы жүйесі:

– Кеңейтілген сақтау және іздеу жүйесі (электрондық мұрағат - EA). Электронды мұрағат - бұл ақпаратты тиімді сақтауға және іздеуге бағытталған кеңсе жүйесінің ерекше жағдайы. Кейбір жүйелерде тиімді сақтауды ұйымдастыру арқылы айқын іздеу, семантикалық іздеу және т.с.с. сияқты толық мәтінді іздеу әдістерінде нақты көрсетіледі: HSM, ақпараттық сақтау орталықтарының кең ауқымы және басқалары.

– Кеңейтілген жұмыс үрдісі (WF) жүйесі. Бұл жүйелер алдын-ала анықталған бағыттардағы («қатты сызықтар») белгілі бір объектілердің қозғалысына кепілдік беру үшін арнайы әзірленген. Объект кез-келген кезеңде өзгертілуі мүмкін, сондықтан әдетте «жұмыс» деп аталады. Мұндай жүйелер «жұмыс ағындары» деп аталады (өкінішке орай, осы термин үшін орыс тілінде дәлдік жоқ). Құжаттарды жұмыстарға қосуға болады, бірақ құжаттар осы жүйелердің негізгі объектісі болып табылмайды. Осы жүйелерді пайдалана отырып, барлық кезеңдерде бұрыннан белгілі және тіркелетін нақты қадамдарды ұйымдастыруға болады.

– Ұйымдастыру және деректерді жинақтауды қолдайтын жүйелер. Жалпы алғанда, осы екі «алдыңғы» элементтің элементтерін біріктіретін гибридік жүйелер. Бұл жағдайда жүйедегі негізгі ұғымдар құжаттың өзі және оны орындау үшін қажет болуы мүмкін. Document Movement басқарушысы («кіріс құжатын сипаттайды») құжаттаған кезде ұйым үшін «қатты» және «еркін» бағыттау қажет, сондықтан екі технология да осындай жүйелерде болуы мүмкін. Бұл жүйелер мемлекеттік құрылымдарда, алдыңғы қатарлы иерархиялы ірі компаниялар кеңселерінде, құжат айналымы үшін нақты ережелер мен рәсімдерде белсенді қолданылады. Сонымен бірге қызметкерлер бірлескен құжаттарды жасайды, шешімдер дайындайды, қабылдайды, атқарады немесе басқарады.

Жүйені тиімді басқару, ауқымды деректерді өңдеу, өндірістік басқарудың автоматтандырылған жүйелерімен интеграциялау, көлемді бөлу, кезең-кезеңмен жүзеге асыру, бөлуді бөлу, күрделі ұйымдық құрылым, ауқымды кәсіпорындарға қолжетімділіктің негізін ұйымдастыру басқалары.

Жалпы қолдау жүйесі (ынтымақтастық). Бұл қазіргі заманғы жүйедегі нарықтық жағдайлардың өзгермелілігін түсіну үшін және «ауыр балластсыз» жылдам және тиімді шарлау үшін «әділетті заттар» болу қажеттілігіне байланысты жұмыс үрдісінің жүйесінде жаңа үрдіс. Алдыңғыдан айырмашылығы, мұндай жүйе ұйымда иерархиялық түсініктерді қамтымайды, бұл жұмыс процесін тіркеуге қатысты емес. Олардың міндеті - ұйымдағы адамдардың географиялық бөлінуін сақтау және осы жұмыстың нәтижелерін сақтау үшін бірге жұмыс жасау. Әдетте «порталдар» терминінде қолданылады. Олар құжаттарды сақтауға және жариялауға, интранетті іздеуге, ақпарат іздеуге, талқылауға, жиналыс құралдарына (нақты және виртуалды) қызмет көрсетеді. Бұл жүйелер жылдам дамып келе жатқан коммерциялық

компаниялар, ірі фирмалар мен мемлекеттік органдардың жұмыс топтары арасында клиенттерді табады.

Жетілдірілген жүйелер. Мысалы, клиенттермен қарым-қатынастарды басқару (CRM), жобаны басқару, шот жіберу, электрондық пошта және т.б. (Мүмкіндіктердің, кеңсе тауарларының және CRM қызметінің күрделілігі)

2.4 Электрондық құжат айналымы жүйесіне арналған типтік талаптар

Егер сіз техникалық сипаттамаларды дайындау үшін стандартты хатты сақтасаңыз, стандартты электрондық құжат айналымы жүйесіне әдеттегі пайдаланушының кіруіне қойылатын талаптар төменде сипатталған болуы мүмкін.

Электрондық құжат айналымы жүйесі:

- құжаттарды және олардың сипаттамаларын сенімді сақтауды қамтамасыз ету;

- құжаттың өмірлік циклі (оны жасау, репликалау, жариялау, жойылған құжатқа қол жеткізуді бұғаттау, мұрағатта сақтауға беру);

- пайдаланушыға құжаттардың әр түрлі түрлерін көруге, олар үшін карталарды жасауға және өңдеуге мүмкіндік береді;

- құжаттарды тиімді тексеру үшін санат иерархиясын жүргізу;

- карточка бойынша құжаттарды іздеу, сондай-ақ толық мәтінді іздеу;

- ұйымдардың иерархиялық құрылымына негізделген, рөлдердің қағидасына сәйкес, жеке пайдаланушылық деңгейде құжаттарға қолжетімділікті қамтамасыз ету;

- HSM технологиясын қолдау;

- барлық пайдаланушыларды және жүйемен байланысты оқиғаларды тіркеу;

- әкімшілік құралдарын әзірлеу керек;

- ақпаратқа қашықтан қол жеткізуді қолдау;

Жетілдірілген жүйелер қолдауы керек:

- үздіксіз жұмысын қамтамасыз ету үшін кластерлік технологиялар;

- географиялық бөлінген ұйымдар;

- деректерді сақтау және беру шифрлеу алгоритмдері;

- цифрлық қолтаңба

Сәулеттік талаптар:

- арнайы бағдарлама серверінің болуы;

- жұқа клиенттің болуы;

- браузерлерді пайдалана отырып, құжаттарға қол жеткізуді қолдау;

- масштабталуға арналған көпплатформа;

Басқа жүйелермен ашықтық пен интеграцияға қойылатын талаптар:

- ағындық кіріс құжаттарымен интеграциялау;

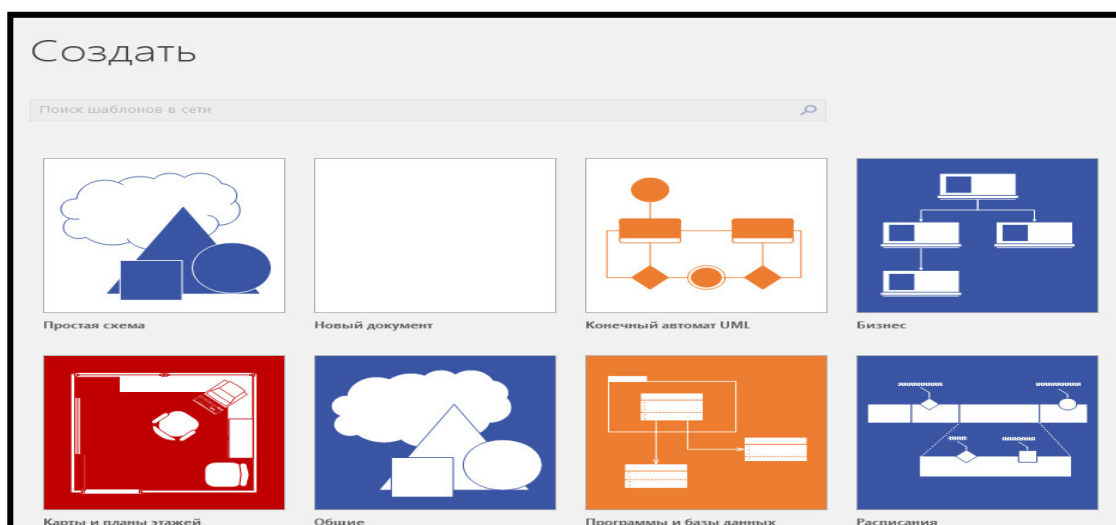
- электрондық поштаға интеграциялау;

- дамыған бағдарламалық интерфейстің (API) болуы;
 - жүйелік пайдаланушылар тізімін сақтау және синхрондау үшін стандартты каталог қызметтерімен (мысалы, LDAP) интеграциялау;
 - пайдаланушы интерфейсін нақты тапсырмаларға бейімдеу мүмкіндігі;
 - жүйені өзінің арнайы компоненттерімен толықтыру мүмкіндігі;
- Құжат атрибуттарын сақтау үшін сыртқы дерекқорды пайдаланған жағдайда әртүрлі ДҚБЖ-мен жұмыс істеу үшін деректер құрылымы мен құралдарының егжей-тегжейлі сипаттамасы болуы керек.

2.5 Microsoft Visio 2016 ортасы

Microsoft Visio Standard 2016 – бұл үлкен трафареттердің күшті графикалық даму платформасы. Қарапайым, оңай түсінуге болатын диаграммалар күрделі ақпаратты жеңілдетуге мүмкіндік береді. Visio стандартына бизнес, трафик кестелері, ұйымдастыру кестелері, базалық блоктық диаграммалар және жалпы көп мақсатты графика кіреді.

Деректер желісіне, өндірістік кәсіпорынға немесе бизнес ақпаратқа байланысты бизнес процесі олардың тиімділігін жылдам анықтауға мүмкіндік береді. Негізгі ақпаратыңыз жаңартылған кезде графика және визуалдар автоматты түрде жаңартылады. Microsoft Visio 2016 2.1 нұсқасында көрсетілген.



2.1-сурет – Microsoft Visio 2016 құралы

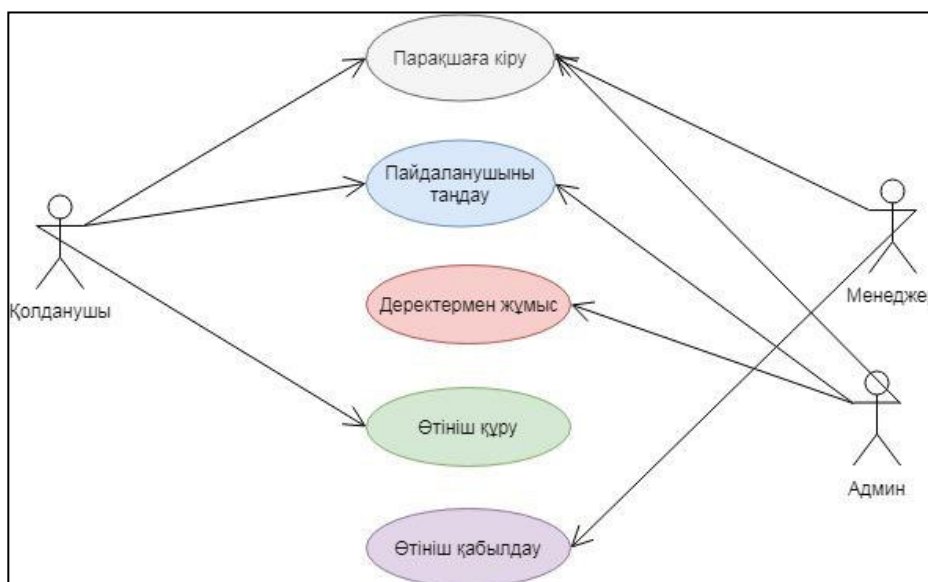
2.6 Прецеденттер диаграммасы

Жүйені құру үшін объектінің нақты аймағын білу қажет. Жүйенің өнімділігі туралы ақпаратыңыз болуы керек. Графикалық жүйелер ретінде әрекет ететін транзакцияларға арналған. Бұл әдіс жүйелік нысан ретінде анықталған жүйе тізіміне сәйкес келмейді. Жүйенің функционалдық құрылымы, барлық радиобайланыс объектілерінің әрекет жоспарлары

жасалды. Презентация диаграммасы пайдаланушының процесті модельдеу және толық реакцияның толық көрінісі үшін арналған. Өнімділік диаграммасы 2.2-суретте көрсетілген.

Жүйе қалай жұмыс істейді және сізге байланысты. Объектілі модель айнымалылар бөліміндегі модель үлгісіне байланысты. Динамикалық моделдеу «сыртқы» деп аталады. Ол типтік емес тапсырмадан басталады және бұл ақпаратты ашады. Осылайша, ақпараттық құрылым пайда болады, яғни сізге қажет жұмыс. Ішкі қозғалыстар мен олардың көзқарасы – жеке жоспар.

Қолдану жағдайлары жүйе мен субъект арасындағы диалогты модельдеуге мүмкіндік береді, сонымен бірге, функцияны көрсете алады. Жүйені әртүрлі пайдалану түрлері көптеген жолдармен пайдалануға арналған. Бағдарлама түрлері - белсенді тақырып анықталған транзакция жүйесімен жасалатын мәмілелер тізбесі, бұл дегеніміз, қажетті нәтиже оңай алынуы мүмкін.



2.2-сурет –Прецеденттер диаграммасы

2.7 Тізбек диаграммасы

Желі картасы объектілердің үйлестірілген келісімдерін тұрақты негізде анықтайды және жоспарлардағы мүмкіндіктерді іске асыру кезінде өзгеретін нысандардың хабарламалық тізбектерінің дәйекті сыныптары мен құрылымдарын көрсетеді.

Сегменттелген диаграммалар әдетте Logiac Via пакетін мақсатты пайдалануды жүзеге асыруымен бірге анықталады. Жетектің сызбасы төрт негізгі элементтен тұрады:

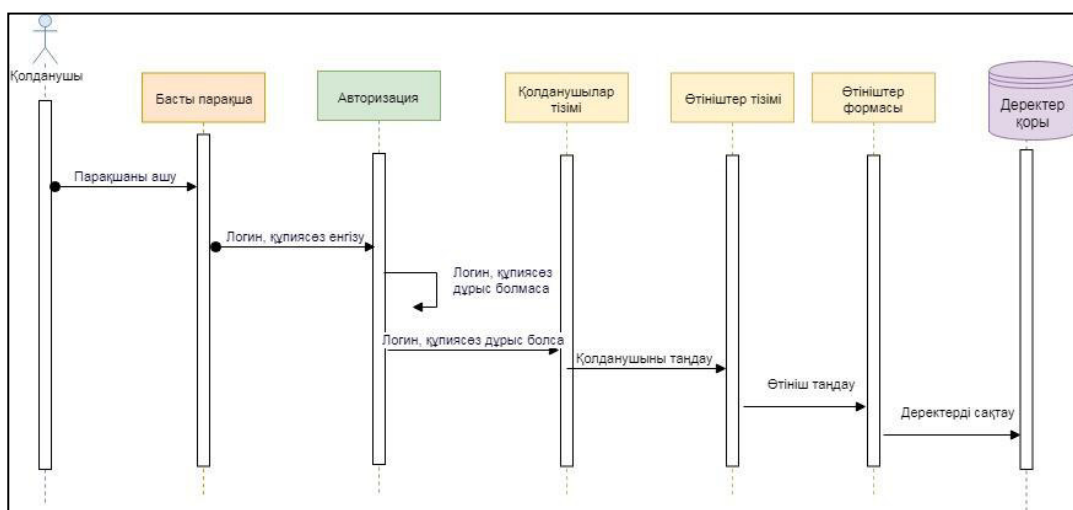
Бұрынғы мәтін. Ол жазылған және сол жақтан басталады. Сол терезеде мінез-құлықтың анықтамасы және өңдеу барысында қолданылатын деректер белгілі;

Егер нысандар «object-class» деп аталмаса, нысан объектісі мен сынып нысанының нысаны жазылады;

Протоколда осы тармақта көрсетілген объектінің келесі пәнге сәйкес келуі көрсетілуі тиіс. Ол бір мезгілде орындалуы мүмкін және осы жүйенің реакциялық реакциясы болуы мүмкін;

Әдістері. Бұл тіктөртбұрыш. Олар ең жақсы жолмен сипатталады. Яғни, осы әдістерге не кіреді. Тік бұрыштың ұзындығы бұрыштағы басқару панелін көрсету үшін пайдаланылуы мүмкін: тікбұрышты жабу немесе жабу әдісінің нақты соңы қол жетімді. Бұл үшбұрыш цикл үшін нысан болып саналады.

Тікбұрыш UML талаптарына байланысты нысандар кестесінде көрсетіледі. Нысанды 3 атауға өзгерте аласыз: тек нысан атауы, нысан және сынып атауы немесе сынып атауы, тізбектік диаграмма 2.3-суретте көрсетілген.



2.3-сурет – Менеджер үшін тізбек диаграммасы

2.8 Күй диаграммасы

Мемлекеттің нақты іс-әрекеттері болмаған жағдайда басқа оқиғаны күтетін келісімнің жағдайы. Нысанның күйі бірнеше немесе одан да көп элементтермен сипатталады.

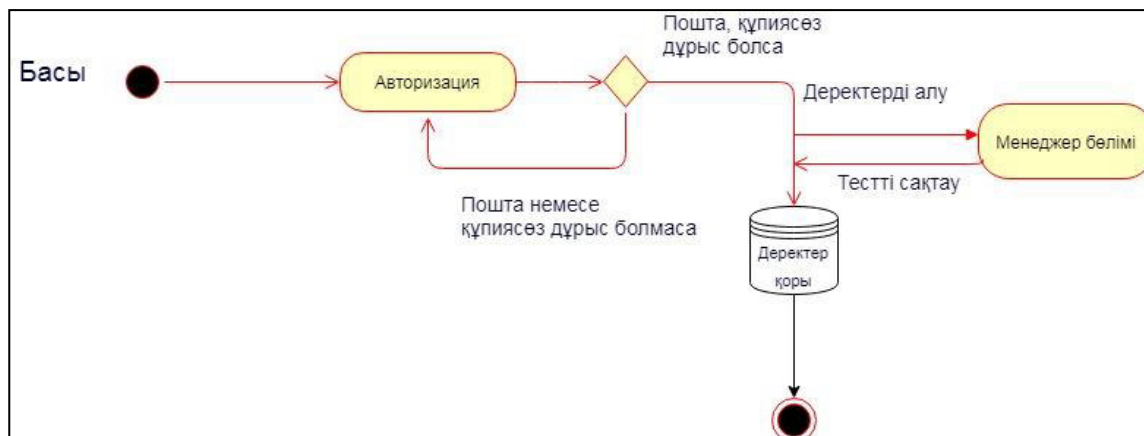
Күй кестесіндегі жағдай оқиғалардың графикалық көрінісіне немесе нысанның біреуінен екіншісіне орналасуына қалай өзгертіні туралы ескертеді.

Нысан уақыт бойынша өз позициясын өзгерте алатын жағдайды түсіну үшін нақты мүмкіндіктер бар. Кәсіпорынның күйі оның атрибуттары мен атрибуттарымен сипатталады.

Мемлекет сынып төлсипатын анықтайды. Кестенің спецификациясы сыныптардың қатарын анықтайды, олардың кейбіреуі объектінің жай-күйін бір мемлекеттен екіншісіне өзгерте алады.

Объектілердің модельдік жай күйі мемлекеттік диаграмма арқылы жасалады: сызба схемасы (автоматты) бұл өзгерістер мен өзгерістерді көрсетеді. Үлгі моделі негізгі сыныптар үшін тіркеледі.

Объектілер белгілі бір нүктеден асатын нүктеге жетеді және кесте 2.4 кестесінде көрсетілген.



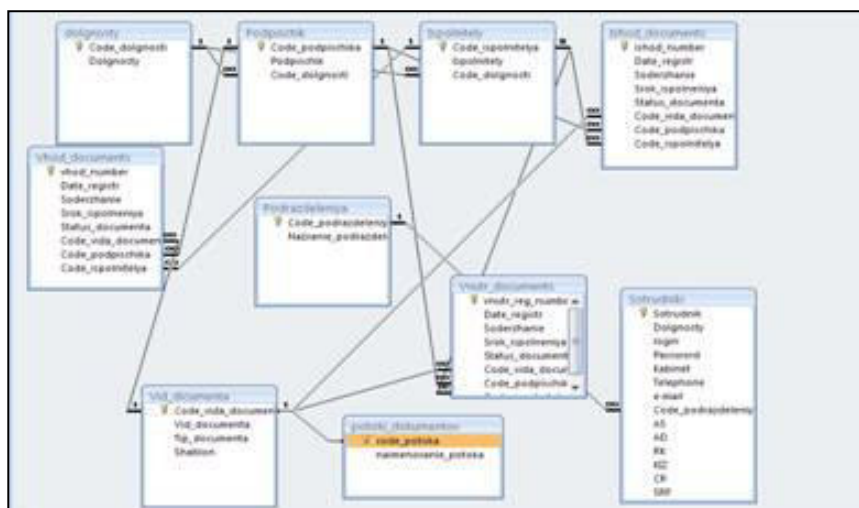
2.4-сурет – Менеджер үшін күй диаграммасы



2.5-сурет – Қолданушы үшін күй диаграммасы

2.9 ER диаграммасы

Деректерді модельдеудің кең таралған құралы "болмыс-байланыс" (ER-diagrams, ERD) диаграммасы болып табылады. Олардың көмегімен пәндік облысқа мағыналы объектілер (болмыстар), олардың қасиеті (атрибут) және бір-бірімен байланысы анықталады. ERD тікелей реляционды деректер базасын жобалау үшін қолданылады. ERD нотациясы алғаш рет П. Ченмен (Chen) 1976 жылы енгізілген, проектінің ER диаграммасы 2.6-суретте көрсетілген.



2.6-сурет – Проектінің ER диаграммасы

2.10 MySQL технологиясы

MySQL еркін қатынастарды басқару дерекқоры. MySQL бастапқыда MySQL AB-мен сатып алынған швед фирмасының Sun Microsystems-ға тиесілі шведтік бренді Oracle корпорациясы тарапынан әзірленіп, қызмет етуде. Өнім GNU General Public License, сондай-ақ өзінің коммерциялық лицензиясы бойынша таратылады. Бұдан басқа, әзірлеушілер лицензияланған пайдаланушылар үшін арнайы функцияларды жасайды. Осы бұйрықтың арқасында репликалау механизмі бірінші нұсқасында пайда болды.

MySQL-DBMS-дің икемділігі көптеген кесте түрлерін пайдалану арқылы беріледі: Пайдаланушылар толық мәтінді іздеуді қолдайтын MyISAM кестелерін және жеке транзакция деңгейіндегі транзакцияларды қолдайтын InnoDB үлгілерін таңдай алады. Сонымен қатар, MySQL DBMS пайдаланушы кестелер кестесін көрсетеді және кестелердің жаңа түрлерін жасау қағидаларын көрсетеді. Ашық сәулет және GPL лицензиялау арқылы MySQL әрдайым жаңа кесте түрлерінде пайда болады.

2008 жылдың 26 ақпанында Sun Microsystems MySQL AB-ді 1 млрд. Долларға сатып алды және Oracle 2010 жылдың 27 қаңтарында Sun Microsystems компаниясын 7,4 млрд. Долларға сатып алды және MySQL-ты өз DBMS желісіне шығарады.

MySQL әзірлеушілер тобы Drizzle, OurDelta, Percona Server және MariaDB секілді әртүрлі кодталған филиалдарды жасады. Барлық осы филиалдар Sunni сатып алу кезінде Oracle компаниясымен қол жетімді болды.

MySQL – бұл шағын және орта қосымшаларға арналған шешім. Серверлер WAMP, AppServ, LAMP және Denver, XAMPP, VertrigoServ қызметтерінің портативті ассамблеяларында орналасқан. Әдетте, MySQL жергілікті немесе қашықтағы сервер ретінде пайдаланылады, бірақ ішкі сервер кітапханасы бар, ол MySQL-қосымшаларға қосылған қосымшалар пакетін қамтиды.

2.11 HTML тілі туралы түсінік

HTML кодын және құжат мәтінін қамтитын HTML құжаты. HTML құжаттары мәтінді пішімдеу, құжат құрылымын жасау, жазбаларды қосу және мультимедиалық нысандарды жасау үшін пайдаланушы кілт сөздерін пайдаланады.

Негізінен, HTML тілі веб-беттерді құру үшін пайдаланылады. Бұл World Wide Web тұжырымдамасынан алыс емес. HTML5 және HTML 4 арасындағы айырмашылықтар 3.3 суретте көрсетілген.

HTML-нің дамуы туралы айтқан кезде, ол бір жерде болады деп айту мүмкін емес. WEB шебері сізде барлық жаңа HTML мүмкіндіктерін қамтамасыз ететін көптеген элементтері бар (ең соңғы HTML 4.01 техникалық сипаттамасы 1991 жылы шыққанымен).

HyperText белгілеу тілі «HyperText Markup Language» -ге аударылады. Бұл тіл веб-парақтарды құрудың негізгі тілі болып табылады және беттің әр түрлі бетін (мәтін, графика, кестелер және т.б.) нақты командалармен орналастыру үшін пайдаланылады. Оның қарапайымдылығына байланысты ол басқа бағдарламалау тілдерінен ерекшеленеді. Информатика негіздерін білмейтін қарапайым оқырман, ол бастапқыда жеке компьютер пайдаланушылары үшін жасалғандықтан, түсінуге және үйренуге болады. HTML арқылы қарапайым және күрделі HTML жасауыңызға болады. HTML тілі сканерлері мен графикасын HTML редакторлары (WEB беттерін көруге арналған арнайы бағдарлама) оқиды.

Дегенмен, барлық кедергілердің бірі браузерлер бір мезгілде бір элементті түсінуге болмайды және жетекші әзірлеушілер көптеген HTML редакторларын мәселені шешуге көмектесетін арнайы құралдарға айналдыруы керек.

2.12 YII2 тірегі

Yii үлкен веб-қосымшаларды жедел дамытуға арналған PHP-тің тиімділігі жоғары құрамдас бөлігі. Бұл кодты қайта пайдалануды максималды түрде пайдалануға мүмкіндік береді және веб-әзірлеуді едәуір жеделдетеді. Yii атауы (Yee немесе [ji:] - қарапайым (қарапайым), тиімді және кеңейтілетін дегенді білдіреді.

Yii туралы әңгіме 2008 жылғы 1 қаңтардан басталды. Zend - PHP 5 кодтау конкурсын (PHP) жеңіп алған PRADO (PHP) жобасындағы кемшіліктерді түзейтін жоба.

PRADO жүйесі ASP.NET FRS платформасына ViewState,PostBacks, Page_Load және OnClick қоса ASP.NET фрагменттерінің бөлімдерін көшіруге әрекет жасады. Мысалы, .NET логикалық болып табылады, бірақ PHP ішіндегі оңтайлы оңтайландырылған жад сақтау үшін Басқару класындағы Rare Fields және Occasional Fields ішіне бөлінеді. PRADO барлық ASP.NET теріс

аспектілерін иеленді: ол баяу күрделі беттерді өңдейді, қарапайым оқыту қисық сызығына ие және орнату қиын болды.

Бір сәтте PRADO негізін құрушысы және әзірлеушісі - Сян Сьюо PHP құрылымы басқаша болуы керек екенін түсінді. 2008 жылдың қазан айында алғашқы альфа нұсқасы 10 айдан бері жабық дамудан кейін шығарылды. 2008 жылғы 3 желтоқсандағы Yii 1.0 нұсқасы.

Басқа тіректермен салыстыру. Сынақ нәтижелері бойынша YPPmark Y өнімділігі жоғары болды.

Әділ болу үшін, осы сынақтардағы кадрлар «Hello World» сияқты жасанды мысалдармен бағаланды. Эксперименттерде жақтаудың басталу уақыты көрсетіледі және тек Yii кешіктіріп бастау функциясының сапасы бар екенін білдіреді (яғни, қажет болған жағдайда код жүктеледі). Yii «жауынгерлік жағдайларда» басқа жүйелерге қарағанда жылдам жұмыс істейтіні туралы ешқандай дәлел жоқ.

Блогтардағы көптеген мақалаларды талдай аласыз. Жалпы алғанда, мынадай үрдістер байқалады:

Yii белсенді дамып келеді. Кейбір жағдайларда, Zend Framework және Symfony салыстырғанда, жүйедегі оқытудың жоғары деңгейі, нәтижелер және прототиптер жазылады. Ол сондай-ақ оның тұрақтылығы мен қауіпсіздігін көрсетеді.

Yii2 құрамдас құрылымына және MVC парадигмасына (Model View Controller) негізделген жоғары сапалы нысанға бағдарланған PHP негізі. PHP жүйесі Yii2 дегеннің не екенін түсіну керек дегенді білу керек.

PHP тірегі – веб-сайттарды және веб-қосымшаларды әзірлеуге арналған негіздемелік, құрылымдық немесе арнайы құралдар жиынтығы. Мұнда айқынырақ ету үшін операцияның мысалы келтірілген. Мысалы, кейбір стандартты емес сайттарды жасауыңыз керек, себебі осы нақты тапсырма үшін ешкімнің CMS-і өзіңіздің жеке мазмұнды басқару жүйесін дамытуыңыз керек. Ол үшін сайт құрылымы туралы ойлап, бағыттаушыларды және түрлі валидаторларды жасауды, қауіпсіздік туралы ойлауды, дерекқормен жұмыс істеуді және т.б.

Барлық осы функцияларды әртүрлі тәсілдермен шешуге болады:

- барлығын нөлге жазыңыз;
- желіні табу және CMS дайын компоненттерін құру;
- PHP-ні қолдана отырып, қолдануға дайын компоненттер бар.

Қысқаша айтқанда, PHP CMS әзірлеу үшін уақытты қысқартуы мүмкін. Бұл жүйе бірнеше минут ішінде орнатылады және «қосылды» болады және бірден күнделікті жұмысты өткізуге көп уақыт жұмсайтын CMS жасау үшін көшуге болады.

Көптеген PHP құрылымдары бар, олардың әрқайсысында өздерінің үлестері мен келісімшарттары, оң және кемшіліктері бар. Барлық заманауи және танымал (Yii2 ерекше) объектілі-бағдарлы бағдарламалау стилінде жазылған және MVC парадигмасын іске асырады.

Yii2 – ең танымал және ең жылдам PHP жүйелерінің бірі, үйрену қиын

және көп функционалдылыққа ие:

– DAO және ActiveRecord дерекқорымен жұмыс істеу үшін пайдаланылады;

– деректер базасын көшіру және REST қолдау;

– GII – интеграцияланған коды генераторы;

– түрлі кәштеу параметрлері;

– үшінші жақ кітапханалары мен сыныптарды қарапайым интеграциялау;

– жоғары сапалы қауіпсіздік құралдары;

– әрекеттер, көмекшілер, негізгі функцияларды кеңейту мүмкіндіктері және т.б.

Үйі2 танымалдылығына байланысты әртүрлі кеңейтімдер мен модульдер түрлі тапсырмаларды шешу үшін жазылғанын атап өту керек, бұл, әрине, даму жылдамдығына әсер етеді.

GUIDE – оқытудың басында әсіресе пайдалы болатын сипаттамалық және кодтық үлгілері бар анықтамалық, бірақ ол сондай-ақ базалық базалық конвенцияларды қамтиды: модельдер, менеджерлер, турлар, модульдер, виджеттер және басқаларды түсінуге көмектеседі.

2.13 PHP программалау тілі

PHP – («Жеке веб-беттерді жасау құралдары») веб-қосымшаларды әзірлеуге кеңінен қолданылатын жалпы мақсаттағы сценарий тілі [9]. Қазіргі уақытта хостинг-провайдерлердің басым көпшілігі қолдайды және динамикалық веб-сайттарды жасау үшін қолданылатын тілдердің арасында көшбасшылардың бірі болып табылады [10].

Тіл және оның аудармашысы (Zend Engine) ашық көзі бар жобада энтузиастардың тобы әзірлейді. Жоба GNU GPL-пен үйлесімсіз өз лицензиясы бойынша таратылады.

Веб-бағдарламалау саласында, атап айтқанда серверлік, PHP - ең танымал сценарий тілдерінің бірі (JSP, Perl және ASP.NET-де пайдаланылатын тілдер). Рrри frontend Құрылыс веб-сайттарында танымалдылық веб-қосымшаларды әзірлеуге арналған қосымша құралдар мен қосымша модульдердің үлкен жиынтығымен анықталады. Ең бастысы:

– POST және GET параметрлерін автоматты түрде алу, сондай-ақ алдын ала анықталған массивтерде веб-сервердің қоршаған орта айнымалысы;

– қосымша модульдер (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, CloudScape және Apache Derby, Informix, Ovrimos SQL , Lotus Notes, DB ++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, PDO интерфейсі), Redis;

– HTTP тақырыптарын автоматты түрде жіберу;

– HTTP рұқсатымен жұмыс істеу;

– куки және сессиялармен жұмыс істеу;

- жергілікті және қашықтағы файлдармен, розеткалармен жұмыс істеу;
- серверге жүктелген файлдарды өңдеу;
- XForms-мен жұмыс істеу.

1994 жылы Дат программисі Расмус Лердорф C-де HTML-Құжаттардың үлгілерін өңдейтін оның онлайн-түйіндемесіне келушілерді есепке алу үшін CGI-скрипттер жиынтығын әзірледі. Lerdorf personal Home Page Tools жиынтығын атады (жеке басты бет құралдары). Көп ұзамай функционалдылық жеткілікті болды және Lerdorf PHP/FI үлгілерінің жаңа интерпретаторын әзірледі (ағылш. Personal Home Page / Forms Interpreter - "жеке басты бет / формалар интерпретаторы") [14].

8 маусым 1995 жыл, Personal Home Page (PHP Tools) version 1.0 — бірінші жария релиз.

1997 жылы ұзақ бета-тестілеуден кейін C — PHP/FI 2.0 жазылған өңдеушінің екінші нұсқасы шықты. Оны әлемнің барлық интернет-домендерінің 1 % (шамамен 50 мың) пайдаланды.

PHP 3.0 нұсқасы қазіргі заманғы бет әлпеті мен бағдарламалау тілінің стилін анықтайтын айтарлықтай өңдеуден өтті. 1997 жылы екі израильдік программист, Энди Гутманс және Зеев Сураски интерпретатор кодын толық жазып алды. PHP 3.0 ресми түрде 1998 жылдың маусымында шығарылды.[8]

PHP 3.0 мықты жақтарының бірі ядроны қосымша модульдермен кеңейту мүмкіндігі болды. Сонымен қатар, PHP-ге өз модульдерімен жұмыс істейтін көптеген үшінші тарап әзірлеушілері тартылған, бұл PHP-ге деректер қорының, хаттамалардың үлкен санымен жұмыс істеуге, API-дің үлкен санын қолдауға мүмкіндік берді. Көптеген әзірлеушілер тілдің жылдам дамуына және оның танымалдығының жылдам өсуіне әкелді. Осы нұсқадан PHP акронимі ескірген "Personal Home Page "орнына" PHP: hypertext Preprocessor " деп аталады.

1998 жылдың қысына қарай PHP 3.0 ресми шыққаннан кейін Энди Гутманс пен Сураски Зеев PHP ядросын қайта өңдеуді бастады. Міндеті кірді өнімділігін ұлғайту, күрделі қосымшаларды және жақсарту модульдерін PHP кодының базисінің Zend Engine аталған жаңа қозғалтқыш қойылған міндеттерді сәтті орындап, алғаш рет 1999 жылдың ортасында ұсынылды. Осы қозғалысқа негізделген және өзімен бірге қосымша функциялар жиынтығын әкелген PHP 4.0, ресми түрде 2000 жылдың мамырында шықты. Өнімділікті жақсартуға қосымша, PHP 4.0 сессияларды қолдау, шығаруды буферлендіру, пайдаланушы енгізетін ақпаратты өңдеудің қауіпсіз тәсілдері және бірнеше жаңа тіл конструкциялары сияқты бірнеше негізгі жаңалықтар болды.

PHP бесінші нұсқасын әзірлеушілер 2004 жылдың 13 шілдесінде шығарды. Өзгерістер Zend (Zend Engine 2) ядросын жаңартуды қамтиды, бұл интерпретатордың тиімділігін айтарлықтай арттырады. XML белгілеу тілін қолдау енгізілді. Java-да қолданылатын модельге ұқсас болған ООП функциялары толығымен қайта өңделген. Атап айтқанда, деструктор, ашық, жабық және қорғалған мүшелер мен әдістер, соңғы мүшелер мен әдістер, интерфейстер мен нысандарды клондау енгізілді. Келесі нұсқаларда сондай-ақ аттардың кеңістіктері, тұйықталу және PHP 5.0 көшу кезінде пайда болған

сандық және сапалық жағынан салыстырылатын айтарлықтай елеулі өзгерістер қатары енгізілген.

PHP-тің алтыншы нұсқасы 2006 жылдың қазанынан бастап жасалды. Мысалы, POSIX тұрақты өрнектері мен "ұзын" суперглобальды массивтердің ядросынан алып тастау, PHP конфигурациялық файлынан `safe_mode`, `magic_quotes_gpc` және `register_globals` директиваларын жою сияқты көптеген жаңалықтар жасалды.ini. Негізгі жаңалықтардың бірі жас жігіттің қолдауы болуы керек еді. Алайда, 2010 жылдың наурыз айында PHP6 әзірлемесі Юникодтың қолдауымен қиындықтарға байланысты перспективасыз деп танылды. PHP6 бастапқы коды тармақтарға ауыстырылды, ал әзірлеудің негізгі желісі 5.4 нұсқа болды.

3. Жобаны жүзеге асыру және тестілеу бөлімі

3.1 Жүйенің архитектурасы

Электрондық құжаттарды басқару жүйесі архитектурасы бірнеше модульден тұрады: басқару модулі, дисплей модулі, кіру модулі және аутентификация және аутентификация модулі. 3-суретте электронды құжаттарды басқару жүйесі компоненттік схема ретінде сипатталады. Әрбір құрамдас бөлек бағдарламалау модулі және ішкі бөлігі ішкі модуль болып табылады, ол негізгі модульдің бөлігі болып табылады.

Басқару модулі жүйенің пайдаланушылары мен пайдаланушыларымен жұмыс істеу логикасын ұйымдастыруға арналған бірнеше модульден тұратын негізгі модуль. Дисплей модулі дерекқорда сақталған ақпарат үшін пайдаланушының браузерінде тікелей көрсетілуі қажет. Деректерге қатынау модулі дерекқорды басқару мүмкіндіктерін қамтиды: дерекқордан деректерді алу, дерекқордағы деректерді жаңарту және сақтау. Сәйкестендіру, түпнұсқалық растама және авторизациялау модулі пайдаланушыларға жүйеге кіруге және пайдаланушылардың кіруіне кіруге арналған.

3.2 Пайдаланылған бағдарламалық жасақтамалар

3.2.1 PhpStorm ортасы туралы

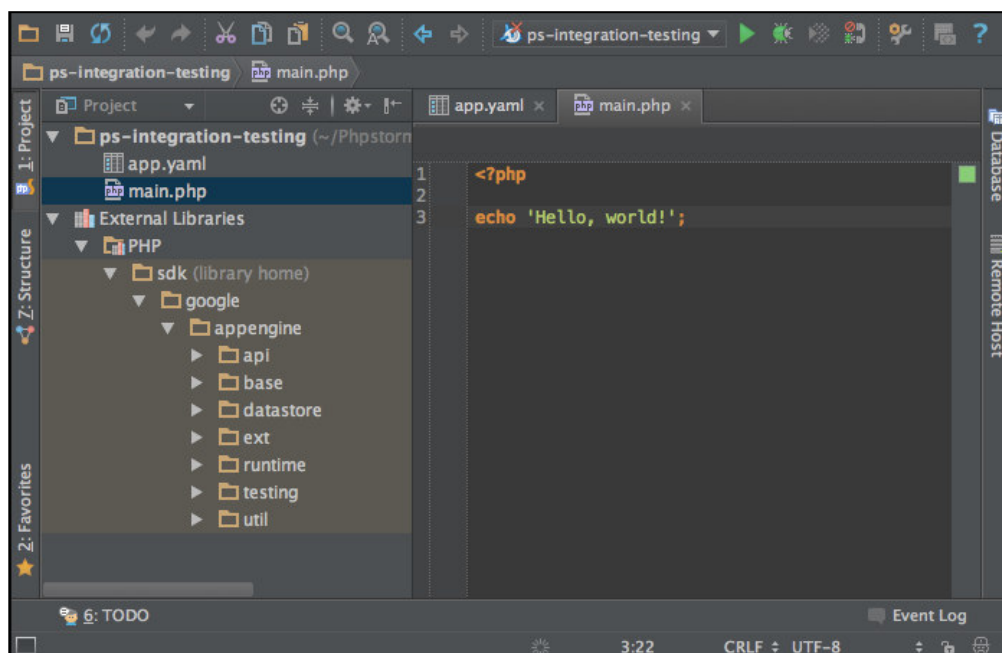
JetBrains PhpStorm – PHP үшін коммерциялық, кросс-платформалық және оңайлатылған сценарийлік ортасы. JetBrains компаниясымен дамытылады, және IntelliJ IDEA кода жазу ортасы іргетасы болып табылатын құрал.

PhpStorm құралы PHP, HTML, JavaScript арналған интеллектуалды редактор болып табылады. Оның құрамында кодты қазіргі жағдайда зерттеу мүмкіндігі, PHP JavaScript бағдарламалу тілдеріне арналған рефакторинг құралдары арқылы қателердің алдын алса болады. PhpStorm құралының жұмыс орны 4.1-суретте көрсетілген.

PHPStorm PHP, JavaScript және PHP үшін ақылды редактор, PHP және JavaScript үшін кодтық талдау, код қателіктері және автоматтандырылған қайта өңдеу. PHPStormалық кодты автотолтыру PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1 және 7.2 спецификацияларын (қазіргі және дәстүрлі жобаларды), генераторлардың, каталогтарды, тақырыптарды, желімдерді, қысқа каталогтарды синтаксисі мен пішімдеуін қолдайды. Толық SQL редакторы алынған сұраудың нәтижелерін өңдеу үшін қол жетімді.

PhpStorm платформасы IntelliJ IDEA, Java-ға арналған. Мақсатты жұмыс істеуге арналған функционалды мүмкіндіктері бар, ол IntelliJ платформасы үшін қол жетімді.

PHPStorm PHP үшін бай және интеллектуалды код редакторын қамтамасыз етеді, оған код жарықтауын, кеңейтілген кодты пішімдеуді, жылдам қателерді тексеруді және ақылды автотолтыру кіреді.



3.1-сурет – PhpStorm ортасының жұмыс орны

PhpStorm құралында PHP 5.3, PHP 5.4, PHP 5.5, PHP 5.6, PHP 7.0 және PHP 7.1 тілдеріне арналған өздігінен кодты толықтырғыш орналасқан. Осыған қоса, құралда генераторлар, атаулар кеңістігі, типаждар және қысқа массивтер синтаксистері бар. Құралда SQL сұраныс тілінің толық түрдегі редакторы мен қоса алған жауаптың өңдеу мүмкіндігі бар.

PhpStorm платформасы IntelliJ IDEA платформасының арқасында құрылған, платформа Java бағдарламалу тілінде жазылған. Платформаның мүмкіндіктерін алдын ала IntelliJ IDEA платформасына дайындалған плагиндер арқылы кеңейтуге болады, немесе, сондай плагиндерді арнайы жазса болады. WebStorm платформасының барлық мүмкіншіліктері PhpStorm платформасында да бар. WebStorm құралы JavaScript бағдарламалау тіліне арналған код жазу ортасы.

PhpStorm платформасының кілттік мүмкіншіліктері:

- интеллектуалды синтаксисті жарықтандырып тұратын PHP кодының редакторы;
- PHP 5.3, PHP 5.4, PHP 5.5, PHP 5.6, PHP 7.0 және PHP 7.1 тілдеріне арналған өздігінен кодты толықтырғыш орналасқан;
- PHP бағдарламалу тілінің алдын ала орнатылған рефакторингі;
- Vagrant, Composer қолдаулары, және, REST клиент, командалық жолақ, SSH консоль;
- қазіргі кездегі үздік тіректерге жақсы қолдау көрсетілген;
- визуалды түрде PHP қосымшаларын өңдеу;
- HTML, CSS, JavaScript редакторы. HTML5, CSS, Sass, SCSS, Less, Stylus, Compass, CoffeeScript, TypeScript, ECMAScript Harmony, Emmet қолдау

көрсетілген;

- фронтендке арналған құралдардың барлығы бар;
- алдын ала орнатылған PSR1/PSR2, Symfony2, Zend, Drupal стилдері;
- нұсқаларды басқару жүйелерімен жүретін интеграция;
- қашықтан басқару жүйесі;
- Live Edit, өзгерткен кодты тікелей браузерден сол сәтте көре алу мүмкіндігі;
- баг-трекерлермен интеграция жасалынған;
- деректер қорымен жұмыс жасауға арналған құралдар;
- кросс-платформалық мүмкіндігі бар, яғни, кез-келген операциялық жүйеде жұмыс жасай береді.

3.2.2 Brackets редакторы

Brackets – веб-әзірлеушілер үшін тегін мәтіндік редактор. Brackets HTML, CSS және JavaScript-мен жұмыс істеуге бағытталған. Сол технология – бұл кросс-платфонды қамтамасыз ететін редактордың өзі. Mac, Windows және Linux операциялық жүйелерімен үйлесімді. Кронштейндер MIT лицензиясы бойынша Adobe Systems компаниясы құрған және әзірленді және GitHub-да қолдау тапты.

Бүгінгі күні қауымдастық Git нұсқасын басқару жүйесі сияқты HTML-кодты нақты уақыттағы браузерде (Live Preview) қарау және FTP (Git-FTP) - мен синхрондау сияқты кодта жұмыс істеу үшін қажетті құралдардың көбін қосатын көптеген кеңейтімдерді жасады. Кез келген адам кеңейтулерді дамытуға және қолдау көрсетуге қатыса алады.

Біріншіден, 2014 жылдың қараша айында тоқтатылған Edge Code-де веб-әзірлеу үшін мәтіндік редакторды әзірлеу басталды. Кейінірек Adobe Fridge-ке ауыстырылды. 1.0 кронштейндері шығарылған кезде, Adobe веб-әзірлеуге ашық бастапқы бағдарламалық жасақтама әзірленіп, енді эксперименттік жоба болмады деп жариялады. Қауымдастықтың 282-ден астам мүшелері және қателерді түзету және жаңа мүмкіндіктер үшін 400-ден астам сұраулар жақшаларда орналасқан. Краштардың әрбір нұсқасы 100 000-нан астам жүктеуден тұрады және 2015 жылғы 16 қаңтардағы жағдай бойынша GitHub бойынша 16-шы ең танымал жоба болып табылады.

GitHub (кронштейн репозиторийі) арналған кронштейндердің репозиторийі бүгінгі күні 2018 жылғы 30 тамыздағы жағдай бойынша 152 филиалы, 110 шығарылымы және 17 700 АҚШ доллары бар. Бастапқы код MIT лицензиясы бойынша еркін қол жетімді. Әзірлеуші функцияларды жақшада өзгертіп, бағдарлама кодын жасайтын өздерінің ыңғайлылығы үшін жекелендіре алады.

Жылдам редакциялау әзірлеушілерге CSS, Color Property және JavaScript элементтерін кірістірілген редакциялауға мүмкіндік береді. Бұл кірістірілген функция бірнеше функцияларға немесе қасиеттерге бір мезгілде қолданылуы

мүмкін және барлық жаңартулар өзгертілген элементтермен байланысты файлға тікелей қолданылады.

Html файлы. HTML элементтеріне жылдам өңдеуді қолдану таңдалған элементтің төменгі жағындағы барлық тиісті CSS сипаттарын көрсетеді. Пайдаланушылар редакторда тікелей жаңа CSS ережелерін жасай алады және CSS-тегінің қасиеттерін HTML файлының мәтінмәнінен шықпай өңдей алады.

Javascript файлы. JavaScript функциялары үшін жылдам өңдеулер HTML элементтері үшін қолданылатын процедураға сәйкес келеді, бірақ ашылмалы тізімдегі таңдалған функцияны көрсетеді. Функцияның барлық жаңартулары тиісті JavaScript файлында тікелей таратылады және жаңартылады.

Hex немесе RGB түсті сипаттары бар файлдар. Түс сипаттары үшін жылдам өңдеу алдын-ала қарау және түс реттеу үшін кіріктірілме түсті бояғышты қайтарады.

Brackets C ++, C, VBScript-ден Java, JavaScript, HTML, Python, Perl және Ruby-ден бірнеше файл түрлерінің кодтарын қолдайды. Толық тізім 38 файл түрін қамтиды. Бұл пайдаланушыға әртүрлі жоба файлдарымен бір уақытта жұмыс істеудің икемділігін береді.

Brackets «PSD линзасы» функциясын қолдайды, бұл суреттерді, логотиптерді және дизайн стилдерін Photoshop-ты ашпай-ақ PSD файлынан тегіс алуға мүмкіндік береді. Бұл алдын-ала қарау мүмкіндігіне қоңырау шалып, Adobe осы мүмкіндікті жетілдірмей тұрып әлі де көп жұмыс істеп тұрғанын хабарлайды. Бұл мүмкіндік әзірлеушілерден оң нәтиже алды, бірақ көптеген проблемалар осы мүмкіндікті шығарудың бастапқы кезеңінде көрсетілді. Кейінірек мәселе кеңейтумен шешілді.

3.2.3 HTML5 және CSS3 негіздері

HTML (HyperText Markup Language) - Әмбебап веб-сайтындағы құжаттар үшін стандартталған пішімдеу тілі. Көптеген веб-беттерде HTML (немесе XHTML) таңбалары бар. HTML браузерлері интерпретациялаған; компьютер немесе мобильді құрылғы мониторын түсіндіру нәтижесінде пайда болатын пішімделген мәтін.

HTML5 - бүкіл әлемдегі мазмұнды жасау және жеткізу тілі. Бұл HTML нұсқасының бесінші нұсқасы. 2014 (алдыңғы, төртінші нұсқасы 1999 жылы шыққан) 2013 жылдан бастап браузерлерге қолдау көрсетілді және әзірлеушілер жұмыс станциясын пайдаланады (көпшілігі HTML Living Standard). HTML5-нің мақсаты мультимедиялық технологияларды қолдау болып табылады, бұл кері сәйкестік, код үшін оқуға қабілеттілік және талдаушылар үшін талдаудың қарапайымдылығы.

World Wide Web бұрыннан HTML 4.01, XHTML 1.0 және XHTML 1.1 стандарттарын қолданып келді. Іс жүзінде, веб-беттер әртүрлі спецификациялармен, соның ішінде веб-браузерлер мен енгізілген қолданбалар сияқты бағдарламалық жасақтама өнімінің техникалық сипаттамаларымен қамтамасыз етіледі. HTML5 HTML және XHTML синтаксисін біріктіретін

жалғыз пішімдеу тілі ретінде жасалған. Құжаттың орналасуын ұзартады, жақсартады және оңтайландырады және күрделі веб-қосымшалар үшін бірыңғай API қосады.

5-ші нұсқаға дейін HTML SGML (ISO 8879 стандартты жалпы пішімдеу тілі) ретінде сипатталады. HTML5 спецификациясы DOM (Document Object Model) үшін жасалды.

XHTML - XML синтаксисін қолданатын және гипермәтінді белгілеу өрісінде XML бағдарламасы болып табылатын HTML-нің тұрақты нұсқасы.

Әмбебап веб-браузердегі HTML әдетте HTTP немесе HTTPS протоколдарын пайдаланатын браузерлерді, мәтіндерді немесе кодторларды пайдаланады.

HTML гипермәтінді белгілеу тілін 1986 жылдан 1991 жылға дейін Женевада CERN-да ағылшын ғалымы Тим Бернерс-Ли әзірледі. HTML HTML-пішімде ғылыми және техникалық құжаттарды ортақ пайдалану үшін құрылды, оны кәсіби емес адамдар пайдалана алады. HTML мазмұны мен семантикалық элементтерін анықтау арқылы SGML күрделілігін сәтті шешті. Идентификаторлар жиі «белгілер» деп аталады. HTML арқылы қарапайым, бірақ әдемі түрде жасалған құжатты оңай жасауға болады. Құжаттың орналасуын жеңілдетуден басқа, HTML гипермедиаға қосылды. Мультимедиа мүмкіндіктері кейінірек қосылды.

Бастапқыда HTML тілі құжаттарды құру және пішімдеу құралы ретінде жасалды және оларды репродукциялармен бөліспеді. HTML пішіміндегі техникалық құралдарын түрлі мәтін (қазіргі заманғы ұйымдастырылған түсті дисплей, компьютердің монохромды экран, шектеулі экран өлшемі ұялы телефон немесе құрылғы және мәтінмен оларды жарақтандыру, аудио Бағдарламаны ойнату) идеалды тәсілі кез келген стилистикалық немесе құрылымдық buzilishlarsiz жылы берілуге тиіс. Дегенмен, HTML-дің ағымдағы қолданысы бастапқы мақсатқа жетпейді. Мысалы, <table> тегі құжаттардағы кестелерді жасауға арналған, бірақ кейде ол сондай-ақ беттің элементтерінің орналасуын жобалау үшін пайдаланылуы мүмкін. Уақыт өте келе, HTML платформасының платформасының тәуелсіздік идеясы мультимедиялық және графикалық дизайндағы заманауи мұқтаждықтар үшін құрбан болды.

HTML пішімделген мәтіндік құжаттар (мысалы, .html немесе .htm кеңейтімі бар құжаттар қалыпты түрде) құжатты пішімделген форматта көрсететін арнайы бағдарламалармен өңделеді. «Браузерлер» немесе «Интернеттегі браузерлер» сияқты браузерлер әдетте пайдаланушыға веб-парақтарды сұрау және шолу (сыртқы құрылғыларға кіру) үшін қажетті интерфейспен қамтамасыз етеді және қажет болған жағдайда, серверге деректерді жібереді. Бүгінгі таңда ең танымал браузерлер Google Chrome, Mozilla Firefox, Opera, Internet Explorer және Safari (Browser # Market Share).

Ресми HTML 1.0 спецификациясы жоқ. 1995 жылға дейін көптеген бейресми HTML стандарттары болды. Олардан айырмашылығы, оған бірден екінші нөмір беріледі.

Версия 3 1995 жылғы наурызда World Wide Web консорциумы (W3C) арқылы енгізілді және кестелерді жасау, мәтінді орналастыру, күрделі математикалық формулаларды көрсету және gif пішімдерін қолдау сияқты көптеген жаңа мүмкіндіктерді ұсынды. Бұл стандарт екінші нұсқамен сәйкес келсе де, браузерлердің жұмыс істеуі қиынға соғады. Нұсқа 3.1 ешқашан ресми түрде шығарылмайды және HTML нұсқасының келесі нұсқасы 3.2 және 3.0 нұсқасының жаңартулары қабылданбады, бірақ Netscape Navigator және Mosaic сканерлері қолдайтын стандартты емес элементтер қосылды.

HTML 4.0 нұсқасында стандартты «тазарту» бар. Көптеген элементтер ескірген және ұсынылмаған (ағылшын ескірген) деп белгіленген. Атап айтқанда, қаріп параметрін өзгерту үшін пайдаланылатын қаріп тегі ескірген (біз оның орнына CSS стиліндегі кестелерді пайдалануды ұсынамыз).

1998 жылы World Wide Web консорциумы жаңа пішімдеу тіліне негізделген HTML4-де жұмыс істей бастады, бірақ XML синтаксисіне сәйкес келеді. Кейіннен жаңа тіл XHTML деп өзгертілді. XHTML 1.0 нұсқасының бірінші нұсқасын World Wide Web консорциумы 2000 жылдың 26 қаңтарында ұсынды.

XHTML 2.0 жыл соңында 2009 жылдың аяғында HTML және XHTMLning үзіліс алдыңғы нұсқаларында сәйкес шығару үшін жоспарланған, бірақ 2 шілде, 2009, World Wide Web Consortium XHTML2 жұмыс тобының өкілеттік мерзімі болып табылады. Нәтижесінде XHTML 2.0 стандартының барлық кейінгі дамуы тоқтатылды.

World Wide Web консорциумы енді HTML5 шығарады. 2007 жылдың 20 қарашасында Интернетте тіл ерекшелігі жобасы пайда болды.

2004 жылдан бастап WHATWG (Web Hypertext Application Technology Working Group) HTML-ды кеңейтетін (бірақ XHTML 1.0 үйлесімді XML синтаксисі) HTML5 деп аталатын веб-1.0 ерекшелігін әзірлейді. Форумдар, аукциондық сайттар, іздеу жүйелері, онлайн-дүкен және т.б. сияқты әдеттегі беттердің семантикасын жақсырақ көрсету үшін, XHTML 2.0-ке сәйкес келмейді.

HTML - тегтелген таңба тілінің құжаттары. Кез-келген HTML құжаты - әрбір элементтің басы мен соңы, арнайы элементтермен көрсетілген элементтер жиынтығы. Деректер бос болуы мүмкін, яғни мәтін немесе басқа деректер жоқ. Бұл жағдайда жабу тегі әдетте анықталмаған (мысалы, «Пайдаланушы» жолы біреуі және жабу қажет емес). Сондай-ақ, элементтердің кейбір элементтерін (мысалы, href = атрибуция атрибуты) анықтайтын атрибуттар болуы мүмкін. Тестелер ашылатын қойындыда көрсетіледі

HTML5 көптеген жаңа синтаксистік функцияларды орындайды. Мысалы, сіз <video>, <audio> және <canvas> элементтерін, сондай-ақ SVG және математикалық формулаларды пайдалана аласыз. Бұл жаңартулар үшінші тарап API және плагиндер үшін желілік графикалық және мультимедиялық құрылғыларды жасауға және басқаруға мүмкіндік береді. <Section>, <article>, <header> және <nav> сияқты жаңа элементтер құжаттың семантикалық мазмұнын байытуға арналған (бет). Бірнеше элементтер мен атрибуттар

алынып тасталғанымен, сол мақсаттар үшін жаңа мүмкіндіктер енгізілді. <a> <menu> және <cite> сияқты кейбір нәрселер өзгертілген, қайта анықталған немесе стандартталған. API және DOM HTML5 спецификациясының негізгі бөлігіне айналды. HTML5 сондай-ақ теру қателерінің кейбірін анықтайды, сондықтан қателерді талдау барлық үйлесімді браузерлерге бірдей қолданылады.

2004 жылы WHATWG World Wide Web консорциумының (W3C) XHTML 2.0-нің келешектегі дамуына назар аударды және 2000 жылдан бері өзгермеген жаңа стандарт үшін HTML 4.01-ні іске қосты [8]. 2009 жылы W3C компаниясы өзінің XHTML 2.0 жұмыс топтарының мерзімі өткенін жариялады және оны жаңартпау туралы шешім қабылдады. Кейін W3C және WHATWG HTML5-ді бөлісті.

Бірнеше жыл бойы HTML5 веб-әзірлеушілері арасында танымал болса да, 2010 жылдың сәуірінде бұқаралық ақпарат құралдарының басты назарында болды. Осыдан кейін Apple Inc. компанияның басшысы. Стив Джобс жазған [10] [11] [12] [13] және HTML5-нің дамуымен аяқтаған «Flash for Ideas» атты мақаланы жазды, ол Adobe бағдарламасының басқа түрлерін көруге немесе пайдалануға қажеті жоқ. Flash [14]. Осыған байланысты веб-әзірлеушілер талқылаған, бірақ HTML5 жақсартылған функционалдылықты ұсынады, бірақ әзірлеушілер браузердің айырмашылықтарын және стандарттардың әртүрлі бөліктерін, сондай-ақ HTML5 және Flash арасындағы функционалдылықты сақтауы керек. айырмашылықтарды ескеру керек.

CSS (/ si:'esəs / ағылшын стилі парақтары - каскадты стильдер кестесі) - құжаттың пішімдеу тілінің көмегімен жазылғанын сипаттайтын ресми тіл.

Бұл HTML және XHTML пішімдеу тілдері арқылы жазылған веб-беттерге арналған SVG немесе XUL сияқты кез келген XML құжатына қолданылуы мүмкін. CSS-ның веб-бетін жасаушылар түстер, қаріптер, жеке блоктар және осы веб-беттердің пайда болуының басқа аспектілерін пайдаланады. CSS-ді дамытудың басты мақсаты - веб-парақ көрінісінен (ресми CSS шығарған) логикалық негізді (HTML немесе басқа пішімдеу тілін пайдалану) ажырату. Бұл бөлу құжаттың қолжетімділігін арттыруға, икемділікті қамтамасыз етуге және оның көрсетілуін бақылауға, мазмұнды күрделілігі мен қайталануын азайтуға мүмкіндік береді. Сонымен қатар, CSS сізге скриншоттар, басылған презентация, дауысты оқу (арнайы дауыс браузері немесе экраннан оқу құралымен) немесе Брайльді пайдаланатын құрылғылар ұсынылған кезде әртүрлі стильдер мен нәтижелерді зерттеуге мүмкіндік береді.

CSS ресми CSS-де жазылған. Ережелер веб-құжатта, өздері ұсынатын пішінде және CSS форматындағы сыртқы файлдарда болуы мүмкін. CSS пішімі - CSS ережелері мен түсініктемелері бар мәтіндік файл.

CSS стилдерін төрт тәсілмен сипатталған веб-құжатқа біріктіруге немесе енгізуге болады:

– егер стиль сипаттамасы бөлек файлда болса, оны <head> элементінде қамтылған <link> элементі арқылы құжатқа қосуға болады:

```
<!DOCTYPE html>
```

```

<html>
<head>
.....
<link rel="stylesheet" href="style.css">
</head>
<body>
.....
</body>
</html>

```

– Стилль файлы ата-аналардан бөлек болғанда, ол <style> элементіндегі @import өрнегімен құжатпен байланыстырылуы мүмкін:

```

<!DOCTYPE html>
<html>
<head>
.....
<style media="all">
@import url(style.css);
</style>
</head>
</html>

```

– стилльдер құжат ішінде сипатталған кезде <head> элементіне кіретін <style> элементіне қосылуы мүмкін:

```

<!DOCTYPE html>
<html>
<head>
.....
<style>
body {
color: red;
}
</style>
</head>
<body>
.....
</body>
</html>

```

– стилльдер құжаттың корпусында сипатталған кезде, олар бір элементтің атрибуттарына орналастырылуы мүмкін

```

<!DOCTYPE>
<html>
  <head>
    .....
  </head>
  <body>
    <p style="font-size: 20px; color: green; font-family: arial,
helvetica, sans-serif">
    .....
  </p>
</body>
</html>

```

3.2.4 JQuery технологиясы

jQuery JavaScript және HTML-ні байланыстыру үшін javascript жасайды. JQuery кітапханасы кез-келген DOM элементіне қол жеткізуге және DOM элементтерінің атрибуттары мен мазмұнына қатынасуға мүмкіндік береді. JQuery кітапханасы сондай-ақ ыңғайлы AJAX-қосылған API қамтамасыз етеді. JQuery - еріктілердің ерікті доноры.

Біріншіден, Джон Резиг өзінің алғашқы бағдарламаларын бастағанда HTML-дан үйренді. Резигтің достарының бірі оны веб-сайтқа (Angelfire арқылы) және QBasic-дің HTML негіздеріне бағдарламалаған. Оның әкесі Рождество үшін екі Рождестволық кітапты берді. Бастапқыда ол Visual Basic бағдарламасында HTML және Веб-дизайнға қызығушылық танытқан бағдарламаны бастады.

Дегенмен JavaScript үшін құмарлық 2004 жылы аяқталды. Кейінірек Резиг информатика бойынша диплом алды және Brand Logic жергілікті брендте жұмыс істеді. Пайдаланушы жасаған сайтты жобалаумен айналысады. Джон басқа әзірлеушілерден кодты пайдаланғандықтан, ол JavaScript-ті терең зерделеуді шешкендіктен, көңілінен шығып, қайғылы болды. Оқуды бастаған кезімде JavaScript-ті көптеген қиындықтарды шешуге өте күшті болатын қарапайым, бірақ ақылға қонымды тіл болатыны туралы қорытындыға келдім. Келесі екі жыл ішінде jQuery аяқталмақ болғанға дейін Джон Java қосымшаларын жасады. JQuery Rezig-ті құрудың басты мақсаты Javascript және кодтық кодты жеңілдету болды, ол браузердің проблемалары туралы алаңдамас үшін оларды пайдаланады. Кітапхана 2006 жылы Нью-Йорктегі BarCamp компьютерлік конференциясында жұртшылыққа ұсынылды.

Мүмкіндіктері:

– CSS-тің таңдаулыларының кейбірі «Sizzle» шолғышының қозғалтқышы болып табылады [4];

– DOM ағаштан, атап айтқанда, XPath-ді қондырма ретінде қолдайды;

– оқиғалар;

– көрнекі әсерлер;

– AJAX плагиндері;

– Javascript плагині

CSS-де jQuery хатты HTML құрылымынан бөледі, себебі ол HTML құрылымынан рендерді бөледі. Мысалы, тінтуір оқиғасына тікелей нұсқаудың орнына басқару элементі JQuery-ге өтеді, ол кілттерді анықтайды, содан кейін оны оқиғалар өңдегіштерін басу үшін ауыстырады. Бұл мінез-құлық пен құрылымдардың бөлінуі javascript javascript деп аталады.

JQuery кітапханасында көптеген тапсырмалар үшін пайдалы функциялар бар. Алайда, кітапхананы жасаушылар jQuery-дің барлық функцияларын біріктіруге тура келмеді, себебі ол үлкен кодты қажет етеді және олардың көпшілігі талап етілмейді. Осылайша, ықшам әмбебап кітапхана ядросы және қосылатын модульдер архитектурасы енгізілді [5]. Бұл ресурс үшін жиналған JavaScript функциясын жинауға мүмкіндік береді.

jQuery әдетте веб-параққа сыртқы JavaScript файлы ретінде қосылады:

```
<head>
  <script src="jquery-2.2.2.min.js">
</script>
</head>
```

Барлық jQuery-де жұмыс істеу \$ функциясы арқылы жасалады. Егер сіз сайтта JavaScript-ті қолданатын басқа кітапханаларды қолдансаңыз, сізге қажетінше \$ жұмсауға болатын болса, junk-jQuery функциясын пайдалана аласыз. Екінші әдіс дұрыс және код өте тым ауыр емес, сондықтан төмендегілерді жаза аласыз:

```
jQuery(function($) {
  // мұнда jQuery функционалдығын қамтамасыз ететін нысан бар сценарий коды
})
```

jQuery-мен жұмыс істеу 2 түрге бөлінеді:

– \$ () функциясын пайдаланып jQuery нысанын алу. Мысалы, CSS селекторына ауыса отырып, jQuery нысанын jQuery нысанының әртүрлі әдістерін қолдана отырып, барлық өлшемдерге сәйкес келетін барлық элементтердің jQuery нысанына қол жеткізе аласыз. Егер әдіс қандай да бір мәнді қайтармауы керек болса, ол jQuery нысанына сілтеме береді, ол ағындық интерфейсте қоңырау әдісінің тізбегін сақтауға мүмкіндік береді.

– объектінің жаһандық әдісін шақыру, мысалы, массивте ыңғайлы итератор болып табылады.

Бір уақытта бірден көп DOM түйінін басқарудың әдеттегі үлгісі CSS функциясын элементтер мен HTML функция элементтері бар jQuery нысаны арқылы қайтарылған таңдалған жолмен CSS деп атау. Бұл элементтер jQuery әдістерімен өңделеді. Мысалы:

```
$("#div.test").add("p.quote").addClass("blue").slideDown("slow");
```

Барлық div элементтері, сынақ сыныбымен бірге барлық элементтерді шақыру арқылы табады, содан кейін барлық көк сыныптар оларға қосылады, Мұнда add, addClass және slideDown әдістерін бастапқы нысанға \$ («div.test») қайтарады, осылайша сіз бұл тізбені сақтай аласыз.

2008 жылғы 28 қыркүйекте ресми jQuery блогы [6] Microsoft корпорациясымен және Nokia Development Team компаниясымен ынтымақтасуға ниет білдірді. Microsoft JQuery және jQuery мысалдарын ASP.NET өніміңізге біріктіруді жоспарлайды және Nokia ұялы құрылғыларына арналған jQuery-ті біріктіреді.

jQuery-JavaScript тілінің дайын функцияларын қамтитын JavaScript кітапханасы, jQuery барлық операциялары JavaScript кодынан орындалады. JQuery кітапханасы html элементтерімен манипуляциялар жасайды, олардың мінез-құлқын және Веб-беттің құрылымын өзгерту үшін DOM пайдалана

отырып. Бұл ретте HTML және CSS бастапқы файлдары өзгермейді, өзгерістер тек пайдаланушы үшін бетті көрсетуге енгізіледі.

Элементтерді таңдау үшін CSS селекторлары қолданылады. Таңдау `$()` функциясы арқылы жүзеге асырылады. Қоңырауда `$()` функциясы нөл немесе одан көп DOM элементтерін айналдыратын JQuery нысанының жаңа данасын қайтарады және олармен әртүрлі тәсілдермен әрекеттесуге мүмкіндік береді. Әр түрлі сценарийлер `document` құжат құрылымын жүктегеннен кейін, браузер беттің HTML кодын DOM ағашына айналдырғанда ғана мүмкін болады. Жүктеу процесін басқару конструкцияны қамтамасыз етеді.

```
jQuery(document).ready(function() {  
    ...  
});
```

Алдымен `document` данасын `jQuery()` функциясына орайды, кейін `function()` {функциясы берілетін `ready()` әдісі қолданылады...} құжат жүктелгеннен кейін орындалатын. Іс жүзінде әдетте `jQuery(function() {жазудың қысқартылған түрі қолданылады...});`, немесе `$(function() {...});`. JQuery кітапханасымен жұмыс кезінде ақпаратты сақтау үшін JavaScript айнымалылары қолданылады. Айнымалы элементтер сақталуы мүмкін. Қайтарылатын элементтерді сақтауға арналған айнымалы атаулар `$` белгісінен басталады, мысалы:

```
$h = $(".list").parent().parent().detach();
```

Бірнеше элементтерді сақтау үшін JavaScript массивтері қолданылады:

```
$k[3] = 15;
```

JQuery кітапханасын веб-бетке екі жолмен қосуға болады:

JQuery файлының нұсқасын Google, Microsoft немесе jQuery.com ескерту. Бұл әдіс "контентті дистрибуциялау желісі" (CDN, content distribution network) қолданады, яғни JQuery файлы басқа веб-сайтта орналасқан, ол пайдаланушы осы файлды оның компьютеріне жіберетін. Бұл тәсілдің айқын артықшылығы — өз веб-серверіне жүктемені төмендету және дистрибутор серверлері желісінің тармақталуына байланысты файлды жүктеуді жеделдету.

JQuery кітапханасының соңғы нұсқасын ресми сайтынан жүктеп алыңыз. Ол үшін сізге мекен-жайға өту қажет jQuery.com және сізді қызықтыратын кітапхана нұсқасын таңдаңыз. Жүктеу үшін JQuery-файлдың екі нұсқасы ұсынылады — ең аз және сығылмаған. Сығылмаған файлдың көлемі 300 Кб-ге жуық, ол түсініктеме береді, сондықтан кодты өңдеу және баптау мақсатында оны пайдалану жақсы. Файлдың ең аз нұсқасы 100Кб-ға жуық салмақ, онда барлық пікірлер мен қажетсіз Бос орындар жойылған, бұл файлды браузермен жүктеуді тездетеді.

Қалқымалы терезеде жүктеп алу үшін және сілтемені тінтуірдің оң жағымен нұқыңыз және «Басқаша сақтау ...» пәрменін таңдаңыз. Содан кейін файлды қалаған қалтаға орналастырыңыз (әдетте «сценарийлер» қалтасы үшін пайдаланылады) және оны бетке қосыңыз:

```
<script src="scripts/jquery-2.0.0.min.js"></script>
```

<Head> тегіндегі jQuery файлына сілтеме жіберу.

jQuery веб-бет элементтерін басқаратындықтан, jQuery-ге CSS мәнер кестесіне сілтеме жіберіңіз.

Егер бұл сценарий jQuery кітапханасын пайдаланса, jQuery файлына қосқаннан кейін `<script> ... </script>` тегтерін жіберіңіз.

Сіз бос HTML элементін жасай аласыз, мысалы, бірнеше жолмен:

- қысқа жазба пішіні арқылы `$("<div>")`
- Нұсқауларды қолдану `$("<div></div>")`
- Нұсқауларды қолдану `$("<div/>")`

Барлық үш әдіс жұмыс істейді, бірақ соған қарамастан, берілген элементте басқа элементтер болуы мүмкін екенін көрсету үшін тегтерді ашу және жабу ұсынылады.

`$ ()` әдісіне жаңа элемент жасаған кезде екінші параметрді элементтің қосымша атрибуттарын анықтайтын JavaScript нысаны ретінде бере аласыз:

```
$("<img/>",  
{src: "images/flower.jpg",  
title: "Rose_in_garden",  
click: function() {...}  
}).appendTo("body");
```

Бұл көрсетілген атрибуттары бар `` элементін жасайды және DOM тармағына қосылады

3.2.5 AJAX туралы мәлімет

AJAX – бұл веб-сервермен браузер деректерін «фонда» алмасудан тұратын веб-қосымшалар үшін интерактивті пайдаланушы интерфейстерін құру тәсілі. Нәтижесінде, деректерді жаңартқанда, веб-бет толығымен қайта жүктелмейді және веб-қосымшалар тезірек және ыңғайлы болады. Орыс тілінде кейде транслитерацияда «Ајах» (мифологиялық атымен ұқсас) деп аталады, бірақ AJAX аббревиатурасы кириллде анықталған аналогы жоқ.

Классикалық веб-бағдарлама үлгісінде:

- пайдаланушы веб-бетке кіріп, оның кейбір элементтерін басады.
- браузер серверге сұраныс жібереді және жібереді.
- жауап ретінде сервер толығымен жаңа веб-бетті жасайды және оны браузерге және т.б. жібереді, содан кейін шолғыш толық бетті толығымен қайта жүктейді.

AJAX пайдалану кезінде:

- пайдаланушы веб-бетке кіріп, оның кейбір элементтерін басады.
- сценарий (JavaScript-де) бетті жаңарту үшін қандай ақпарат қажет екенін анықтайды.
- браузер тиісті сұрауды серверге жібереді.
- сервер сұрау жіберілген құжаттың тек сол бөлігін қайтарады.

– сценарий алынған ақпаратқа (толық бетті қайта жүктеместен) негізделген өзгертулерді енгізеді.

AJAX тәуелсіз технология емес, бірнеше байланысты технологияларды қолдану тұжырымдамасы. AJAX екі негізгі принципке негізделген:

– серверге динамикалық қол жеткізу технологиясын толығымен қайта жүктеместен, мысалы, XMLHttpRequest (басты нысан) арқылы пайдалану;

– балалардың кадрларды динамикалық құру арқылы [1];

– `<script>` тегінің динамикалық жасауы арқылы.

– Google Analytics-те енгізілгендей, `` тегінің динамикалық құрылуы арқылы.

– беттің мазмұнын динамикалық өзгерту үшін DHTML пайдалану;

Интерфейстік әрекеттер DOM элементтерімен (Document Object Model) қолданысқа өзгертіледі, олар пайдаланушының қол жетімді деректерін өңдеу үшін пайдаланылады, нәтижесінде оларды ұсыну өзгереді. Міне, қозғалыстар мен тінтуірді басу, сондай-ақ пернелерді басу. Каскадты стильдер кестелері немесе CSS (ағылшын каскадты стиль кестелері) бағдарлама элементтеріне дәйекті көрініс пен сезім береді және DOM нысандарына кіруді жеңілдетеді. XMLHttpRequest нысаны (немесе ұқсас механизмдер) сервермен асинхронды өзара әрекеттесу үшін, пайдаланушы сұрауларын өңдеу және қажетті деректерді процесте жүктеу үшін пайдаланылады.

Осы төрт технологияның үшеуі - CSS, DOM және JavaScript - DHTML (Dynamic HTML). Кейбір мамандардың (кітаптардың) айтуынша, [3] 1997 жылы пайда болған DHTML қоры үлкен уәде берді, бірақ оларды ақтамады.

Деректерді беру пішімі ретінде қарапайым мәтіннің фрагменттері, HTML коды, JSON немесе XML пайдаланылуы мүмкін.

AJAX термині алғаш рет 2005 жылы 18 ақпанда Джесси Джеймс Гарреттің «Веб-қосымшаларға жаңа көзқарас» мақаласында жарияланды [4]. Гаррет терминологияны клиенттің оларға ұсынған жаңа технологиялар жиынтығын атаған кезде түсіндірді.

Дегенмен, бір немесе бірнеше формаларда, мысалы, 1998 жылы Microsoft ұсынған «қашықтан сценарийлер» тәсілінде немесе IFRAME HTML элементін пайдаланып, 1996 жылы Internet Explorer 3-те пайда болған көптеген технологияларды қолдана бастады.

AJAX Google-ді Gmail, Google Maps және Google ұсыныста қолданғаннан кейін танымал болды

AJAX веб-интерфейсті әзірлеу әдістерінің жиынтығы үшін сәнді атау болып табылады, ол веб-парақты көрінетін қайтадан жүктеместен серверге динамикалық сұраулар жасауға мүмкіндік береді: пайдаланушысы браузер деректерді сұраған кезде байқамайды.

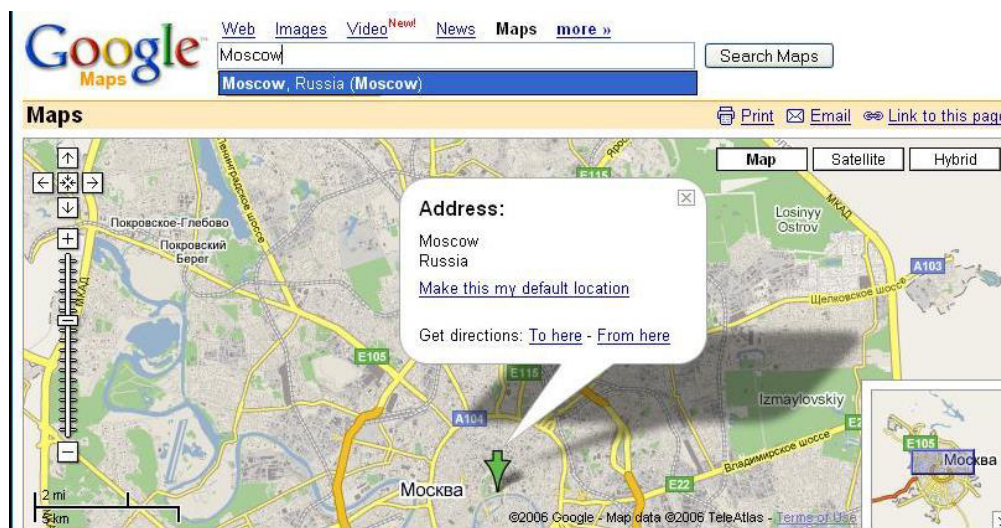
AJAX беттерді жаңарту қажеттілігі болмаған кезде динамикалық және асинхронды веб-әзірлеуді қамтамасыз етеді. Біздің мақала сізді ұзақ таныс және сүйікті технологиялармен үйлестіру арқылы сізге веб-әзірлеудегі жаңалықтардың мүмкіндіктері туралы таныстырады.

Мүмкін, кез-келген әзірлеуші әдеттегі емес, жаңа емес веб-бетті аса қызықты нәрсеге айналдыруды армандайды. Енді он жыл бұрын веб-технологияға біраз өмір салуға тырысуға болады. Егер сізге жаңару қажет болса, байытып, өзіңіздің шығармашылықты интерактивті етіңіз, AJAX-мен танысыңыз.

Google немесе Gmail веб-клиентін пайдаланғанда, сіз AJAX негізіндегі шешімді алдыңыз. Бұл бет жанартылмаған динамикалық және асинхронды әрекеттерді қамтамасыз ететін конгломерат технологиясы. AJAX арқылы пайдаланушы ең кеңейтілген клиенттермен жұмыс істеу сияқты веб-беттермен өзара әрекеттеседі.

Қазіргі уақытта көптеген адамдар AJAX туралы айтады. AJAX-нен кейінгі технология инновацияға мүлде ұқсамайды, бірақ жақында XMLHttpRequest нысанын қолданатын кейбір қуатты жаңа қосымшалар пайда болды, олар клиенттік бейнені жаңартып тұру тұжырымдамасына жаңа өмірге дем берді.

Бұл жаңа қосымшалардың ең танымал түрі Google Maps болып табылады. Оны пайдаланып, планетаның картасында белгілі бір аумақты табуға болады, одан кейін кішігірім нысандарға жылжып, бетті жылжытпай картаны сүйреңіз.



3.2-сурет – Google Maps

AJAX - бұл Интернет пайда болғаннан бері қолданылған технологиялар жинағы. Ал мұнда AJAX көзделген мүмкіндіктері (ұсынылған Dzhis Garrett Жақып (Джесси Джеймс Garrett) ретінде, ол бірінші асинхронды JavaScript + XML үшін мерзімін 'AJAX «ойлап») болып табылады:

- XHTML және CSS арқылы стандартты презентация;
- Динамикалық дисплей және құжат нысанының үлгісін қолданып өзара әрекеттесу;
- XML және XSLT арқылы деректер алмасу және манипуляциялау;
- XMLHttpRequest арқылы асинхронды деректерді алу;

– JavaScript бірге байланыстырады.

Қысқаша айтқанда, AJAX беттерді үнемі жаңартып отырудың қажеті жоқ жылдам жауап веб-қосымшаларын жазуға мүмкіндік береді. AJAX - барлық негізгі браузерлер қолдайтын қарапайым технология. Қысқаша айтсақ, AJAX-ті енгізудің жалғыз алғышарты JavaScript-ті білу болып табылады.

AJAX әрекеті портал сценарийінде көрсету оңай: электрондық хаттарды қарау. Көптеген веб-порталдар пайдаланушыларға негізгі беттің пошта жәшігінің мазмұнын алдын-ала қарауға мүмкіндік береді. Дегенмен, хабардың мазмұнын көру үшін, әрбір хабарды бөлек басып, бет әр уақытта жаңартылуы керек. Осы мысалда сіз бар AJAX веб-технологияларын қолданып, Outlook Express және Mozilla Thunderbird сияқты ұқсас бай мінез-құлықты іске асыру мүмкіндігін көре аласыз.

Осыған сүйене отырып, портлет тек соңғы хабарлар тізіміне ғана емес, сонымен бірге өздерінің алдын-ала қарауын қамтамасыз ететін портлетті салады - олардың бәрі басты бетке жаңару қажет жоқ.

Хабардың мазмұнына қол жеткізу үшін хабарларға қол жеткізуге мүмкіндік беретін сервер компоненті қажет. Біздің мысалда симулятор ретінде қарапайым сервлет бар, ол келесі форматта үтірлермен бөлінген хабарларды көрсететін: кімнен, тақырыптан, күнінен және хабардың корпусынан:

```
@Override
protected void service(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
    if ("1".equals(req.getParameter("message_id"))) {
        resp.getWriter().println( "John Doe,Greetings,10 10-2005,Hi.I am
doing good" ) ;
    } else if ("2".equals(req.getParameter("message_id"))) {
        resp.getWriter().println( "Joanna Doe,Hi,10-10-2005,Document is
complete." );
    } else if ("3".equals(req.getParameter("message_id"))) {
        resp.getWriter().println("Joanna Doe,Hi,10-10-2005,Meeting is at
3.");
    }
    ...
} else {
    resp.getWriter().println( "NA,NA,NA,Nothing to display" );
} //end else
} //end service
```

Порталда порталды қарау үшін портал терезесі / портлеті болады, сол жағында кіріс хабарлардың оңайлатылған тізімімен және оң жағындағы көру панелімен қарауға болады. Егер хабарды сол жақта бассаңыз, серверге сұрау және экрандағы хабарламаның мәтінін нақты уақыт режимінде көрсету алдын ала қарау тақтасында көрінеді, AJAX осы процеске қатысады.

Көптеген бағдарламашылар JavaScript-ні отладтау әдістерінің жоқтығы мен жоғары қателік сезімталдығына байланысты шартты түрде оңтайлы шешім ретінде қабылдайды. Сондықтан AJAX - ымыралы шешім. Бұл технология Java немесе C # секілді сенімді тілдерден және презентацияларды қолдана отырып, JavaScript-тің инновациялық шабытынан пайда көруге көмектеседі.

AJAX танымалдығы мен JavaScript-ны пайдалануды кеңейту контекстінде браузер сатушылары JavaScript объектілеріне қатысты инновацияларды жалғастырады және осы объектілердің үйлесімділігін жақсартуға, олардың қауіпсіз пайдаланылуына және түзетулерін жақсартуға көмектеседі деп үміттенеміз.

Сонымен бірге AJAX болашақта Интернет-порталдардың жаңа буындарын және интерактивті қосымшаларды қалыптастыра алады. AJAX көмегімен Yahoo, Google немесе MSN сияқты интернет жаңалықтар порталдарында пайдаланушылар бірдей беттен, нақты тақырыптарды қоса, барлық тақырыптар бойынша ақпаратты ала алады. Қолданыстағы веб-технологиялар мен Интернет-инфрақұрылымды пайдалана отырып іске асырылуы мүмкін клиенттердің әр түрлі мүмкіндіктері тартымды көрінеді. Интерактивті қосымшалар қазірдің өзінде AJAX-ді бейімдеді - мысалы, Google осы технологияны Gmail-тан танымал электрондық пошта клиенті үшін пайдаланады. Бұл үрдіс жалғасады деп үміттенеміз.

Даму тобы бұл технологияға қарапайым қолжетімділікті және оқудың қарапайымдылығын жақсы көреді. Жоғарыда айтылғандай, бұл технология қазіргі заманғы барлық браузерлерде қол жетімді. Сонымен қатар, ол J2EE немесе .NET секілді күрделі бағдарламалау дағдыларын қажет етпейді, сонымен бірге, соңғы пайдаланушыны нәтижелермен таңдай аласыз.

AJAX танымал болған сайын, үшінші тарап кеңейтімдері, мысалы, отладтау, кросс-платформаны әзірлеу, XML өңдеу сияқты күрделі тапсырмаларды жеңілдету пайда болады. Бүгінде пайдалы болуы мүмкін ең маңызды кеңейтімдердің кейбірі:

– Greasemonkey – DHTML сценарийлерін орнатуға мүмкіндік беретін Firefox кеңейтімі. Кейбір блог авторлары Greasemonkey сипаттарын пайдаланып AJAX бағдарламаларын қадағалауға және түзетуге мүмкіндік беретін сценарийлер жасаған.

– Sarrisa – AJAX-мен бірге браузерлердің іске асырылуын және тестілеуді ең генерикалық XML операцияларын автоматтандыру және жеңілдететін скрипт кітапханасы (мысалы, URL-дан жүктелген XML құжаты, іске қосу, XPath өңдеу, XSLT).

– Direct Web Remoting (DWR) – JavaScript бағдарламаларының Java сервер кодын қашықтан белсендіруге мүмкіндік беретін қашықтағы AJAX-негізделген Java кітапханасы.

– FireBug – Firefox кеңейтімі.

Артықшылықтары:

– Трафикті үнемдеу. AJAX көмегімен веб-бағдарламамен жұмыс жасағанда трафикті азайтуға мүмкіндік береді, себебі бүкіл бетті жүктеудің

орнына тек өзгерген бөлікті жүктеу немесе JSON немесе XML пішімінде деректер жиынтығын қабылдау және беру үшін ғана JavaScript мазмұнын өзгерту арқылы бет мазмұнын өзгерту жеткілікті.

– Сервер жүктемесін азайту. AJAX-ді дұрыс іске асыра отырып, ол бірнеше рет серверде жүктемені азайтады. Атап айтқанда, сайттың барлық беттері көбіне әртүрлі файлдарға қоңырау шалып, сценарийлерді өңдеуге арналған уақытша элементтер («тақырып», «навигациялық панель», «жертөле» және т.б.) және кейде дерекқорға сұраулар) - барлық беттің жүктемесін тек мазмұн бөлігін жасау мен беру арқылы ауыстыратын болсақ, мұның бәрін алып тастауға болады.

– Интерфейстің әрекет етуін жылдамдату. Өзгертілген бөліктің жүктелуі әлдеқайда жылдам болғандықтан, пайдаланушы әрекеттерінің нәтижесін тезірек және беттің жыпылықтауынсыз (толық қайта жүктеу кезінде туындайтын) көреді.

– Интерактивті өңдеу мүмкіндіктері. Мысалы, Google-да іздеу сұрауын енгізгенде, кеңестер ықтимал сұрау опциялары арқылы көрсетіледі. Тіркеу кезінде көптеген сайттарда пайдаланушы атқа кіреді және бұл атаудың бар немесе жоқ екендігін бірден көреді. AJAX сөйлесу бөлмелерін, әкімшілік панельдерді және уақыт өткен сайын өзгертін деректерді көрсететін басқа құралдарды бағдарламалау үшін пайдалы.

– Мультимедиа тоқтатылмайды. Бет қайта жүктелмейді, ойнатқыш жұмысын жалғастырады. Себебі AJAX аудио және бейне хостинг кезінде құнды.

Кемшіліктері:

– Стандартты браузер құралдарымен интеграцияның жоқтығы. Динамикалық түрде құрылған беттер браузер беттерге кіру тарихында тіркелмейді, сондықтан «Артқа» түймесі жұмыс істемейді, бұл пайдаланушыларға бұрын көрсетілген беттерге оралу мүмкіндігін береді, бірақ бұл мәселені шешуге болатын сценарийлер бар. Тұрақты URL мекенжайы бар беттің мазмұнын өзгертудің тағы бір кемшілігі - бетбелгіні қалаған материалға сақтаудың мүмкін еместігі. Мәселе History.pushState [5] көмегімен табысты шешілуі мүмкін.

– Динамикалық жүктелген мазмұн іздеу жүйелерінде қол жетімді емес. Іздеу механизмдері JavaScript-ды орындай алмайды, сондықтан әзірлеушілер сайттың мазмұнына қол жеткізудің баламалы жолдарын ойластыруы керек (бірақ 2017 жылға қарай ол бұдан былай сәйкес емес

– Бухгалтерлік есептің сайт статистикасының ескі әдістері маңызды емес. Көптеген статистикалық қызметтер жаңа бет көріністерін бақылайды. AJAX кеңінен қолданылатын сайттар үшін мұндай статистика өзектілігін жоғалтады.

– Жобаның күрделенуі. Деректерді өңдеу логикасы қайта бөлінеді-деректерді бастапқы форматтау процестерін клиент жағына бөлу және ішінара көшіру жүреді. Бұл форматтар мен типтердің тұтастығын бақылауды

қиындатады. Технологияның соңғы әсері кодтауға және жобаны басқаруға арналған шығындардың негізсіз өсуімен, сондай-ақ соңғы пайдаланушылар үшін сервистің қолжетімділігінің төмендеу тәуекелімен нивелленуі мүмкін.

– Браузерде JavaScript қосылған қажет. JavaScript қауіпсіздік тұрғысынан өшірілуі мүмкін. Және, әрине, AJAX-беттер толық емес браузерлерге, роботтарға және веб-мұрағаттарға жету қиын.

– Кейбір Ajax сценарийлері стандартты емес кодтауды көрсету мәселелері. AJAX және кириллица мәселелері туралы интернетте көп айтылды

– Қатты бағдарламалау кезінде төмен жылдамдық. AJAX жылдамдықты арттыруға арналған. Бірақ AJAX-сұраулар бір бетте көп болғанда және, мысалы, әрбір нұқу бойынша тізімді жүктейді, AJAX-бет тіпті баяу дәстүрлі болады.

– Сенімсіз қосылыстарда жаман мінез-құлық. Егер Байланыс жиі жоғалса (тасушының жоғалуына немесе арнаның жүктелуіне байланысты), кәдімгі бетті кем дегенде қайта жүктеуге болады. AJAX-бет (мысалы, "шексіз" айналдыру) басынан бастап қайта жүктеуге және тоқтап тұрған жерден іздеуге тура келеді. Параллельді жұмыс-Ajax ерекшелігі - мұнда аю қызметі, дель және кішкентай қосылыстарға тар арна жоқ, және кейбіреулердің үзілуі ықтималдығы жоғары. Тарих API ішінара шешіледі.

– Басқа сайттармен сұрау салу қатері. AJAX-сұраудың нәтижесі JavaScript-код (атап айтқанда, JSON) болуы мүмкін. XMLHttpRequest тек бір доменде ғана әрекет етеді, бірақ `<script>` - жоқ. Егер жазсақ

```
<script type="text/javascript" src="http://example.org/inbox.php (недоступная ссылка)"></script>
```

сонда HTML-ге осы компьютерде авторизацияланған пайдаланушының кіріс хаттары қапшығы енгізіледі example.org ескерту. Қорғау үшін POST сұрау қолданылады. Бірақ GET идепотентті болып саналады және сондықтан кэширленеді, POST-жоқ, Сондықтан Google жауаптың басына шексіз цикл енгізеді: AJAX жауап бере алады, соның ішінде циклды алып тастайды, ал тэг `<script>` скриптті бар және цикл ретінде қосады.

3.3 OnLine өтініш жұмысының жалпы схемасы

OnLine өтініш жүйесінің кестелер тізімі 3.3-суретте көрсетілген.

PostgreSQL 10.6 (Ubuntu 10.6-0ubuntu0.18.04.1) выполняется на localhost:5432 -- Вы зарегистрированы как 'postgres'

phpPgAdmin: PostgreSQL: postgres: public:

Таблица	Пользователи	Tablespace	Estimated row count	Действия										Комментарий
arrival_drugs	postgres		0	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		приход лекарств
departments	postgres		81	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
disease_history	postgres		928	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
doctor_adaptive_marker	postgres		140	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		адаптивные маркеры докторов
doctors	postgres		648	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
doctors_selected_forms	postgres		608	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
drug_application	postgres		103	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		заявка на препарат
drug_circulation_department	postgres		0	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		отделы оборота лекарственных средств
forms	postgres		417	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
forms_group	postgres		0	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
hospital_adaptive_marker	postgres		0	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		адаптивный маркер больницы
hospital_forms	postgres		1541	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
hospital_types	postgres		0	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
hospitals	postgres		7	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
list_preparats	postgres		84	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		список препаратов
managers	postgres		0	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
markers	postgres		2660	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
medicinal_services	postgres		2070	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
messages	postgres		39	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		Обмен сообщениями
messaging	postgres		11	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
mkb	postgres		14749	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
operation	postgres		2788	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
patient_record	postgres		3128	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		запись пациента к врачу
patients	postgres		11480	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
patient_care	postgres		187	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
plans	postgres		738	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
ranks	postgres		0	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
ready_phrase_doctor	postgres		251	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
receptions	postgres		3055	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
rows	postgres		8768	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
scanner_history	postgres		0	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		
schedule_apparatus	postgres		59	Просмотреть	Выбрать	Вставить	Очистить	Изменить	Удалить	Перестроить	Анализировать	Перестроить индекс		расписание аппаратур

3.3-сурет – Кестелер тізімі

PostgreSQL 10.6 (Ubuntu 10.6-0ubuntu0.18.04.1) выполняется на localhost:5432 -- Вы зарегистрированы как 'postgres'

phpPgAdmin: PostgreSQL: postgres: public: disease_history:

Column	Тип данных	Not Null	По умолчанию	Ограничения	Действия			Комментарий
history_id	bigint		nextval('s_disease_history::regclass')		Просмотреть	Изменить	Привилегии	Удалить
form_id	bigint	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
patient_id	bigint	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
doctor_id	bigint	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
form_data	text				Просмотреть	Изменить	Привилегии	Удалить
referer_doctor_id	bigint	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
c_date	date	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
c_time	time(0) without time zone	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
status	smallint	NOT NULL	1::smallint		Просмотреть	Изменить	Привилегии	Удалить
deleted	smallint	NOT NULL	0::smallint		Просмотреть	Изменить	Привилегии	Удалить
visible	smallint	NOT NULL	1::smallint		Просмотреть	Изменить	Привилегии	Удалить
file	text				Просмотреть	Изменить	Привилегии	Удалить
filename	text				Просмотреть	Изменить	Привилегии	Удалить
temporary_form_data	text				Просмотреть	Изменить	Привилегии	Удалить
hospital_id	integer				Просмотреть	Изменить	Привилегии	Удалить

Просмотреть | Выбрать | Вставить | Очистить | Удалить | Добавить атрибут | Изменить

3.4-сурет – Сұралымдар ұяшықтар тізімі

phpPgAdmin: PostgreSQL: postgres: public: doctors:

Column	Тип данных	Not Null	По умолчанию	Ограничения	Действия			Комментарий
doctor_id	bigint	NOT NULL	nextval('s_doctors::regclass')		Просмотреть	Изменить	Привилегии	Удалить
login	character varying(50)	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
password	character varying(100)	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
name	character varying(80)	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
surname	character varying(80)	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
patronymic	character varying(80)	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
hospital_id	bigint	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
department_id	bigint	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
specialty_id	bigint	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
rank	smallint	NOT NULL			Просмотреть	Изменить	Привилегии	Удалить
deleted	smallint	NOT NULL	0::smallint		Просмотреть	Изменить	Привилегии	Удалить
visible	smallint	NOT NULL	1::smallint		Просмотреть	Изменить	Привилегии	Удалить

Просмотреть | Выбрать | Вставить | Очистить | Удалить | Добавить атрибут | Изменить

3.5-сурет – Қолданушылар ұяшықтар тізімі

Жобаны жүзеге асыру үшін серверлік тіл ретінде php тілі, mvc технологиясы және yii2 қосымшасы қолданылды.

Сайттың жұмыс істеу сапасын арттыру мақсатында MVC, jquery, json, ajax қолданылды.

Model-View-Controller (MVC, "модель-көрініс-Контроллер", "Модель-Вид-Контроллер") — қосымшаның деректерін, пайдаланушы интерфейсін және басқару логикасын үш жеке компонентке бөлу схемасы: модель, ұсыну және контроллер — әрбір компоненттің модификациясы тәуелсіз жүзеге асырылуы мүмкін.

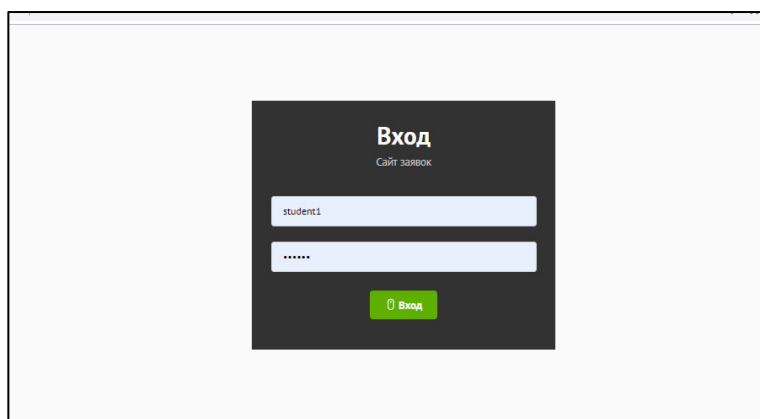
Бұл тұжырымдаманы қолданудың негізгі мақсаты бизнес-логика (модель) оны визуализациялаудан (ұсыныстан, түрден) бөлімшесінен тұрады. Мұндай бөлу есебінен кодты қайта пайдалану мүмкіндігі артады. Осы тұжырымдаманы қолдану пайдаланушы сол деректерді бір уақытта әр түрлі контекстерде және/немесе әр түрлі көзқараспен көруі тиіс жағдайларда неғұрлым пайдалы.

MVC қатаң жүзеге асырылмайды болғандықтан, ол әртүрлі болуы мүмкін. Бизнес-логика орналасуы тиіс жалпы қабылданған анықтама жоқ. Ол контроллерде де, модельде де болуы мүмкін. Соңғы жағдайда модель барлық деректер мен функциялары бар барлық бизнес-объектілерді қамтитын болады.

jQuery ([deɪ̯ kwɘəɹi]) - JavaScript және HTML өзара әрекеттесуіне бағытталған JavaScript кітапханасы. JQuery кітапханасы кез келген DOM элементіне оңай қол жеткізуге, Dom элементінің атрибуттары мен мазмұнына қарап, оларды айла-шарғы жасауға көмектеседі. Сонымен қатар, jQuery кітапханасы AJAX-пен жұмыс істеуге ыңғайлы API ұсынады.

AJAX (asynchronous JavaScript and XML аббревиатурасы) – бетті қайта жүктеусіз сервермен өзара әрекеттесу технологиясы. Бетті түгелдей жаңарту қажет емес болғандықтан, сайтпен жұмыс істеу жылдамдығы және оны пайдалану ыңғайлылығы артады.

Жүйеге кіру үшін қолданушыдан құпия сөз бен логинін теру талап етіледі.



3.6-сурет – Жүйеге кіру беті

Авторизация кезінде логин мен пароль салыстырылады, егер олар дұрыс болса, онда USER_HASH жолына ДБ-ға кәштелетін және қосылатын кездейсоқ жол жасалады. Сонымен қатар, пайдаланушының IP мекен-жайы жазылады(бірақ бұл біздің опциональды болады, өйткені біреу Proxu арқылы

отырады, ал біреу IP динамикалық... мұнда Пайдаланушы өзі қауіпсіздікті немесе ыңғайлылықты таңдайды). Пайдаланушының кукиіне біз оның бірегей идентификаторын және генерацияланған hash жазамыз.

Дұрыс нұсқа-сессияларда немесе куктарда тексеруді ұйымдастыру, егер бар болса - пайдаланушы авторландырылған болса, сессия немесе Кука әдісімен тексеру, авторландыру бетінен керекті бетке қайта бағыттау немесе жүйе жобаланғанына байланысты контентті динамикалық түрде беру. Егер жоқ болса-авторизация бетінің мазмұнын беру немесе оған қайта бағыттау. Авторизация бетін индекстеу қажет болса, SEO Бұл жерде де әсер етеді, тек қайта бағыттау.

Сессиялар жұмыс істеу үшін, ештеңе жоқ, ең бастысы, скрипт басында әрқашан session_start () орындалады. Нәтижесінде, тексеру пайдаланушы авторланбаған жағдайда, қалпына келтіру керек барлық скриптер беттерді жұмыс істейді.

Қолданушы өз логині мен құпия сөзін дұрыс толтырса роліне байланысты арнайы бетке өтеді.

ФИО	Дата заявки	Время заявки	Кол-во новых заявок(2019-05-20)	Завершить прием
Студент1	2019-05-01	15:04:04	1	✘

3.7-сурет – Студенттер мен мұғалімдер үшін өту беті.

Қолданушы бұл бетте бірнеше командаларды жүзеге асыра алады. Олар сұралым қосу, сұралым тізімі, қолданушыға жіберілген сұралым тізімдері, қолданушы жіберген сұралымдар.

Сайт заявок	Написать заявку	Студент1 (Завершить работу)	
Выполнить	Направить	Удалить	История заявок
Ваши созданные формы			
Заявление (академический отпуск)			
Разрешение на перевод			

3.8-сурет – Студенттер мен мұғалімдер үшін өту бетінің негізгі құрамы

Сұралым қосу батырмасын басу қолданушы келесі бетке өтеді.

Сайт заявок Написать заявку Студент1
(Завершить работу)

Ректору АУЭС
Сагитовой С.С.
от студента(ки)

специальности

гр.

обучающегося на

Ф.И.О. студента

ЗАЯВЛЕНИЕ

Прошу Вашего разрешения на предоставление академического отпуска в связи (причина)

Медицинскую справку ВКК ((Повестку военкомата),(свидетельство о рождении ребенка)) прилагаю.

3.9-сурет – Сұралым қосу беті

Қолданушы бұл бетте сұралымды , уақытын және керек болған жағдайда файлды т.б. қоса алады және қосу батырмасын басу арқылы сұралымды сақтайды.

Деректерді формалардан серверге жібергенде, кейде пайдаланушы енгізген ақпаратты сақтау қажет, содан кейін бетті қайта жүктегеннен кейін, пішін мәндерін қалпына келтіру және деректерді қосуға кедергі келтіретін қателерді көрсету қажет.

Javascript – бұл күш. Сонымен қатар, ол мәліметтерді жоғалту ықтималдығын азайтатын барлық өрістер толтырылғанша ғана емес, сондай-ақ деректерді Ajax көмегімен бетті қайта жүктемей жіберуге де болады. Бұл жағдайда деректер экрандардан жоғалмайды, керісінше, бұл нысанды жою үшін деректерді жібергеннен кейін қандай әрекеттерді қабылдау мәселесі жиі кездеседі. Бұл әдіс осы нысандарды жоғалтып емес, өте сенімді, бірақ өкінішке орай, барлық браузерлер оны түсінбейді. Мысалы, барлық ұялы телефондар мен смартфондар javascript-ті қолдамайды, сонымен қатар ajax-ті қолдамайды, сондықтан кейбір өрісті толтыруды ұмытып қалсаңыз, оларға жол бермейді.

Сайт заявок Написать заявку Студент1
(Завершить работу)

Выполнить Направить Удалить История заявок

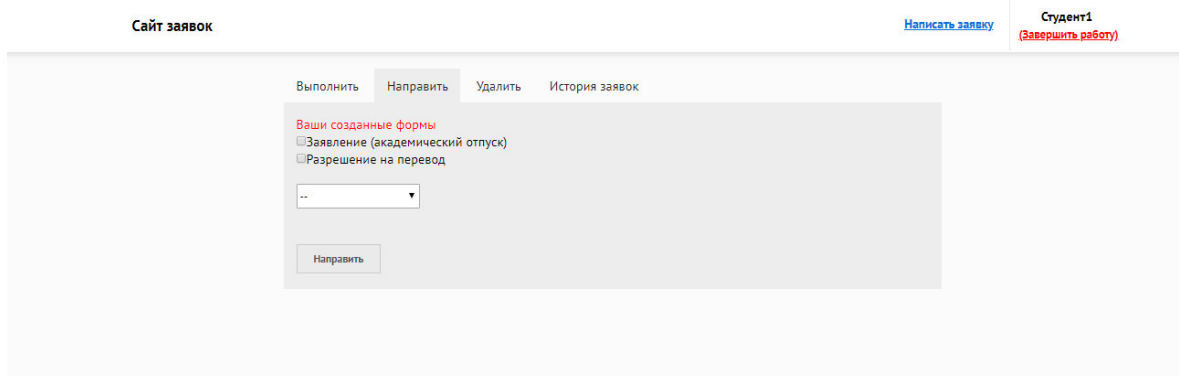
№	Дата	Время	Форма	Выполняющий врач	Направляющий врач	
1	2019-05-20	09:32:58	Разрешение на перевод	Студент1	-	Открыть

3.10-сурет – Сұралым тізімі

Қолданушы келген келген сұраныстарға жауап бере алады ол үшін статус ұяшығындағы іске асырылды немесе кері қайтарылды батырмасын басу керек.

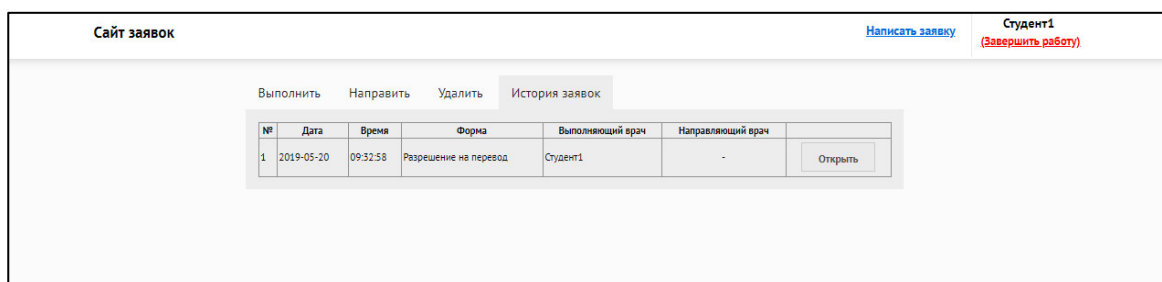
Егер қолданушы сұранысты басқа қолданушыға жібергісі келсе бөлісу батырмасын басып керек қолданушыны таңдай алады. SQL-инъекциядан

қорғауды қамтамасыз етеді. Ең оңтайлы әдіс-функцияның көмегімен барлық түскен сансыз деректерді өңдеу



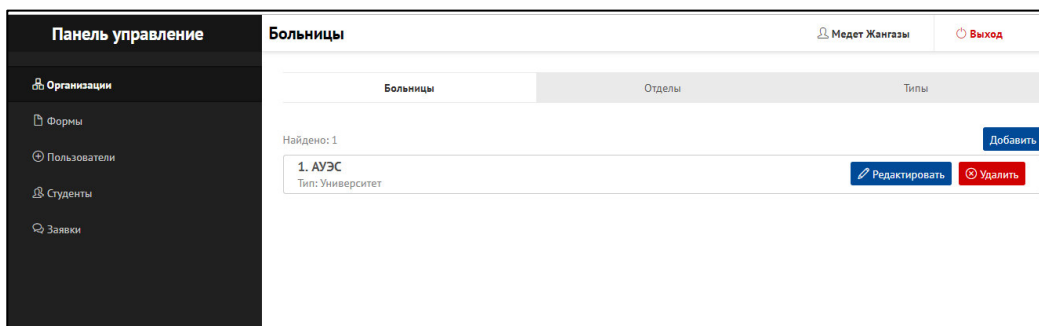
3.11-сурет – Сұраныспен бөлісу

Есеп батырмасын басу арқылы қолданушы барлық сұраныс тізімдерін және статустарын көру мүмкіндігіне ие болады. Программаны жұмыс істеуі сапасын арттыру мақсатында ајах, mvc, jquery, json программаларын қолдандым.



3.12-сурет - Барлық сұраныс тізімдері

Админ ролі бар қолданушы өз логині мен құпия сөзін енгізіп өту батырмасын басқанда келесі бетке өтеді.



3.13-сурет – Админ парақшасы

4 Экономикалық бөлім

4.1 Экономикалық бөлімнің мақсаты

Бұл дипломдық жұмыста ЖОО арналған білім алушының білімін бағалауға арналған тестілеудің OnLine бағдарламалық қамтамасын құрылды. Жүйе тестілеу және тестілеуді дайындау бойынша қолмен орындалатын операциялардың күрделілігін төмендетуге арналған.

Жүйені енгізудің тиімділігі студенттерді тестілеу үрдісін автоматтандыру болып табылады.

Еңбекақы төлеу шығындарын төмендету қаржы шығындарын азайтады, бұл өнімділіктің және үнемдеудің жалпы өсуіне әкеледі. Бұл бөлімнің негізгі мақсаты ғылыми зерттеулердің шығындар көлемін анықтау, әлеуметтік өндірістегі негізгі және тиісті нәтижелерді пайдаланудың экономикалық тиімділігін анықтау, дипломдық жұмыстың техникалық мәселелерін шешу болып табылады. Қабылданған ғылыми-техникалық шешімнің тиімділігін бағалау барлық қажетті шығыстар мен шығындарды ескеріп, белгілі бір схемаға сәйкес бірқатар қажетті есептеулерді талап етеді.

Алынған ғылыми-техникалық шешімдердің тиімділігін бағалау белгілі бір схема бойынша бірқатар қажетті есептеулерді талап ететін барлық қажетті шығындар мен шығыстарды талап етеді.

4.2 Бағдарламалық өнімдерді әзірлеудің күрделілігін есептеу

4.1-кесте – Жұмыстарды кезендер мен түрлерге бөлу және олардың еңбек қарқындылығын бағалау

Жұмыс бөлімдері	Жұмыстың түрі	Әзірлеу күрделілігі, адам× сағ.
1	Пәндік аймақты талдау	40
2	Техникалық тапсырмаларды әзірлеу	53
3	Деректер базасын жобалау	72
4	Бағдарламаның интерфейсін әзірлеу	180
5	Жүйені тестілеу	150
6	Құжаттау	60
Барлығы		505

4.3 Бағдарламалық өнімді әзірлеуге кеткен шығындарды есептеу

Бағдарламалық өнімді әзілеуге шығыстарды анықтау келесі мәселелерді қамтитын тиісті бағалауды жасау арқылы жасалады:

- материалдық шығындар;
- еңбек шығындары;
- әлеуметтік салық;

- негізгі құралдардың амортизациясы;
- басқа шығындар.

4.2-кесте – Материалдық ресурстарға кеткен шығындар

Материалдық ресурстың аты	Өлшем бірлігі	Саны	Бір дана құны, тг	Сомасы, тг
Қалам	Дана	10	100,00	1000,00
Қағаз	Орам	5	1400,00	7000,00
Дәптер	Дана	20	150	3000,00
Барлығы				11000,00

4.2-кесте – Негізгі ресурстарға кеткен шығындар

Материалдық ресурстың аты	Өлшем бірлігі	Саны	Бір дана құны, тг	Сомасы, тг
Ноутбук Lenovo 300	Дана	1	230000,00	230000,00
ПО Sublime Text	Дана	1	77086.33	77086,33
Барлығы				307086,33

Энергия шығындары бағдарламалық қамтамасыз етуді әзірлеу кезеңінің ұзақтығына, бағдарламалық жасақтамаға жұмсалатын кВт/сағатқа және 1кВт/сағ тарифіне негізделген. Алматы қаласында заңды тұлғалар үшін тариф 2019 жылы ҚҚС есебімен 1 кВт / сағ үшін 18,32 теңгені құрайды («АлматыЭнергоСбыт» ЖШС ресми сайтында берілген деректерге сәйкес). Шығындар 4.3-кестеде берілген.

Электр қуатына кеткен шығындардың жалпы сомасын (Z_3) (4.1) формула бойынша есептеледі:

$$Z_3 = \sum_{i=1}^n M_i \cdot K_i \cdot T_i \cdot C \quad (4.1)$$

мұндағы, M_i - тұтынылатын қуат, кВт;

K_i - пайдалану коэффициенті ($K_i=0.9$);

T_i - жабдық жұмыс уақытысағ;

C - кВт-сағат электр энергиясының бағасы, тг/кВт×сағ;

i – электроқұрылғы түрі;

n – электроқұрылғы саны.

$$Z_3 = 0.9 \cdot 0.8 \cdot 505 \cdot 18.32 = 6661,15$$

$$Z_3 = 0.3 \cdot 0.7 \cdot 505 \cdot 18.32 = 1942,83$$

Электр қуатына кеткен шығындардың қорытындысы 4.3-кестеде көрсетілген.

4.3-кесте – Электроэнергия шығындары

Жабдық атауы	Тұтын ылаты н қуат, кВт	Қуаттың пайдалану коэффицие нті	Жабдықты ң жұмыс уақыты, сағ	Электроэн ергия құны, тг/кВт сағ	Сомасы, тг
Ноутбук Lenovo 300	0.9	0.8	505	18,32	6661.15
Жарықтандыру	0,3	0,7	505	18,32	1942,83
Барлығы					8603,98

4.4 Еңбекақы төлеу шығындарын есептеу

2019 жылы инженер-әзірлеушінің орташа жалақысы 200 000 тг, ал программисттікі 160 000 тг (Алматы қаласы үшін).

Қызметкердің айлық жұмыс сағаттары (4.2) формула бойынша анықталады:

$$Ч_m = N_m \cdot Ч_{pd} \quad (4.2)$$

мұндағы, $Ч_m$ —қызметкердің айлық жұмыс сағаттары;

N_m —айлық жұмыс күндер саны;

$Ч_{pd}$ —бір күндік жұмыс сағаттар саны.

$$Ч_m = 21 \cdot 8 = 168$$

Қызметкердің жұмыс сағатының құнын (4.3) формула бойынша анықталады:

$$ЧC_i = \frac{ЗП_i}{ФРВ_i}, \quad (4.3)$$

Инженер-әзірлеуші:

$$ЧС_i = \frac{200000}{168} = 1190,48 \text{ тг}$$

Программист:

$$ЧС_i = \frac{160000}{168} = 952,38 \text{ тг}$$

мұндағы, $ЗП_i$ – i -ші қызметкердің айлық жалақысы, тг;

$ФРВ_i$ – i -ші қызметкердің жұмыс уақыттың айлық фонды, сағ.

Бағдарламалық өнімдерді әзірлеудің күрделілігін анықтау үшін 4.1-кесте қолданылады.

Бағдарламалық өнімдерді әзірлеудің күрделілігі инженер-әзірлеуші үшін 263 адам×сағатты құрайды (пәндік аймақты талдау, тапсырмаларды айқындау, техникалық тапсырмаларды әзірлеу, деректер базасын жобалау, бағдарлама интерфейсін әзірлеу, жүйені тестілеу, құжаттау).

$$T_1 = 40 + 53 + 72 + 150 + 60 = 303$$

Бағдарламалық өнімдерді әзірлеудің күрделілігі программист үшін 263 адам×сағатты құрайды (тапсырмаларды айқындау, бағдарлама модулін әзірлеу, жүйені тестілеу).

$$T_1 = 40 + 180 + 150 = 370$$

Еңбекақы шығындарының жалпы сомасы ($З_{тр}$) (4.4) формула бойынша есептелінеді:

$$З_{тр} = \sum_{i=1}^n ЧС_i \cdot T_i \quad (4.4)$$

мұндағы $ЧС_i$ – i -ші қызметкердің бір сағаттық жалақысы тг;

T_i – бағдарламалық өнімді әзірлеудің күрделілігі, адам×сағ;

i – қызметкердің категориясы;

n – бағдарламалық өнімді әзірлеумен айналысатын қызметкерлер саны.

Инженер-әзірлеуші:

$$З_{тр} = 1190,48 \cdot 303 = 360715,44 \text{ тг}$$

Программист:

$$З_{тр} = 952,38 \cdot 370 = 352380,60 \text{ тг}$$

Жалпы сомасы:

$$З_{тр,о} = 360715,44 + 352380,6 = 713096,04 \text{ тг}$$

4.4-кесте – Еңбекақы шығындары

Квалификация	Бағдарламалық өнімді әзірлеудің күрделілігі, адам×сағ.	Бір сағаттық еңбекақысы, тг/сағ	Сомасы, тг
Инженер-әзірлеуші	303	1190,48	360715,44
Программист	370	952,38	352380,60
Барлығы			713096,04

Қосымша еңбекақы:

$$Z_{\text{қос}} = Z_{\text{тр}} \cdot 10\%$$

$$Z_{\text{қос}} = 713096,04 \cdot 0,1 = 71309,60 \text{ тг}$$

Еңбекақы қоры:

$$\Phi_{\text{зп}} = Z_{\text{тр.о}} + Z_{\text{доп}}$$

$$\Phi_{\text{зп}} = 713096,04 + 71309,60 = 784405,64 \text{ тг}$$

Әлеуметтік салықты есептеу:

$$H_c = (\Phi_{\text{зп}} - \text{ОПВ}) \cdot 11\%$$

мұндағы, НЗҚ – негізгі зейнетақы қоры- $\Phi_{\text{зп}}$ -ның 10%-ы.

$$H_c = (784405,64 - (784405,64 \cdot 0.1)) \cdot 0.11 = 77656,15 \text{ тг}$$

Негізгі қордың амортизациясы есебі. Амортизациялық аударымдардың жалпы сомасы келесі формуламен есептелінеді:

$$Z_{\text{тр}} = \sum_{i=1}^n \frac{\Phi_i \cdot H_{Ai} \cdot T_{\text{НИР}i}}{100 \cdot T_{\text{эф}i}}, \quad (4.5)$$

мұндағы, Φ_i – i -ші НҚ құны, тг;

H_{Ai} – i -ші НҚ жылдық амортизация нормасы, %;

$T_{\text{НИР}i}$ – бағдарламалық өнімді әзірлеудің барлық уақытындағы i -ші НҚ жұмыс уақыты, сағ;

$T_{\text{эф}i}$ – i -ші НҚ жылдық жұмыс уақытының эффективті қоры, сағ/жыл;

i – НҚ түрі;

n – НҚ саны.

НҚ амортизацияның жылдық нормасының есебі:

$$H_{Ai} = \frac{100}{T_{Ni}}$$

$$H_{Ai} = \frac{100}{4} = 25 \quad (4.6)$$

мұндағы, T_{Ni} – i -ші НҚ-ң мүмкін қолданыс уақыты, жыл.

Бағдарламалық өнімді әзірлеуге жұмсалған бағдарламалық жасақтаманың жұмыс уақытын табу үшін 4.1-кестенің мәліметтері қолданылады.

Бағдарламалық өнімді әзірлеу кезіндегі Open Server Panel және PhpStorm бағдарламалық жасақтамаларының жұмыс уақыты 351 сағатты құрайды (деректер базасын жобалау, бағдарлама интерфейсін әзірлеу, бағдарлама модулін әзірлеу, жүйені тестілеу).

$$T_1 = 72 + 180 + 150 = 402$$

Жабдық:

$$З_{AM} = \frac{230000 \cdot 25 \cdot 505}{100 \cdot 1920} = 15123,70 \text{ тг}$$

Бағдарламалық жасақтама:

$$З_{AM} = \frac{77086,33 \cdot 25 \cdot 402}{100 \cdot 1920} = 4034,99 \text{ тг}$$

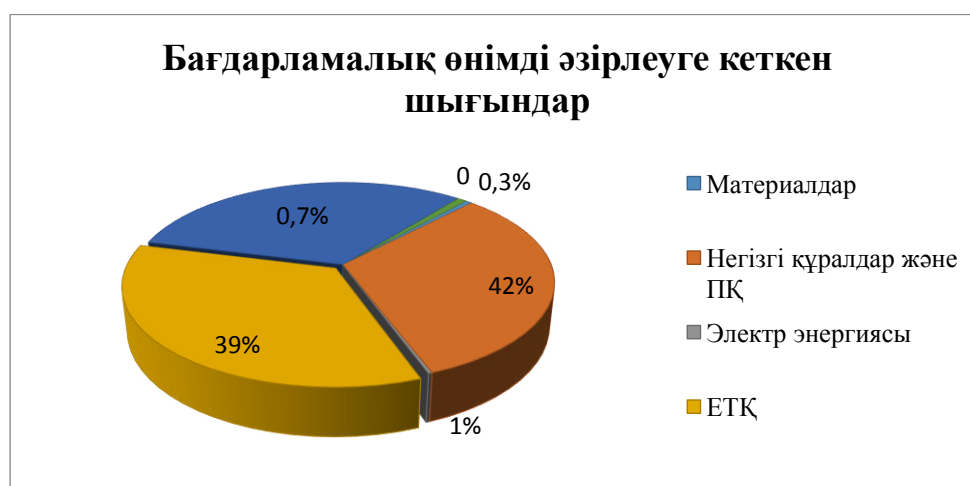
4.5-кесте – Негізгі қорлар амортизациясы (НҚ)

Жабдық және бағдарламалық жасақтама атауы	Жабдық және бағдарламалық жасақтама құны, тг	Амортизацияның жылдық нормасы, %	Жабдық және бағдарламалық жасақтаманың эффективті уақыт қоры, сағ/жыл	БӨ-ні әзірлеуге жұмсалған құрылғының және бағдарламалық жасақтаманың жұмыс уақыты, сағ	Сомасы, тг
Ноутбук SonyVAIO SVE171E1 3V	230000	25	1920	505	15123,70
БЖ PhpStorm	75709,44	25	1920	402	4034,99

Жалпы осы бағдарламалық өнімді әзірлеуге кеткен шығындар 4.6-кестеде көрсетілген.

4.6-кесте – Бағдарламалық өнімді әзірлеуге жұмсалған шығындар кестесі

Шығындар түрі	Сомасы, тг
1. Материалдық шығындар, соның ішінде:	
- материалдар	11000,00
- негізгі құралдар және ПҚ	307086,33
- электр энергиясы	8603,98
2. ЕТҚ	784405,64
3. Әлеуметтік салық	713096,04
4. Негізгі қордың амортизациясы	19158,69
Барлығы	1843350,69



4.1-сурет – Бағдарламалық өнімді әзірлеуге кеткен шығындар

4.5 Бағдарламалық өнімнің мүмкін бағасын анықтау

Қолданылатын бағдарламалық өнім үшін келісім-шарттық бағасы (C_d) (4.7) формула бойынша есептеледі:

$$C_d = Z_{НИР} \cdot \left(1 + \frac{P}{100}\right), \quad (4.7)$$

мұндағы, $Z_{НИР}$ —бағдарламалық өнімді әзірлеуге жұмсалған шығындар (1.6-кестеден), тг;

P – бағдарламалық өнімнің орташа табыстылығы – 25%.

$$C_d = 1843350,69 \cdot (1 + 0.25) = 23041880,36 \text{ тг}$$

2019 жылы ҚҚС мөлшері 12% деңгейінде белгіленеді
ҚҚС қоса алғанда, іске асыру бағасы мынадай формула бойынша есептеледі:

$$C_p = C_d + C_d * НДС.$$
$$C_p = 23041880,36 + 23041880,36 \cdot 0,12 = 25806906,00 \text{ тг}$$

4.6 Бағдарламалық өнімнің экономикалық бөлімі бойынша қорытынды

Осы бөлімде әзірленген бағдарламалық өнімді талдау оны жүзеге асырудың өзектілігі мен экономикалық пайдасы тұрғысынан жасалады. Ең алдымен, бағдарламалық өнімді құру бойынша жұмыстың күрделілігі мен мерзімдері анықталды. Келесі қадам - әзірленген веб-қосымшаны іске асыру және енгізу үшін қажетті шығындарды есептеу. Соңғы кезең осы бағдарламалық өнімді енгізудің экономикалық тиімділігін есептеу болып табылады.

Осылайша, әзірленген бағдарламалық өнімнің құны тең 702 704,13 теңге,
Іске асыру құны = 25806906,00 теңге
Бағдарламалық өнімнің табыстылығы = 460837,67 теңге

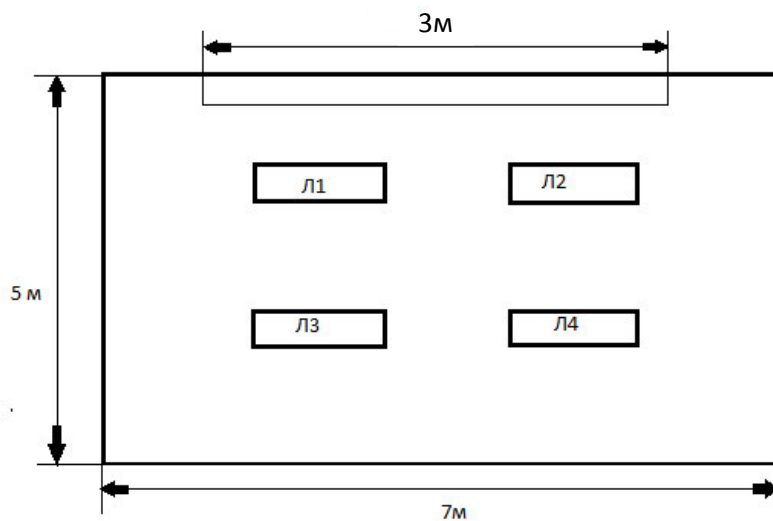
5 Өміртіршілік қауіпсіздігі бөлімі

Дипломдық жұмыстың тақырыбы – ЖОО арналған білім алушының білімін бағалауға арналған тестілеудің OnLine бағдарламалық камтамасын құру. Екі программист жұмыс жасайтын кабинеттің ұзындығы 7 м, ені 5 м және биіктігі 3 м. Бұл бөлмеде жұмыс орнының желдету жүйесі мен дыбыс өткізбеу жүйесі талапқа сәйкес келеді, өйткені онда шулы құрылғылар жоқ және кәзіргі уақыттағы кондиционер ауа температурасын, ауа ылғалдылығын сақтайды. Ал жарықтандыру жүйесін дұрыстау керек болды. Осы себепті дипломдық жұмыстың бұл бөлімінде бөлменің жарықтандыру жүйесін есептеп, екі программистке жұмыс жасауға қолайлы жағдай жасалу туралы бірауыздан шешім қабылданды.

5.1 Табиғи жарықтандыру

Екі программист жұмыс жасайтын бөлменің ұзындығы 7 м, ені 5 м және биіктігі 3 м.

Көрініс жұмыстар санаты – III, в, терезе биіктігі – 1,8 м, терезенің басталу биіктігі 0,6 м, 5.1-суретте көрсетілген.



5.1-сурет – Жұмыс жасау бөлмесінің сызбасы

Бұл бөлмеде ені 3 м және биіктігі 1,8 м болатын бір терезе бар.

Бөлмені қалыпты жарықтандыру үшін жарық саңылауларының ауданы $S = 5.4 \text{ м}^2$ болатындығын тексерейік. Келесі формула арқылы жанама жарықтандыру кезіндегі жарық саңылауларының ауданын алдын ала есептеуден тұрады.

$$100 \frac{S_0}{S_n} = \frac{e_N K_3 \eta_0}{\tau_0 r_1} K_{3d}, \quad (5.1)$$

мұндағы, S_0 – жанама жарықтандыру кезіндегі жарық саңылауларының ауданы, m^2 ;

S_n – бөлме еденінің ауданы, m^2 ;

e_n – ТЖЕ-нің қалыптасқан мәні;

$K_3 = 1,2$ – қордың еселеуіші;

τ_0 – жарық өткізудің жалпы еселеуіші;

r_1 – жарықтың арқасында бөлменің беткі қабатынан және ғимараттың іргелес қабатынан көрінетін, жанама жарықтандыру кезіндегі ТЖЕ-нің қалыптасқан мәнінің өсу қарқынын ескеретін коэффициент;

$K_{30} = 1$ – ғимараттарды қарсы көлеңкелеу терезелерін көрсететін коэффициент.

Еден ауданы (5.2) формуламен анықталады:

$$S_n = L * B \quad (5.2)$$

Осылайша, еден ауданы мынаған тең:

$$\begin{aligned} L &= 7 \text{ м,} \\ B &= 5 \text{ м,} \\ S_n &= 7 * 5 = 35 \text{ м}^2 \end{aligned}$$

Табиғи жарық коэффициентінің нормаланған мәні (5.3) формула бойынша есептеледі

$$e_N = e_n * m_N \quad (5.3)$$

мұндағы, N – табиғи жарықтың болуына қарай әкімшілік-аумақтық аймақтың топтарының саны;

$e_n = 1,2$ – табиғи жарықтандыру жүйесі және осы бөлменің көрініс жұмыстарының санаты сипаттамасына сай ҚНМЕ 23-05-2011 бойынша таңдалған табиғи жарықтандырудың коэффициентінің мәні, яғни жанама жарықтандыру кезіндегі ТЖЕ қалыптасқан мәні;

$m_N = 0,65$ – жарық саңылауларының түріне қарай, SIP кестелерінде орналасқан, олардың көкжиегінде бағдарлануын және әкімшілік топтар тобының санына байланысты жарық климатының коэффициенті оңтүстіктен сыртқы қабырғалардағы жарық климатының коэффициенті.

Өртүрлі аймақтардағы ғимараттар үшін ТЖЕ қалыптасқан мәні e_N келесі формулада

$$e_N = e_n * m_N = 1.2 * 0.65 = 0.78 \%$$

Бір жақты жарықтандырылған бөлменің тереңдігі

$$l = B - 1 = 5 - 1 = 4 \text{ м}$$

η_0 жарық сипаттамаларының мәнін алу үшін келесі арақатынас есептелінді

$$\frac{L}{l} = \frac{7}{4} = 1,75$$

$$\frac{l}{h_{расч}} = \frac{l}{(h_{ок} + h_{нок}) - h_{pn}} = \frac{4}{(1,8 + 0,6) - 0,8} = 2,5$$

Алынған мән кестедегі мәнмен сәйкес келмегендіктен интерполяциялаймыз

$$x = 9,5 + \frac{(10,5 - 9,5) * (2,5 - 2)}{3 - 2} = 10$$

$$\eta_0 = 10$$

τ_0 жарық жіберу коэффициентінің жалпысы (5.4) формуламен анықталады

$$\tau_0 = \tau_1 * \tau_2 * \tau_3 * \tau_4 \quad (5.4)$$

мұндағы, τ_1 – жарықөткізгіш жабдықтың түрі. Терезе шыны парағы үшін: жалғыз, $\tau_1 = 0.8$;

τ_2 – терезе байланысы кезінде жарықтың жоғалуын ескеріп отыратын коэффициент. Ағаш терезе рамалары үшін $\tau_2 = 0.75$;

τ_3 – қолдаушы құрылымдардағы жарықтың жоғалуын ескеріп отыратын коэффициент, жанама жарықтандыру 1ге тең;

$\tau_4 = 1$ – күннен қорғау құрылғылары кезінде жарықтың жоғалуын ескеріп отыратын коэффициент.

Белгілі мәндерді (5.4) формулаға қою арқылы τ_0 табамыз:

$$\tau_0 = \tau_1 * \tau_2 * \tau_3 * \tau_4 = 0.8 * 0.75 * 1 * 1 = 0.6$$

Жанама жарықтандыруға арналған r_1 коэффициентін анықтаймыз. Бұл үшін табамыз: бөлменің тереңдігін әдеттегі жұмыс бетінің деңгейінен терезенің жоғарғы жағына дейін қатынасы:

$$\frac{l}{h_{расч}} = \frac{4}{1,6} = 2,5$$

есептік нүкте мен сыртқы қабырға арасындағы бөлменің тереңдігіне ара қатынасы

$$\frac{l}{B} = \frac{4}{5} = 0,8$$

бөлменің ұзындығының тереңдікке қатынасы

$$\frac{L}{l} = \frac{7}{5} = 1,4$$

Төбенің, қабырғалардың және еденнің орташа өлшенген коэффициенті (5.5) формула бойынша анықталады

$$\rho = \frac{\rho_{ном} + \rho_{см} + \rho_{пол}}{3} \quad (5.5)$$

(5.5) формула бойынша

$$\rho = \frac{\rho_{ном} + \rho_{см} + \rho_{пол}}{3} = \frac{50 + 30 + 10}{3} = 30\% = 0,3$$

Алынған мәндер кестедегі мәндерге сәйкес келмейтіндіктен, біз интерполяция жасаймыз

5.1-кесте – r_1 коэффициентінің мәні

Бөлменің тереңдігін терезенің жоғарғы жағындағы әдеттегі жұмыс бетінің деңгейіне дейін қатынасы	Есептелген нүктенің қашықтығының сыртқы қабырғадан бөлме тереңдігіне қатынасы	Жанама жарықтандыру кезіндегі r_1 мәні		
		Есептелген нүктенің қашықтығының сыртқы қабырғадан бөлме тереңдігіне қатынасы		
		0,3		
		Бөлме ұзындығының оның тереңдігіне қатынасы		
		0,5	1	2 және оданда көп
2,5-нан жоғары	0,1	1	1	1
3,5-ға дейін	0,2	1	1,05	1,05
	0,4	1	1	1,1
	0,6	1,6	1,8	1,6
	0,8	1,9	1,7	1,4

	1	2,6	2,2	1,7
--	---	-----	-----	-----

$$x = 1.7 + \frac{(1.4 - 1.7) * (1.4 - 1)}{2 - 1} = 1.58$$

сонда $r_1 = 1.58$

Берілген мәндерді алып, жанама жарықтандыру кезіндегі жарық саңылауларының ауданын (5.1) формулаға қойып есептейміз:

$$S_0 = \frac{S_n e_N K_3 \eta_0 K_{30}}{100 \tau_0 r_1} = \frac{35 * 0.78 * 1.2 * 10 * 1}{100 * 0.6 * 1.58} = 3.455 \text{ м}^2 \approx 3 \text{ м}^2$$

Қорытынды. Бұл кеңседе жарық саңылауының ауданы тексерілді және ол кеңседе жеткілікті болып табылады. Қыс мезгілінде және тәуліктің түнгі уақыттарында қосымша жарық көздері керек деген қорытындыға келдік.

5.2 Жасанды жарықтандыруды тексеру

Бөлмеде жасанды жарықтандыру жалпы жарықтандыру шамдарындағы 3570 лм жарық ағыны бар 4 65 Вт люминесцентті лампалармен жүзеге асырылады (шам типі – ПВЛМ). 300 лк жарық жасау үшін III-санатты көрініс жұмыстарын жеткілікті қамтамасыз ете алатын пайдалану коэффициенті әдісі арқылы шамдардың санын есептейміз. Басқару бөлмесінде қуаты 65 Вт, жарық ағыны 3570 лм, диаметрі 38 мм және ұзындығы түйреуіштерімен 1515 мм болатын ЛД-65 газоразрядты люминесцентті лампасы қолданылады.

Шамдардың санын анықтау үшін келесі формула коллданылады.

$$N_{\text{л}} = \frac{S_{\text{сум}} K_3 Z E_H}{\Phi_{\text{л}} n \eta} \quad (5.6)$$

мұндағы, Z – минималды жарықтанудырудың коэффициенті (орташа және минималды жарықтандырудың қатынасы). Есептеу кезінде z коэффициенті 1,1÷1,2 аралығында қабылданады;

$n = 2$ – шамдағы лампа саны;

η – кеңсе индксіне және шағылысу коэффициентіне байланысты, қолданыс коэффициенті

K_3 – қор коэффициенті. Шаң жоқ жердегі кеңселер үшін $K_3=1,3$

Жарықтандыру қондырғысын пайдалану кезінде жұмыс беттеріндегі жарықтандыру жарық көздерінің жарық сәулесін, лампалар мен жарықтандыру құрылғыларының ластануын, сондай-ақ жарықтандырылған бөлменің қабырғаларын және төбесін лаптау арқылы азайтылады. Сондықтан, жарықтандыру қондырғысының қуатын анықтау кезінде қауіпсіздік

коэффициенті енгізіледі. Қауіпсіздік коэффициенті ауаның ластану деңгейіне, яғни шаңнан, түтіннен, түктің және т.б. байланысты

Біркелкі жарықтылықты сипаттайтын Z факторы көптеген айнымалылардың функциясы болып табылады және шамдар арасынан ілгіштің (L / h) есептік биіктікке дейінгі арақатынасына тәуелді болады, бұл ретте ұсынылған Z мәндері күрт артады. λ ұсынылған мәннен асып кетпегенде, сіз:

$Z = 1,15$ – қыздыру лампалары және сынапт доғалы лампа(ДРЛ) үшін;

$Z = 1,1$ – люминесцентті лампа үшін [15].

Жарықтандыру қондырғыларын пайдалану коэффициенті η – сәуле түсетін ағынның жұмыс бетіне сәуле көзінен шыққан жарық ағынына қатынасы.

Пайдалану коэффициенті люминесатор түріне, бөлменің геометриялық өлшемдеріне және бетінің көрсететін коэффициенттеріне байланысты.

Пайдалану коэффициентін анықтау үшін кеңсенің индексін анықтау керек:

$$i = \frac{S_n}{(L + B)h_{расч}} = \frac{A * B}{(L + B)h_{расч}} \quad (5.7)$$

мұндағы, S , L , B – кеңсенің сәйкес ауданы, ұзындығы, ені

(5.7) формулаға алынған мәндерді қойып, мынадай нәтиже аламыз:

$$i = \frac{A * B}{(L + B)h_{расч}} = \frac{7 * 5}{(7 + 5) * 2} = 1.4$$

потолок	80	80	80	70	50	50	30	0
стены	80	50	50	50	50	30	30	0
пол	30	30	10	20	10	10	10	0
0.6	65	43	34	41	40	34	33	28
0.8	74	53	43	50	48	42	41	36
1	81	60	49	57	54	48	48	42
1.25	87	69	57	64	61	56	55	49
1.5	91	74	62	69	65	60	59	54
2	96	82	68	76	70	66	65	60
2.5	100	87	73	80	74	71	70	65
3	102	92	77	84	78	75	73	69
4	105	96	80	87	80	78	76	72
5	106	99	83	90	82	80	79	75

5.1-сурет – Люминесцентті лампаның жарық ағынын пайдалану коэффициенті

Есептеу кезінде жалғыз лампаның жарық ағыны берілген әдіс бойынша келесі формуламен шығарылады.

мұндағы, E_{\min} – минималды нормальді жарықтандыру, лк;
 k – қор коэффициенті (люминесцентті лампа үшін $k=1,3$);
 S – жарықтандыру ауданы, м²;
 Z – минималды жарықтандыру коэффициенті (теңсіз жарықтандыру коэффициенті);
 N – шам саны;
 n – шамдағы лампа саны;
 h – шамдар ағынының біреуінің бірлігінде пайдалану коэффициенті.

Жасанды жарықтандыру жеткіліктілі ме соны тексеру үшін E_{\min} есептеу керек және ол 300 лк-дан үлкен боу керек.

$$E_{\min} = \frac{\Phi_l * N * n * \eta_3}{S * Z * K} = \frac{3570 * 4 * 2 * 0.42}{35 * 1.1 * 1.3} = 240$$

Есептеулерге қарағанда E_{\min} аз шықты. Сол себепті бұл мәселені шешу керек. Мен жарық ағыны 3570 лм, қуаты 65 Вт болатын люменисцентті лампаны жарық ағыны 4070 лм, қуаты 80 Вт болатын жаңа люменисцентті лампаға ауыстыруды жөн санадым. Содан кейін тағы да E_{\min} тексеріп шықтым:

$$E_{\min} = \frac{\Phi_l * N * n * \eta_3}{S * Z * K} = \frac{4070 * 4 * 2 * 0.42}{35 * 1.1 * 1.3} = 273$$

Қорытындылай келе, кеңсенің жарықтың жайлы болуына жеткілікті болды. Шамдардың қуатын арттыру керек, яғни 65 Вт үлкен қуатты лампаны таңдау керек. Осылайша қуаты 80 Вт, жарық ағыны 4070 лм болатын ЛД люминесцентті лампасы шамдардағы жалпы жарықтандыру үшін таңдалды.

Жарық ағыны 4070 лм тең 300 лк жарықтандыруды құруға арналған шамдар санын келесі формуламен анықтаймыз:

$$N = \frac{S_{\text{помещ}} K_3 Z E_H}{\Phi_l n \eta} = \frac{35 * 1.3 * 1.1 * 300}{4070 * 2 * 0.42} = 4.391 \approx 4$$

Яғни көрініс жұмысының разряды III болатын 300 лк жарықтандыру құру үшін шам типі ЛД-80 және жарық ағыны 4070 лм тең 8 лампасы бар 4 шам қолданылады.

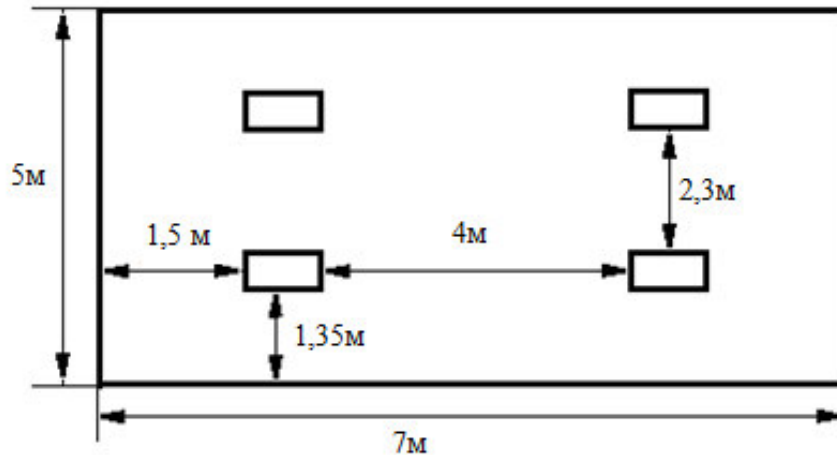
Жұмыс орындағы шамның іліну биіктігін (5.8) формуламен анықтаймыз

$$h_{\text{расч}} = H_{\text{пом}} - (H_{\text{св}} + H_{\text{р.п.}}), \quad (5.8)$$

мұндағы, $H_{\text{св}}=0$ – шамдардың көтерілу биіктігі, м;

$H_{\text{р.п.}}=0,8$ – еденнен жоғары жұмыс бетінің қашықтығы, м;

$H_{\text{пом}}=3$ – бөлме биіктігі, м.



5.2-сурет – Бөлмедегі жарық саңылаулары мен шамдардың орналасу сызбасы

Шамдардың іліну биіктігі:

$$h_{\text{расч}} = H_{\text{пом}} - (H_{\text{св}} + H_{\text{р.п.}}) = 3 - (0 + 1) = 2 \text{ м}$$

Шамдардың арасындағы қашықтығын (5.9) формуламен анықтаймыз:

$$L = \lambda * h \tag{5.9}$$

мұндағы, L – Көршілес шамдар немесе люминесцентті шамдар арасындағы қашықтық;

h – жұмыс бетінен шам іліну биіктігі.

Люминесцентті лампалармен шамдарды қолдану кезінде

$$\lambda = 1.2 \div 2.4$$

Осылайша, шамдар арасындағы арақашықтығын табамыз:

$$L = \lambda * h = 2 * 2 = 4 \text{ м}$$

Қатар тұрған шамдар арақашықтығын табамыз:

$$L_b = \lambda * h_p = 2,3 * 1 = 2,3 \text{ м.}$$

Қабырға мен шамдар арақашықтығын табамыз:

$$l_a = l_b = L_b / 3 + [0,3; 0,5] = 2,9 / 3 + 0,5 = 1,5 \text{ м.}$$

Шамдар арасындағы арақашықтығы:

$$L_a = L_b - 1 = 2.3 - 1 = 1.3 \text{ м.}$$

Жарық саңылаулары мен шамдардың орналасу сұлбасы 5.2-суретте көрсетілген

5.3 Өміртіршілік қауіпсіздігі бөлімі бойынша қорытынды

Есептеулер бойынша жасанды жарықтандыруды қалпына келтіру тексерілді. жарық ағыны 3570 лм, қуаты 65 Вт болатын люменисцентті лампаны жарық ағыны 4070 лм, қуаты 80 Вт болатын жаңа люменисцентті лампаға ауыстыруды туралы шешім қабылданды.

Қорытынды

Қорыта келе, қазіргі технология заманында құжаттарды, сұраныстарды және ақпарат алмасуды заманауи құрылғылардың көмегімен оңайлату, қауіпсіз және ыңғайлы ету өте маңызды. Интернеттің адамдар арасында кең қолданысқа ие болуы және интернеттің жылдамдығының артуына байланысты веб сайттар үлкен сұранысқа ие. Дипломдық жұмыс университет ішіндегі барлық сұралымдарды қамтиды. Сұраным сайты оқытушылар мен студенттердің уақытын үнемдеп барлық сұралымдардың тез әрі сапалы орындалуына қол жеткізеді.

Осы жұмыста жұмыс үрдісін автоматтандырудың негізгі бағыттары зерттеліп, университет бөлімінде автоматтандырылған жүйе қалыптастырылды, жүйеге қойылатын функционалдық талаптар анықталды, жүйеге құжат айналымы мен пайдалану схемасы әзірленді.

Жұмысты дайындау барысында OSPanel бағдарламасы және қосымшалар HTML, PHP, JavaScript, және Mysql программалық құралдар пайдаланылды.

Әдебиеттер тізімі

- 1 Э. Харрис. PHP и MySQL для начинающих. – М.: Кудиц-Образ, 2005.
- 2 В. Фаронов. Программирование баз данных. – М.: Нолледж, 2000.
- 3 Х. Скотт. Администрирование Web-сервера Apache и руководство по электронной коммерции. – М.: Вильямс, 2001.
- 4 М.В. Дмитриева. JavaScript. Быстрый старт - СПб.: БХВ-Петербург, 2002.
- 5 Хассон, У. Критерии качества дистанционного образования / У. Хассон. - М.: Высшее образование, 2002. -365с.
- 6 Яковлев А.А. Раскрутка сайтов. Основы, секреты, трюки / А.А. Яковлев, В.Г. Ткачев. - СПб. : БХВ-Петербург, 2010. - 344с.
- 7 Скляр, Д. PHP. Рецепты программирования / Д. Скляр. - СПб.: БХВ - Петербург, 2007. -736с.
- 8 Бенкен Е.С. PHP, MySQL, XML. Программирование для Интернета / Е.С. Бенкен. - СПб. : БХВ-Петербург, 2007. - 310с
- 9 Куатова Д.Я. Экономика предприятия. - Алматы: «Экономика», 2011.
- 10 Экономика, организация и управление на предприятии. Учебник / Корсаков М.Н., Ребрин Ю.И., Федосова Т.В., Макареня Т.А., Шевченко И.К. и др.; под ред. Боровской М.А. - Таганрог: ТТИ ЮФУ, 2008. -440с.
- 11 <http://adilet.zan.kz/rus/docs/P070001400> «О системе оплаты труда гражданских служащих, работников организаций, содержащихся за счет средств государственного бюджета, работников казенных предприятий» Постановление Правительства Республики Казахстан от 29.12.2007г. N1400.
- 12 Стандарт организации. Общие требования к построению, изложению, оформлению и содержанию учебно-методических и учебных работ. СТ НАО 56023-1910-04-2014. Издание официальное. - Алматы, 2014.

А қосымшасы (міндетті)

Техникалық тапсырма

А.1 Кіріспе

Қазіргі технология заманында құжаттарды, сұраныстарды және ақпарат алмасуды заманауи құрылғылардың көмегімен оңайлату, қауіпсіз және ыңғайлы ету өте маңызды. Интернеттің адамдар арасында кең қолданысқа ие болуы және интернеттің жылдамдығының артуына байланысты веб сайттар үлкен сұранысқа ие. Әрбір адамның қолында смартфондардың болуы интернет арқылы адамдарға қызмет көрсетуді өте ыңғайлы етеді. Университет аумағында студенттердің, оқытушылардың және қызметкерлердің сұраныс алмасуларын жеңілдету мақсатында дипломдық жұмысымды сұраныс жасау бағытында алдым. Студенттер, оқытушылар қашықтықтан өзіне қажетті құжаттарға сұраныс жасап, құжаттың дайын болғандығы жөнінде ақпаратты біле алады. Оқытушылар сұраныс арқылы кабинеттегі бұзылған құралдар жайында инженерлерге хабарлай алады. Дипломдық жұмыс университет ішіндегі барлық сұралымдарды қамтиды. Сұраным сайты оқытушылар мен студенттердің уақытын үнемдеп барлық сұралымдардың тез әрі сапалы орындалуына қол жеткізеді.

Қазіргі уақытта ақпараттық жүйе құруға пайдаланатын бірнеше технологиялар бар: HTML, PHP, PERL, ASP.NET, Silverlight және т.б. Бұл технологиялардың әрқайсысы белгілі бір мақсатта құрылған.

Менің жобам PHP, MySQL технологиялары көмегімен ЖОО арналған білім алушының білімін бағалауға арналған тестілеудің OnLine программалық қамтамасын әзірлеу болып табылады.

А.1.1 Өңдеудің мақсаты мен қызметі

Дипломдық жобаның мақсаты – университет ішінде құжаттарға немесе қызметтерге сұраныс беруді веб қосымшасын қолдана отырып ыңғайлы, әрі жылдам ету. Веб-қосымша оқытушыларға және студенттерге ең аз уақытқа тиімді жүргізуге мүмкіндік береді.

Дипломдық жұмыстың нәтижесінде келесі негізгі тапсырмалар орындалуы қажет:

- талдау бөлімі;
- жобалау бөлімі;
- жүзеге асыру және тестілеу бөлімі.

Программалық қамтаманы әзірлеу барысында PHP скрипт тілі, HTML гипермәтіндік тілі, CSS каскадты стильді кестелер тілі, MySQL деректер қоры және клиенттің жағында да, сервердің жағында да жұмыс жасайтын сценарийлерді жазуға арналған ОБП (объектілік бағытталған программалау) тілі – JavaScript қолданылады.

А қосымшасының жалғасы

А.1.2 Қолдану саласы

Қолдану саласы – Жоғарғы оқу орындары

А.1.3 Анықтамалар, терминдер және қысқартулар

Анықтамалар, терминдер және қысқартулар А.1-кестеде көрсетілген.

А.1-кесте – Анықтамалар, терминдер және қысқартулар

Терминдер немесе қысқартулар	Анықтамалар
UML	бағдарламалық дамытуға графикалық объект модельдеу сипаттау үшін тілді, бизнес-процесстерді модельдеу, жүйелер инженерлік және ұйымдастырушылық құрылымдарды салыстыру
ДҚБЖ	Деректер қорының басқару жүйесі
MySQL	MySQL – еркін байланыс дерекқоры. MySQL бастапқыда MySQL AB-мен сатып алынған швед фирмасының Sun Microsystems-ға тиесілі шведтік бренді Oracle корпорациясы тарапынан әзірленген және қызмет етеді.
PHP	PHP (ағылшынша: PHP: Hypertext Preprocessor – «PHP: еренмәтінді алдын-ала үдірістегіш») – кеңінен қолданылатын скрипттер тілі, ол арнайы түрде WEB-программалау үшін өңделген.
HTML	HTML сөзі – ағылшын тіліндегі “HyperText Markup Language” деген тіркестің қысқартылған нұсқасы. html тілі – ғаламторда документтерді (файл, Интернет парақшасы...) белгілеудің стандартты тілі.
WWW	World Wide Web Интернеттегі барлық құжаттар мен мультимедиа ресурстарын сипаттайды

А.2 Жалпы сипаттамасы

Дерекқор жұмысы клиент-сервер технологиясына негізделген. Оқытушы бөлімінде – студенттерге арналған ақпараттарды өңдеп отырады және студенттер тарапынан түскен сұрақтарды мазмұнына қарай топтайды.

А.2.1 Пайдаланушылық интерфейстер

Деректер қоры және бағдарламалық қамтаманы әзірлеу орындаушымен
А қосымшасының жалғасы

толық игеріледі.

Бұл жобада екі пайдаланушылық интерфейс қарастырылған: Әкімшіге арналған интерфейс; қолданушыға арналған интерфейс.

Әкімшіге арналған интерфейсте дерекқормен толық жұмыс жасауға мүмкіндік бар, басқа да мүмкіндіктері жоба аймағында шексіз. Қолданушыға арналған интерфейсте әкімші бекіткен сұрақтарды көруге және өтініш жазуға мүмкіндіктері қарастырылған.

А.2.2 Аппараттық интерфейстер

Компьютерлерге қойылатын жалпы талаптар:

- а) процессор – жоғары;
- ә) оперативті жады – 256 МВ жоғары;
- б) диск тұлғалы кеңістік – ~ 52 МВ + қолданушылар файлдарын сақтауға қажет орын;
- в) Internet желісіне қатынау;
- г) жұмысты қолдайтын видеокарта;
- ғ) пернетақта, манипулятор тышқан.

А.2.3 Программалық интерфейстер

Серверлік программалы компоненттер:

- а) Windows 7 операциялық жүйесі;
- б) ДҚБЖ MySQL Server 5.0;
- в) PHP соңғы нұсқасы сайтты басқару жүйесі;

Клиенттік программалы компоненттер:

- а) Windows 7/8 операциялық жүйесі;
- б) Internet Explorer, Mozilla Firefox, Opera.

А.2.4 Коммуникациялық интерфейстер

Пайдаланушылардың компьютерлерінде интернетке қатынауға бар модем құралдары немесе жылдамдығы 100 Мб/с кем емес желілі карта болуы тиіс. Сервер мен клиенттерді байланыстыратын TCP/IP протоколы қолданылады.

А.2.5 Жады бойынша шектеулер

Серверлік және клиенттік машиналар үшін жалпы жады бойынша қойылатын талаптар, оларда орналасқан жұмысқа қажетті программалық компоненттердің қойылатын талаптармен анықталады.

А.2.6 Адаптация бойынша талаптар

Программалық кешен Windows 7/8 платформаларында жұмыс істеуі тиіс, бұл немесе басқа платформаларды қолдануда қажетті программалық компоненттер ыңғайлы орнатылуы керек.

А.3 Өңдеуге талаптар

А.3.1 Функциональды талаптар

ЖОО арналған білім алушының білімін бағалауға арналған тестілеудің OnLine программалық қамтамасында әрбір қолданушы өзінің логин және құпиясөзімен қатынас құру қажет. Сол кезде қолданушы өзіне қажетті сұрақтарға жауап алады және өзге де сұрақтарға жауап бере алады.

А.3.1.1 Шығыс деректері

Серверге кіретін мәліметтер ретінде, қолданушылардың сұранысы, объектінің түрі және қосымша мәліметтерді жіберу.

Сонымен қатар, қолданушылармен байланыс орнатын, контакты ақпараттар, яғни қолданушының аты-жөні, телефон номері, мекен-жайы сияқты толық мәліметтер болуы қажет.

Администратор жағынан мынандай мәліметтер шығуы қажет: пайдаланушыдан түскен ақпараттар іріктеледі, егерде пайдаланушы тарапынан қойылып отырған мәселе немесе ұсыныстар есептеу техникасы және бағдарламалық қамтамасыз ету ғылымы саласынан тыс немесе басқа мақсаттарда болған жағдайда алынып тасталынады.

А.3.1.2 Нәтижелер

MySQL жүйесі жойылған компьютерлерді қолданылушылар қосылуға мүмкіндігі бар өзінің серверін ұсынады. ДҚБЖ MySQL – PHP-де қолданылатын көптеген дерекқорлардың арасындағы бірі болып табылады.

SQL(Structured Query Language) – құрылымдық сұраныс тілі. Бұл тіл – әр түрлі деректер қорына қатынау үшін арналған стандартты құрал ретінде болып келеді.

А.3.1.3 Функциональды талаптарды ұйымдастыру

Жүйенің жұмысын, техникалық тапсырманың талаптарымен салыстыру. Барлық сынау жұмыстары, анықталған құрал – жабдықтарда өткізілуі қажет.

Ол келесідегідей құралдармен, яғни жергілікті желі арқылы
А қосымшасының жалғасы

байланысқан екі компьютер арқылы жүзеге асады. Оның біріншісі сервер болады да, оған Windows (64) операциялық жүйесі, Apache сервері және PHP 7, MySQL Server 5.6 программалық қамтамалары орнатылады. Ал клиенттік машинаға Windows (64) операциялық жүйесі, интернет қосылу мүмкіндігін беретін бағдарламалар орнатылады.

Класс бойыншы талаптарды ұйымдастыру – бұл технология алдымен жіктелуін анықтау керек болатын талаптар ұйымының объектті-бағдарланған стилін қолданады, яғни класстарды таңдау эквиваленті.

Заттық облысының негізгі (базалық) түсініктерін жіктегеннен кейін, қалған талаптарды алынған категориялар немесе класстар бойынша орналастырады. Ол үшін істің екі ыңғайы бар:

– класстарды талаптарды ұйымдастыру тәсілдері сияқты қарастыруға болады, бірақ жобалауда оларды міндетті қолданушы деп есептемеу;

– объектілі-бағдарланған жобалауда және іске асыру талаптары үшін өңделген класстарды қолдануға болады.

Екінші пункті бағдарламалау тілінің функциясымен және әрбір функционалды толық талаптар арасында сәйкестік құрып, әдістермен және толық талаптар арасындағы біртекті жалғасуды қолдайды.

Іске асыру және жобаның, талаптар арасындағы қатал талаптарды қолдау, келешекте жобалауда қолданылатын класс бойынша талаптарды ұйымдастыру үлкен артықшылығы болып табылады. Бұдан басқа қазіргі әлем түсініктеріне сәйкес келетін кластар үшін, оларды қайта қолдану үлкен ықтималдылығы бар.

А.3.2 Функциональды емес талаптар

Берілген сұрақ бойынша жауап алу үшін уақыт керек.

А.3.2.1 Өнімділігі

Жүйе сервердің және кез келген операциялық жүйенің максималды жүктемесі бойынша тиімді жұмыс істеуі керек. Демонстрациялар мен іздеу жүйелерінде үлкен көлемде деректер болуы керек. Веб беттердің орташа жүктеу уақыты 28,8 Кбит / с-тан 30 сағаттан аспауы керек. Бөлек беттер үшін ең көп жүктеу уақыты 35 секундтан аспауы керек, бірақ барлық пайдаланушылардың 30% -дан аспауы керек. Бастапқы бет үшін ең көп жүктеу уақыты 40 аспауы керек.

А.3.2.2 Қатынау және сенімділік

Жүйе пайдаланушылардың сенімді тоқтаусыз жұмысын қамтамасыз етіп, олардың кез-келген ісіне тоқтаусыз қызмет түрін көрсетуі керек.

Жүйе бірнеше клиенттік түйіндер қосылған кезде тиімді жұмыс істеуі тиіс және клиент – компьютерлер мен сервер-компьютердің синхронизациясын қамтамасыз етуі керек.

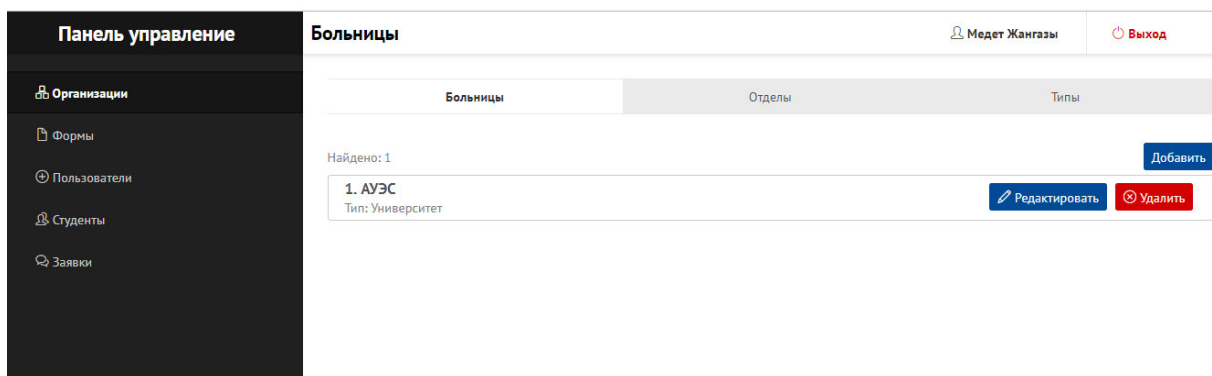
А.3.2.3 Қатені өңдеу

Барлық жүйелік қателер өңделіп, пайдаланушының қате туралы хабарлары көрсетілуі керек. Хабар алмасу жүйесінде пайдаланушылар дұрыс емес ақпаратты енгізуі керек.

А.3.2.4 Интерфейсті талаптар

Дерекқордың құрылымының талаптары - дерекқорды навигациялау негізгі мәзір арқылы жүзеге асырылады. Дерекқор қазақ тілінде жасалады. Дерекқорды өзгертуге тек әкімші ғана құқылы.

Келесі кезең – кешендегі ақпараттарды өзгерту және тұрақты жаңартып отыру. Кешенді ыңғайлы және шапшаң жаңарту үшін, кешенді басқару құралдары игеріледі. Оның арқасында кешенді сүйемелдеуді – веб-қосымшаның администрациялаудың негізгі түсініктерін білетін пайдаланушы асыра алады. Веб-қосымша ақпаратты жаңарту Web-интерфейс арқылы жүзеге асады. Веб-қосымша дерекқорын өзгерту администратор интерфейсі арқылы орындалады. Төмендегі суретте веб-қосымшаның бас беті көрсетілген.



А.4-сурет – Оқытушы терезесі

А.3.2.5 Шектеулер

Әкімшілік бөлім сайт безендірілуіне және оның құрылымына өзгеріс енгізуіне мүмкіндік бермейді, тек қана оның ақпараттық бөліміне рұқсат етеді.

Веб-қосымшаның жұмысқа қабілеттілігі провайдер серверінің жұмысына

тығыз байланысты, яғни сервер жұмысының өзгерісі дерекқор жұмысында көрсетілім табады.

А.3.3 Кері талаптар

Барлық енгізілетін мәліметтер алдын ала тексерістен өту керек.

Ә ҚОСЫМШАСЫ

Программа листингі

```
<!DOCTYPE html>
<html>
<head>
    <?php include_once(P."backend/template/external.php"); ?>
    <title><?php echo $title; ?> | Панель управление</title>
</head>
<body>

<div class="col-3 h100">
    <?php include_once(P."backend/template/menu.php"); ?>
</div>
<div class="col-9 h100">
    <?php include_once(P."backend/template/head.php"); ?>
    <div class="admin-content">
        <div class="admin-top">
            <p class="aleft">Найдено: <?php echo count($forms); ?></p>
            <a href="#add-form" class="aright">Добавить</a>
            <div class="clear"></div>
        </div>
        <?php foreach($forms as $form): ?>
            <div class="admin-item">
                <div class="col-8">
                    <a href="<?php Page::uri(); ?>
manager/panel/form/update/<?php echo $form["form_id"]; ?>"><h3><?php
echo $form["name"]; ?></h3></a>
                    <p><?php echo $form["description"]; ?></p>
                </div>
                <div class="col-4 cntr">
                    <div class="aright">
                        <a href="<?php Page::uri();
?>manager/panel/form/update/<?php echo $form["form_id"]; ?>" class="update"><i
class="icon-pencil"></i> Редактировать</a>
                        <a href="<?php Page::uri();
?>manager/panel/form/delete/<?php echo $form["form_id"]; ?>" class="delete"><i
class="icon-close"></i> Удалить</a>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

        <?php endforeach; ?>
    </div>
</div>
<!-- get Modal -->
<div id="add-form" class="modal">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h3 class="modal-title">Добавить форму</h3>
                <a href="#close" title="Close" class="close">×</a>
            </div>
            <div class="modal-body form">
                <form method="POST">
                    <label>Имя</label>
                    <input type="text" name="name" required>
                    <br>
                    <label>Описание</label>
                    <textarea name="description" required></textarea>
                    <br>
                    <label>Формат</label>
                    <select name="format">
                        <option value="4">A4</option>
                        <option value="5">A5</option>
                        <option value="6">A6</option>
                        <option value="7">A7</option>
                    </select>
                    <br>
                    <label>Ориентация</label>
                    <select name="orientation">
                        <option value="p">Portret</option>
                        <option value="m">Landscape</option>
                    </select>
                    <button type="submit" name="add_form">Добавить</button>
                </form>
            </div>
        </div>
    </div>
</div>
<!-- end Modal -->
</body>
</html>

```

```

<!DOCTYPE html>

```

```

<html>
<head>
    <?php include_once(P."backend/template/external.php"); ?>
    <title><?php echo $title; ?> | Панель управление</title>
</head>
<body>
<div class="col-3 h100">
    <?php include_once(P."backend/template/menu.php"); ?>
</div>
<div class="col-9 h100">
    <?php include_once(P."backend/template/head.php"); ?>
    <div class="admin-content">

        <div class="admin-transition">
            <div class="col-4 cntr"><a>Организации</a></div>
            <div class="col-4 cntr tab"><a href="<?php Page::uri();
?>manager/panel/departments">Отделы</a></div>
            <div class="col-4 cntr tab"><a href="<?php Page::uri();
?>manager/panel/hospital-types">Типы</a></div>
            <div class="clear"></div>
        </div>

        <div class="admin-top">
            <p class="aleft">Найдено: <?php echo count($hospitals);
?></p>

            <a href="#add-hospital" class="aright">Добавить</a>
            <div class="clear"></div>
        </div>

        <?php $i=0; foreach($hospitals as $hospital): $i++;?>
            <div class="admin-item">
                <div class="col-8">
                    <a href="<?php Page::uri();
?>manager/panel/hospital/update/<?php echo $hospital["hospital_id"];
?>"><h3><?php echo $i." ". $hospital["name"]; ?></h3></a>
                    <?php $type =
Hospital::getType($hospital["type_id"]); ?>
                    <p>Тип: <?php echo $type["name"]; ?></p>
                </div>
                <div class="col-4 cntr">
                    <div class="aright">
                        <a href="<?php Page::uri();
?>manager/panel/hospital/update/<?php echo $hospital["hospital_id"]; ?>"
class="update"><i class="icon-pencil"></i> Редактировать</a>

```

```

        <a href="<?php Page::uri();
?>manager/panel/hospital/delete/<?php echo $hospital["hospital_id"]; ?>"
class="delete"><i class="icon-close"></i> Удалить</a>
        <div class="clear"></div>
    </div>
</div>
<div class="clear"></div>
</div>
<?php endforeach; ?>
</div>
<!-- get Modal -->
<div id="add-hospital" class="modal">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h3 class="modal-title">Добавить больницу</h3>
                <a href="#close" title="Close" class="close">×</a>
            </div>
            <div class="modal-body form">
                <form method="POST">
                    <label>Имя</label>
                    <input type="text" name="name" required>
                    <br><br>
                    <label>Тип</label>
                    <select name="type">
                        <?php foreach ($types as $type): ?>
                            <option value="<?php echo $type["id"]; ?>"><?php
echo $type["name"]; ?></option>
                        <?php endforeach; ?>
                    </select>
                    <br><br>
                    <label>Адрес</label>
                    <input type="text" name="address" >
                    <br><br>
                    <label>Телефон</label>
                    <input type="text" name="phone">
                    <br><br>
                    <label>Описание</label>
                    <textarea name="description"></textarea>
                    <br>
                    <button
                                                                type="submit"
name="add_hospital">Добавить</button>
                </form>
            </div>
        </div>
    </div>
</div>

```



```

        </div>
    </div>
</div>
</div>
<!-- end Modal -->
</body>
</html>
<!DOCTYPE html>
<html>
<head>
    <?php include_once(P."backend/template/external.php"); ?>
    <title>Вход | Панель управление</title>
</head>
<body>

<div class="admin-background">
    <div class="admin-login">
        <h1>Панель управление</h1>
        <?php
        if(isset($errors) AND is_array($errors)):
            foreach($errors as $error):
                ?>
                    <p class="errors"><?php echo $error; ?></p>
                <?php
            endforeach;
        endif;
        ?>
        <form method="POST">
            <input type="email" name="email" placeholder="Email"
<?php echo $email; ?>>
            <input type="password" name="password"
placeholder="Пароль">
            <button type="submit" name="login">Вход</button>
        </form>
    </div>
</div>

</body>
</html>

```

Б қосымшасы
(міндетті)

Ендіру актісі